Schutz der Privatsphäre in kontext- und ortsbezogenen Diensten

Florian Dorfmeister

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität München

vorgelegt von
Florian Dorfmeister

Tag der Einreichung: 21. Oktober 2016

Schutz der Privatsphäre in kontext- und ortsbezogenen Diensten

Florian Dorfmeister

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität München

vorgelegt von
Florian Dorfmeister

1. Berichterstatter:	Prof. Dr. Claudia Linnhoff-Popien
2. Berichterstatter:	Prof. Dr.-Ing. Lars Wolf
Tag der Einreichung:	21. Oktober 2016
Tag der Disputation:	28. März 2017

Eidesstattliche Versicherung

(siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

Florian Dorfmeister

Zusammenfassung

Mit der immensen Verbreitung von Smartphones als leistungsstarke, mobile Endgeräte nimmt auch die Nutzung kontext- und insbesondere ortsbezogener Dienste stetig zu. Derartige Anwendungen vereinfachen die Interaktion mit dem eigenen Endgerät oder externen Systemen, ermöglichen neuartige Nutzungserlebnisse und innovative Dienste, die auf den aktuellen Nutzungskontext zugeschnitten sind. Bei einem Großteil der hierfür an Dritte kommunizierten Informationen handelt es sich jedoch um persönliche Daten, deren unkontrollierte Herausgabe aus Sicht der Privatsphäre problematisch erscheint.

In der vorliegenden Arbeit werden drei unterschiedliche Möglichkeiten zum Datenschutz von Kontextinformationen vorgestellt. Allen Verfahren ist gemein, dass sie im Gegensatz zu vielen bestehenden Systemen ohne die Existenz einer als vertrauenswürdig deklarierten dritten Partei auskommen. Stattdessen wird jeweils eine rein clientseitige Durchsetzung von Privatsphärepräferenzen angestrebt, wodurch eine personalisierte Dienstnutzung ermöglicht und die Gefahr eines zentralen Datenlecks vermieden wird.

Der erste Ansatz beschäftigt sich damit, dem Benutzer ein effektives, allgemeingültiges Werkzeug zur feingranularen, situations- und rezipientenabhängigen Verwaltung von Kontextinformationen zur Verfügung zu stellen. Es wird ein Ontologie-basiertes Kontextmodell entwickelt, auf dessen Grundlage die Definition und konsistente Durchsetzung situationsabhängiger Freigaberegeln möglich ist. Zudem wird eine vollständige Systemarchitektur zur Kontextverwaltung sowie deren Integration in ein mobiles Betriebssystem beschrieben.

Der zweite Ansatz ermöglicht die privatsphäreschonende Umsetzung der verkehrsadaptiven Online-Routenplanung. Unter Verwendung standardmäßig zur Verfügung stehender Dienstschnittstellen wird dafür gesorgt, dass keine externe Komponente den exakten Start- und Zielpunkt einer Routenanfrage in Erfahrung bringen kann. Anhand einer umfangreichen Evaluation werden der Trade-Off zwischen Privatsphäre, Kommunikationsaufwand und Dienstqualität untersucht und verschiedene Optimierungsmöglichkeiten aufgezeigt.

Als drittes wird ein umfassendes Konzept zur Herstellung von Standortanonymität vorgestellt, das sich generisch für die privatsphärekonforme Positionsfreigabe in unterschiedlichen Ausprägungen ortsbezogener Dienste eignet. Hierfür werden die topologiebasierte Erstellung k-anonymer Verschleierungszonen sowie verschiedene Freigabestrategien entwickelt, die auch die zeitliche Korrelation aufeinanderfolgender Ortsangaben berücksichtigen. Dies ermöglicht den effektiven Schutz persönlicher Daten selbst bei kontinuierlichen Positionsupdates gegenüber einem Angreifer mit umfangreichem Kartenwissen.

Abstract

With the widespread prevalence of smartphones as powerful ultra-mobile devices, also the usage of context-aware applications and location-based services continually grows. Such applications improve the way a user interacts with his own device and external systems. Furthermore, they enable previously unknown user experiences and offer innovative services tailored to the user's current situation. The majority of context information that has to be communicated to external parties in order to use such services, however, is considered personal data. From a privacy oriented perspective, the release of this kind of information hence has to be controlled and leakage must be prevented.

This work presents three different means for protecting a user's context information. In contrast to many existing approaches, each of the proposed systems has been designed to operate without the existence of an omniscient, trusted third party acting as an anonymizer. Instead, enforcement of a user's privacy preferences is executed locally on the user's device, which allows for personalized services and avoids the perils of a central privacy bottleneck.

The first approach proposes an effective and generally applicable tool allowing the user to manage his context information in a fine-grained, context-aware and recipient-dependent way. To this end, a new ontology-based context model will be developed, which forms the foundation for the definition and assertion of situation-dependent access control rules set up by the user. Additionally, the overall system architecture as well as its integration into a modern mobile operating system will be described.

The second approach presents a client-side implementation for using traffic-adaptive online route planning services in a privacy-preserving manner. Only using the unmodified standard query interfaces of existing services, the system assures that no external party is able to learn the exact endpoints of the user's route request. By means of empirical evaluation on the actual road network, the trade off between privacy, communication overhead, and quality of service will be analyzed. Also, different optimizations will be discusssed.

Thirdly, a holistic concept for continuously protecting a user's location privacy will be presented, which is generally applicable to the release of location information and all different kinds of location-based services. A topology-aware creation of k-anonymous cloaking regions will be developed as well as different strategies for the release of location information, which also take into account the spatiotemporal correlation of successive location updates. These allow for an effective protection of a user's location privacy even for continuous location updates and in face of a strong attacker with extensive map knowledge.

Inhaltsverzeichnis

Danksagung

Herzlich bedanken möchte ich mich bei Frau Prof. Dr. Claudia Linnhoff-Popien, die mich während der Betreuung dieser Dissertation stets unterstützt und motiviert hat. Mein weiterer Dank gilt Herrn Prof. Dr.-Ing. Lars Wolf für die Übernahme des Koreferats und die anregende Diskussion meiner Ansätze.

Für ihre große fachliche und moralische Unterstützung möchte ich mich auch bei all meinen Freunden und Kollegen am Lehrstuhl für Mobile und Verteilte Systeme bedanken.

Mein größter Dank aber gebührt meiner Familie, insbesondere meiner wundervollen Frau und zugleich besten Freundin Judyta, die mich in jedem Moment mit großer Liebe und viel Verständnis unterstützt hat.

1 Einführung und Motivation

„We know where you are, with your permission.
We know where you've been, with your permission.
We can more or less know what you're thinking about."
— Eric E. Schmidt, *damals CEO Google* [234]

Der kommerzielle Erfolg von Smartphones, Tablets, Wearables und Co. führt zu einer allgegenwärtigen Verfügbarkeit leistungsfähiger, mobiler Endgeräte. Kontextbezogene Anwendungen und – als deren am häufigsten genutzte Vertreter – ortsbezogene Dienste sind dadurch im Alltag vieler Menschen zu einem selbstverständlichen Hilfsmittel geworden. Die zu Ende des letzten Jahrhunderts u.A. von Mark Weiser noch als wissenschaftliche Vision formulierte Idee einer von Computern und künstlicher Intelligenz durchdrungenen Welt [248] wurde damit in einigen Lebensbereichen bereits Realität.

Durch die verschiedenen in Geräten, Gebäuden und Umwelt verbauten Sensoren fällt eine Vielzahl an personenbezogenen Daten an, die mittels Verfahren zur Kontexterkennung analysiert und interpretiert werden. Darauf basierend wählt das eigene Smartphone z.B. Musik aus, die zur aktuellen Situation oder Aktivität passt. Das „smarte" Zuhause oder Bürogebäude steuert proaktiv Heizung, Fenster und elektronische Gerätschaften. Die scheinbar intelligenten Begleiter erkennen Fortbewegungsmodus, Aktivitäten und sogar Emotionen und Stimmungen ihres Nutzers [164, 139]. Im Zusammenspiel mit Armbändern und Uhren sammeln sie seine Fitness- und Vitalwerte. Sie wissen, wann er zuletzt Sport getrieben hat, können Stürze und Unfälle registrieren, Hilfe holen und Leben retten [41]. Oder sie können dem Versicherer ihres Trägers mitteilen, dass bei der geringen Menge an Bewegung und der Anzahl an Überstunden im letzten Jahr das Gesundheitsrisiko drastisch gestiegen ist.

Kontextbezogene Anwendungen behalten den Kalender ihres Besitzers im Blick und geben rechtzeitig vor einem Termin Bescheid, wann man aufbrechen sollte, um angesichts der aktuellen Verkehrslage nicht zu spät zu kommen. Man wird benachrichtigt, wenn sich Freunde oder andere, von einem Online-Dienst als interessant eingestufte Nutzer in der nächsten Umgebung befinden. Man verwendet den automatisch ermittelten Standort, um Sehenswürdigkeiten, Geldautomaten oder gut bewertete Restaurants in der Umgebung zu finden. Wie selbstverständlich nutzt man heute sein Telefon als Navigationsgerät, um unter Berücksichtigung der in Echtzeit ermittelten Verkehrssituation den schnellsten Weg von A nach B zu finden.

Durch die kontinuierliche Preisgabe des eigenen Standorts und der gefahrenen Geschwindigkeit an den Anbieter eines online Routing-Dienstes trägt man selbst aktiv zur Erhöhung der Dienstqualität bei. Für den Endnutzer sind diese Dienstangebote meist nicht mit finanziellen Aufwendungen verbunden. Zur Finanzierung solcher Dienste wird dem Nutzer stattdessen in vielen Fällen kontextbezogene Werbung eingeblendet. Auch hierfür wird i.d.R. der aktuelle Standort übermittelt, um relevante Angebote ausspielen zu können.

Bei vielen dieser Daten wie z.B. Aktivitäten, Vitalwerte oder aktuelle, zukünftige sowie regelmäßig besuchte Orte, handelt es sich jedoch um zutiefst persönliche und somit schützenswerte Informationen. Die vorliegende Arbeit setzt genau an diesem Punkt an und untersucht verschiedene Möglichkeiten, wie die Privatsphäre eines Benutzers unter Erhaltung einer möglichst hohen Dienstqualität bei der Verwendung kontext- und ortsbezogener Dienste geschützt werden kann. Ziel ist es zum einen, dem Nutzer effektive Einstellungsmöglichkeiten an die Hand zu geben, mit denen er festlegen kann, mit welchen Parteien er in bestimmten Situationen persönliche Daten teilen möchte oder nicht. Selbst wenn die Preisgabe von Informationen z.B. für die Nutzung eines ortsbezogenen Dienstes unerläßlich ist, sollte der Nutzer zum anderen immer noch individuell entscheiden können, ob ihm eine optimale Dienstqualität oder ein gewisser Grad an zugesicherter Privatsphäre wichtiger ist.

Im Rahmen dieser Arbeit wird gezeigt, dass für die Erreichung dieser Ziele grundsätzlich verschiedene Herangehensweisen zum Einsatz kommen können und dass selbst unter Bereitstellung von sinnvoll verschleierten Kontextinformationen oft weiterhin eine hohe Dienstqualität erzielt werden kann.

1.1 Datenschutz von Kontextinformationen

Die eingangs zitierte Aussage des ehemaligen Google-Chefs ist im selben Maße faszinierend wie befremdlich. Dessen war er sich wohl selbst bewusst, denn nach jedem unheimlichen Fakt liefert er den Hinweis „*mit Ihrer Erlaubnis*" nach.[1] Er drückt damit aus, dass ja niemand gezwungen werde, seine persönlichen Daten in diesem Maße preiszugeben – und es dennoch viele tun.

Ob sich jedoch alle Nutzer die Nutzungsbedingungen jeder installierten App durchlesen oder die angeforderten Rechte prüfen, ist fraglich. In vielen Fällen führt eine Ablehnung z.B. der Standortfreigabe gemäß des „Friss-oder-Stirb"-Prinzips zudem zur absoluten Unbrauchbarkeit einer Anwendung.

In *1984* hat George Orwell eine kontinuierliche, individuelle Überwachung der Bürger als staatlich aufgezwungene Dystopie vorhergesehen. Viele Nutzer von Smartphones und Wearables tragen die dazu notwendigen technischen Hilfsmittel stattdessen heute freiwillig bei sich. Zudem haben sie sich auch noch auf eigene Rechnung damit ausgestattet.

[1] Interessanterweise wird dieser Nachsatz in allen auffindbaren schriftlichen Zitaten über das geführte Interview, wohl der Dramatik zuliebe, weggelassen. Die hier zitierte Version ist der original Videoaufnahme entnommen, das unter der angegebenen Quelle vorliegt.

Dem Thema Datenschutz wird von Politik und Wirtschaft, Medien und Bürgerrechtlern seit einigen Jahren große Aufmerksamkeit geschenkt. Zwischen der EU und den USA wurde jüngst das *Safe Harbor*-Abkommen abgeschafft und das sogenannte *Privacy Shield* errichtet. Ob dabei nutzerfreundliche Entscheidungen getroffen wurden, wird in Frage gestellt [121]. Angela Merkel bezeichnet die Unmenge an täglich über jeden von uns anfallenden Informationsfetzen als „Rohstoff des 21. Jahrhunderts" [140]. Auch ihre mitunter belächelte Aussage, das Internet sei für uns alle „Neuland" [155], trifft zu: Wer kann wirklich abschätzen, was mit den gesammelten Daten passiert? In wessen Hände sie gelangen, mit welchen anderen Datenquellen sie verbunden werden und wofür sie eines Tages ggf. verwendet werden? Wurde nicht jedes eroberte „Neuland" erst einmal ausgebeutet, bevor man sich möglicher Folgen gewahr wurde?

Beispiele für den Missbrauch solcher über die Smartphones der Nutzer erhobenen Daten gibt es viele: In London beobachten Mülleimer die Bewegungen der Bürger durch die Stadt mittels integrierter WLAN-Basisstationen [99]. Der Navigationsanbieter *TomTom* wurde dabei ertappt, die Fahrtenlogs seiner Benutzer an die niederländische Regierung zu verkaufen, die damit lukrative Positionen für Blitzeranlagen ermitteln wollte [113]. Die Standortdaten, die von dem populären Smartphone-Spiel *Angry Birds* für die Ausspielung ortsbezogener Werbung mehrmals pro Minute [76] gesammelt werden, werden von NSA und GCHQ mitgelesen und ausgewertet [230]. Derselbe Artikel zitiert ein *geleaktes* internes Dokument des britischen Geheimdienstes über die Auswertung der Datenströme mobiler Anwendungen, in dem es heisst:

> „*Das bedeutet gewissermaßen, dass jeder, der Google Maps auf seinem Smartphone nutzt, an der Unterstützung eines GCHQ-Systems arbeitet.*" [230]

Die Enthüllungen durch Edward Snowden legen offen, was davor nur befürchtet wurde. Die Dunkelziffer ähnlich gelagerter Fälle dürfte hoch sein, denn welches Unternehmen, das mit den persönlichen Daten von Nutzern in Berührung kommt, gibt Absprachen und Verpflichtungen mit Dritten oder ungewollten Datenverlust durch innere oder äußere Einflüsse gerne offen zu – gesetzt den Fall, dass diese überhaupt bemerkt werden?

Welche Daten gesammelt werden und wie diese aussehen, ist vielfältig. Gemein scheint allen Formen von digitalen Information über Menschen und deren Verhalten jedoch zu sein, dass jeder, insbesondere Regierungen und Wirtschaft, so viele davon wie möglich in ihren Besitz bringen möchten. Die Anbieter, die sich über den Handel mit diesen Daten finanzieren, bleiben meist bewusst im Hintergrund und sind weitgehend unbekannt [51].

Auf der anderen Seite stehen die eigentlichen Eigentümer der Daten, die Smartphone-Benutzer, die sich dem Wert ihrer Daten erst noch bewusst werden müssen. Hier ist es auch Aufgabe der Wissenschaft, Potentiale und Gefahren verständlich zu machen, die aus der Preisgabe z.B. von Standortinformationen entstehen [19]. Zudem müssen Mittel gefunden werden, die dem Nutzer informierte Entscheidungen ermöglichen und echte Wahlmöglichkeiten bieten.

Obwohl sich im Alltag Viele scheinbar kaum dafür interessieren, welche Daten von wem über sie erhoben werden, finden sich das unerlaubte Sammeln persönlicher Daten durch Unternehmen sowie durch den Staat im Jahr 2015 nach Korruption und Cyber-Terrorismus unter den Top-5-Ängsten der amerikanischen Bevölkerung [42]. [235] zeigt, dass sich 91% der US-Bevölkerung gegen den Gedanken sträuben, dass ohne ihr Wissen zu kommerziellen Zwecken Daten über sie gesammelt werden. 72% gehen davon aus, dass die Informationen, die über sie erhoben werden, negative Auswirkungen haben können.

Marketing-Treibende interpretieren das scheinbare Desinteresse gerne so, die Bürger würden den Tausch „Persönliche Daten gegen mögliche Rabatte" gut finden. Laut [235] ist diese Interpretation jedoch grundlegend falsch und auf lange Sicht gefährlich, was das Vertrauen der Menschen in Unternehmen betrifft. Stattdessen liegt eine Resignation der Konsumenten vor, die sich ohnmächtig der Datensammel-Wut ergeben, während sie sie eigentlich fürchten.

Spätestens seit dem ausgehenden 19. Jahrhundert setzt sich die akademische Welt mit dem Begriff der Privatsphäre auseinander. Als Meilenstein gilt der Aufsatz des Anwalts und Richters Louis Brandeis aus dem Jahr 1890, *The Right To Privacy* [246], in dem er „*the right to be let alone*" formuliert. Ausgangspunkt für die Forderung nach einem Recht auf Privatsphäre war auch damals eine neue Technologie: Zeitungen hatten gerade damit begonnen, Klatschseiten mit Fotografien von Personen abzudrucken, die ohne deren Wissen aufgenommen wurden – ein Problem, das es vor der neuartigen „*instantaneous Photography*" so noch nie gab. Brandeis argumentiert, dass sich Technologien stets weiterentwickeln und es Gesetze und Gesellschaft daher auch müssen [246].

Auch in modernen Online-Diensten, die Kontextinformationen verarbeiten, stellt der Schutz der Privatsphäre ein elementares Problem dar, um das sich Wissenschaft, Gesellschaft und Politik gleichermaßen sorgen sollten. In Deutschland besteht seit dem sog. *Volkszählungsurteil* des Bundesverfassungsgerichts von 1983 das *Recht auf informationelle Selbstbestimmung*. Hier wird jedem Menschen das Grundrecht garantiert, „*unter den Bedingungen der modernen Datenverarbeitung* [...] *grundsätzlich selbst über die Preisgabe und Verwendung seiner persönlichen Daten zu bestimmen.*"[2]

Konfrontiert mit der Menge an Standortinformationen, die im Rahmen ihrer normalen Smartphone-Nutzung an externe Parteien übermittelt werden, zeigt sich der Großteil der Nutzer überrascht und würde einer derartigen Datensammlung nicht zustimmen, wäre man sich deren Ausmaßes bewusst [17]. Jedoch mangelt es sowohl an Wissen über derartige Datenströme als auch an effektiven und flexiblen Kontroll- und Einstellungsmöglichkeiten.

Grundsätzliche Ziele einer privatsphäreverträglichen Nutzung kontextbezogener Dienste wurden bereits in zahlreichen Vorarbeiten formuliert und haben sich etabliert. Viele Ansätze treffen bei der Konzeption ihrer Schutzmechanis-

[2]BVerfG, Urteil v. 15. Dezember 1983, Az. 1 BvR 209, 269, 362, 420, 440, 484/83

men jedoch Annahmen, die angesichts realer Entwicklungen korrigiert werden müssen, um in der Praxis einsetzbare Verfahren zu erhalten. Diese grundlegenden Überlegungen werden im nächsten Abschnitt kurz erläutert.

1.2 Ziele der vorliegenden Arbeit

Die breite und dauerhafte Akzeptanz kontextabhängiger Anwendungen ist nur unter Berücksichtigung individueller Datenschutz-Bedürfnisse möglich. In der vorliegenden Arbeit sollen daher neue Ansätze entwickelt werden, wie sich derartige Dienste privatsphärekonform nutzen lassen, ohne die Kontrolle über die eigenen Daten auf- oder in fremde Hände geben zu müssen. Im anzustrebenden Idealfall soll erreicht werden, dass man als Nutzer personalisierter, kontextbezogener Anwendungen stets informiert, effektiv und feingranular über die Freigabe, Präzision und Richtigkeit seiner persönlichen Daten bestimmen kann.

Die vorliegende Arbeit beschäftigt sich mit technischen Möglichkeiten zum Schutz dieser persönlichen Daten und dabei v.a. mit den folgenden Fragen:

1. Wie kann dem Nutzer ein ausdrucksstarker Mechanismus zur effektiven, individuellen und situationsabhängigen Verwaltung seiner Kontextinformationen an die Hand gegeben werden?

2. Wie können sich ortsbezogene Dienste sinnvoll nutzen lassen, ohne dass ein Dienstnutzer seine Privatsphäre völlig aufgeben muss und stattdessen ein messbarer Grad an Anonymität eingehalten werden kann?

Viele Vorarbeiten gehen davon aus, dass zur Umsetzung einer privatsphärekonformen Dienstnutzung eine vertrauenswürdige dritte Partei (engl. *Trusted Third Party*, TTP) existiert, die sich stellvertretend für ihre Nutzer um die Durchsetzung von Privatsphärepräferenzen oder die Anonymisierung der Kommunikation kümmert. Eine solche Komponente verhält sich per Definition nicht bösartig, sondern existiert rein zu dem Zweck, die Privatsphäre ihrer Nutzer zu schützen.

In der Frühphase ortsbezogener Dienste, welche sich grob um das Jahr 2000 einordnen lässt, lag die Einführung einer solchen Komponente nahe. Neben dem Schutz der Privatsphäre konnte damit auch weiteren Problemen des mobilen Rechnens entgegengewirkt werden: Leistungsschwache mobile Endgeräte und gleichermaßen teure wie schmalbandige mobile Datenverbindungen. Nicht nur für den Schutz der Privatsphäre, sondern allgemein für die effiziente Umsetzung ortsbezogener Dienste empfahl es sich in diesem Szenario, so viele Aufgaben wie möglich an leistungsstarke, zentrale Komponenten auszulagern.

Im Rahmen der vorliegenden Arbeit soll jedoch insbesondere aus den folgenden Gründen nicht auf eine solche Komponente zurückgegriffen werden:

- Eine TTP stellt grundsätzlich einen Single-Point-of-Failure und somit ein lukratives Angriffsziel sowie – erfahrungsgemäß – einen wahrscheinlichen Kandidaten für Datenlecks dar [202].
- Es scheint fraglich, welches kommerzielle Interesse daran besteht könnte, eine solche Komponente zu betreiben. Verursacht die Inanspruchnahme Kosten für den Nutzer, sinkt die Nutzungsbereitschaft.
- Moderne Smartphones sind leistungsfähig genug, Verschleierungsmechanismen lokal auszuführen. Die existierenden mobilen Ökosysteme sehen beim Zugriff auf Kontextdaten zudem keine Vermittlung durch einen zentralen Server vor, sondern die lokale Installation von Apps, die über die API des Betriebssystems direkt auf diese Informationen zugreifen.

Im Folgenden wird davon ausgegangen, dass nur das mobile Endgerät des Nutzers eine vertrauenswürdige Instanz darstellt, die sich jederzeit selbst um die Freigabe oder ggf. um die Verschleierung von Kontextinformationen kümmert. Jeder externen Partei sollen nur so viele Details zur Verfügung gestellt werden, wie unbedingt nötig. Aus diesem Grund werden durchwegs Verfahren angestrebt, die nur das Endgerät des Nutzers als vertrauenswürdig ansehen und sich nicht auf die Existenz einer TTP verlassen.

Des Weiteren wird in vielen Arbeiten angenommen, dass der Dienstanbieter spezielle Schnittstellen bereitstellt und serverseitige Anpassungen vornimmt, um selbst verschleierte Anfragen korrekt beantworten zu können. Oft geht dies mit erheblichen Mehraufwand für den Anbieter einher und hat in der Praxis keinen Einzug gehalten. Es wird daher angenommen, dass nur solche Verfahren praktikabel sind, die keine Modifikationen auf Seite des Dienstanbieters voraussetzen und sich stattdessen an den standardmäßig verfügbaren Dienstschnittstellen orientieren. Im Idealfall merkt dieser nicht einmal, ob ein Nutzer verschleierte oder unverfälschte Daten an ihn übermittelt. Im Gegenzug ist davon auszugehen, dass die Dienstqualität eines kontextbezogenen Dienstes unter der Verwendung von verschleierten Informationen leidet. Den dabei auftretenden Trade-Off gilt es zu ermitteln, um den Nutzer eine Abwägung zwischen Quality-of-Service (QoS) und Anonymität zu ermöglichen.

Zudem wird angenommen, dass der Nutzer grundsätzlich daran interessiert ist, kontextabhängige Anwendungen und ortsbezogene Dienste zu nutzen, jedoch mit einem Detailgrad der Informationspreisgabe, der ihm in der aktuellen Situation angemessen erscheint. Zudem zeigt er – wenn dies seine Privatsphärebedürfnisse nicht gefährdet – auch die Bereitschaft, hochwertige Kontextinformationen zur Verfügung zu stellen, z.B. im Rahmen von partizipativen Sensornetzen oder Crowdsourcing-Anwendungen.

Diese Bedingungen stellen realitätsnahe Annahmen dar, an denen sich neue Verfahren zum Schutz der Privatsphäre orientieren sollen. Im Rahmen dieser Arbeit werden drei verschiedene Ansätze vorgestellt, die unter diesen Voraussetzungen eine privatsphärekonforme Nutzung kontext- und ortsbezogener

Dienste ermöglichen: So wird einerseits ein umfassendes Framework für die Verwaltung von Kontext auf einem mobilen Endgerät vorgestellt. Dieses übernimmt eine zentrale Rolle und kümmert sich um die Akquise, Verschleierung und Verwaltung von Kontextinformationen im Allgemeinen. Zum Zweiten wird die privatsphäreschonende Umsetzung eines populären ortsbezogenen Dienstes, die online Routenplanung, präsentiert und evaluiert. Als drittes wird ein neues Verfahren zur kontinuierlichen, topologiebezogenen Standortverschleierung entwickelt, dass unterschiedlichen Diensten gegenüber einen individuell einstellbaren Grad an *Location Privacy* gewährleistet.

1.3 Vorveröffentlichungen

Die in dieser Arbeit beschriebenen Verfahren und Erkenntnisse beruhen zum Teil auf im Vorfeld international veröffentlichten Konferenz- und Journalbeiträgen. Diese bereits erfolgten Publikationen sowie der persönliche Beitrag des Autors an den jeweiligen Veröffentlichungen werden im Folgenden kurz beschrieben. Zu Beginn jedes Kapitels, in dem die jeweiligen Inhalte verwendet werden, erfolgt noch einmal eine dedizierte Auflistung der bereits publizierten Ergebnisse sowie neuer Inhalte.

Bei allen der nachfolgend aufgeführten Veröffentlichungen, bei denen Frau Prof. Dr. Linnhoff-Popien als Lehrstuhlinhaberin und Doktormutter des Autors mitgewirkt hat, fand dies stets in einer beratenden Funktion und durch wertvolles Feedback zu den Inhalten der Arbeiten statt.

[66] skizziert einen Lösung für die privatsphärezentrische Modellierung von Kontextinformationen auf einem mobilen Endgerät. Die Erarbeitung der darin vorgestellten Zielsetzungen und Lösungsansätze sowie die Einordnung in die Literatur sind dem Autor der vorliegenden Arbeit anzurechnen. Sebastian Feld stand als Diskussionspartner hinsichtlich verschiedener Umsetzungsdetails zur Verfügung und gab hilfreiche Hinweise aus Sicht der IT-Sicherheit. Dr. Stephan Verclas hat auf Basis seiner Industriekenntnis einzelne Aspekte des Konzepts in Bezug auf ihre Einsetzbarkeit für Geschäftsanwendungen bewertet.

In [65] wurde dieser Ansatz zu einem umfassenden System zur Verwaltung von Kontextinformationen ausgebaut und insbesondere um die technischen Details der Modellierung und Zugangsverwaltung ergänzt. Auch hier sind alle grundsätzlichen Lösungskonzepte auf den Autor der vorliegenden Arbeit zurückzuführen. Sebastian Feld hat erneut mit einer kritischen Betrachtungsweise Probleme erkannt und durch Diskussion zu deren Lösung beigetragen. [66] und [65] bilden zusammen die Grundlage für Kapitel 3.

In [68] wird *PrOSPR* vorgestellt – eine clientseitige, privatsphäreschonende Umsetzung der online Routenplanung. Die Formulierung der Problemstellung, die Konzeption der Gesamtlösung sowie der Großteil der Umsetzungdetails (ca. 90%) stammen vom Autor der vorliegenden Arbeit. Die Anteile der weiteren Autoren der Publikation lassen sich wie folgt einordnen: In Zusammenarbeit mit Kevin Wiesner entstand das in [68] eingesetzte Verfahren zur Erzeugung

der Verschleierungszonen, das in [253] für die Herstellung von Privatsphäre in partizipativen Sensornetzen konzipiert wurde. Darüber hinaus war Kevin Wiesner als Diskussionspartner an der Auswertung der Ergebnisse beteiligt. Marco Maier wirkte bei der Recherche und Aufbereitung verwandter Arbeiten mit. Michael Schuster half bei der Implementierung der verwendeten Evaluationsumgebung. [68] stellt den Ausgangspunkt für Kapitel 4 dar.

1.4 Struktur der Arbeit

In Kapitel 2 werden zunächst die Grundlagen kontext- und ortsbezogener Anwendungen beschrieben. Dazu zählen eine Definition des Kontextbegriffs sowie eine kurze Übersicht über Verfahren zur Kontexterkennung auf mobilen Endgeräten. Darüber hinaus werden zentrale Konzepte aus dem Forschungsfeld der Privatsphäre in kontext- und ortsbezogenen Diensten erläutert sowie ein aktueller Überblick über den Stand der Wissenschaft gegeben.

Angesichts der Fülle an Techniken zum Schutz der Privatsphäre aus der Literatur und der Individualität von Privatsphärepräferenzen wird in Kapitel 3 ALPACA vorgestellt. Dabei handelt es sich um einen generischen Ansatz zur privatsphärezentrischen Modellierung und Verwaltung von Kontextinformationen auf mobilen Endgeräten. Im Gegensatz zu existierenden Kontextmodellen und bestehenden Systemen zur Kontextverwaltung zeichnet sich ALPACA dadurch aus, dass es stets dem *Privacy-by-Design*-Prinzip [40] folgt. So werden durchgängig Konzepte zum Schutz persönlicher Daten gemäß individueller Privatsphärepräferenzen integriert.

Das darauffolgende Kapitel 4 beschäftigt sich mit speziell mit dem Schutz von Standortinformationen im Rahmen ortsbezogener Dienste. Zunächst wird dabei eine Lösung für die privatsphäreschonende Nutzung von online Routenplanern entworfen, die für ihre Diensterbringung stets exakte und aktuelle Standortdaten benötigen. Im Rahmen einer umfangreichen Evaluation auf echtem Kartenmaterial wird untersucht, wie der Trade-Off zwischen Privatsphäre, Dienstqualität und Kommunikationsaufwand ausfällt und welchen Einfluss verschiedene Optimierungsmöglichkeiten haben.

Aufbauend auf den gewonnenen Erkenntnissen wird schließlich ein neuer Ansatz für die Verschleierung von Standortinformationen bei der Nutzung kontinuierlicher ortsbezogener Dienste entwickelt, der neben Informationen über semantische Orte auch das zugrundeliegende Straßennetz berücksichtigt. Hierbei entsteht eine allgemein einsetzbare Lösung für die clientseitige Vermeidung ortsbasierter Inferenzangriffe, die sich für unterschiedlichste Ausprägungen ortsbezogener Dienste nutzen lässt.

In Kapitel 5 werden die vorgestellten Verfahren und Ergebnisse zusammengefasst. Es wird ein Ausblick auf mögliche künftige Forschungsarbeiten gegeben und es werden Fragestellungen formuliert, die unmittelbar an die Ergebnisse der vorliegenden Arbeit anknüpfen.

2 Grundlagen kontextbezogener Dienste und bestehende Schutzmechanismen

Dieses Kapitel erläutert Grundlagen und Begriffe, die für das Verständnis kontext- und ortsbezogener Anwendungen von Bedeutung sind. Dazu zählt zum einen eine formale Definition des Kontextbegriffs, wie er im Rahmen der vorliegenden Arbeit verwendet wird sowie zum anderen eine grundlegende Beschreibung und mögliche Einteilung kontext- und ortsbezogener Dienste in verschiedene Anwendungsklassen. Weitere Grundlagen stellen die unterschiedlichen Herangehensweisen und Möglichkeiten zur Gewinnung von Kontext- und Standortinformation auf mobilen Endgeräten dar. Anhand dieser Verfahren wird exemplarisch gezeigt, welch Vielzahl an Kontextinformationen sich bereits heute allein mit Hilfe eines Smartphones ermitteln lässt.

Der Schutz der Privatsphäre bei der Verwendung kontext- und ortsbezogener Dienste stellt das Kernthema der vorliegenden Arbeit dar. Daher werden anschließend etablierte Konzepte und Schutzmechanismen aus der Literatur vorgestellt, die den Stand der Wissenschaft wiedergeben und eine Einordnung der vorliegenden Arbeit ermöglichen sollen. Insbesondere auf das breite Feld der Anonymität und Privatsphäre bei der Preisgabe von Standortinformationen wird detailliert eingegangen.

2.1 Definitionen und Grundlagen

Bereits im Jahr 1991 hat Mark Weiser, ein Visionär auf dem Gebiet des *Ubiquitous Computing* – auf Deutsch in etwa das „allgegenwärtige Rechnen" – seine Vorstellung vom Computer des 21. Jahrhunderts festgehalten [248]. Der Rechner als physische Maschine, mit der ein Nutzer explizit interagiert, tritt dabei in den Hintergrund, geht durch Miniaturisierung und unsichtbare Vernetzung quasi in der Umgebung auf und umgibt den Menschen mit einer allgegenwärtigen, künstlichen Intelligenz, um ihn bei seinen Aufgaben zu unterstützen.

25 Jahre später ist dieser Zustand zwar noch nicht erreicht, mit leistungsstarken mobilen Endgeräten und quasi wegfallenden Beschränkungen bei der Bandbreite mobiler Internetverbindungen ist man dieser Vision jedoch schon ein gutes Stück näher gekommen. Hinzu kommen immense Fortschritte in der künstlichen Intelligenz, Sprach- und Bilderkennung sowie der mittlerweile produktreifen Klasse der sog. Wearables und dem wachsenden Internet-of-Things.

Wichtige Voraussetzung für die Realisierung dieser Vision ist, dass Mensch und Computer anders miteinander kommunizieren, als das bisher der Fall war. Im Gegensatz zu zwischenmenschlicher Kommunikation fehlen bei der klassischen Interaktion mit einem Computer über Maus, Tastatur und Bildschirm nämlich einige wichtige, nonverbale Komponenten. Anders als ein menschlicher Kommunikationspartner weiß ein Computer üblicherweise nichts über den *Kontext*, in dem diese Kommunikation stattfindet. Alle relevanten Informationen müssen stattdessen explizit angegeben werden, was diese Kommunikation vergleichsweise mühsam gestaltet.

2.1.1 Der Kontextbegriff

In diesem Abschnitt soll geklärt werden, was unter dem abstrakten Begriff „Kontext" bzw. unter kontextabhängigen Anwendungen zu verstehen ist. Dafür gibt es in der Wissenschaft unterschiedliche Definitionen, die über die Zeit hinweg immer wieder leicht umgeformt und – jeweils neuen Entwicklungen Rechnung tragend – angepasst oder gemäß der individuellen Sichtweise verschiedener Autoren uminterpretiert worden sind [174].

Ziel des kontextabhängigen Rechnens ist es, über Sensoren und Inferenzmechanismen so viele möglicherweise bedeutsame Informationen wie möglich zu sammeln, um auf Basis deren Interpretation die Kommunikation zwischen Mensch und Maschine intuitiver und intelligenter zu gestalten. Möglichst viele Abläufe sollen dadurch automatisiert werden. Eine überaus plakative, aber wenig aussagekräftige Definition von Kontext von Roussaki et al. lautet daher:

> „*Everything can be considered to be context information.*" [203]

Während diese Aussage zwar eindrucksvoll die nur durch technische Unzulänglichkeiten nach oben hin beschränkte Fülle von Kontextinformationen umschreibt, ist sie wenig geeignet für eine wissenschaftliche Auseinandersetzung.

Die wohl berühmteste und am häufigsten zitierte Definition von Kontext ist die von Abowd und Dey, in der Kontext wie folgt formalisiert wird:

> „*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*" [2]

Somit handelt es sich bei Kontext allgemein um jedes Stück an Information, das dazu verwendet werden kann, die Situation einer Entität zu beschreiben, sofern diese Entität relevant für die Interaktion des Nutzers mit einer Anwendung ist.

Während diese Definition auf der einen Seite zwar immer noch sehr generisch ist, schränkt sie auf der anderen Seite stark ein, was als Kontext zu verstehen ist. Unterschiedliche Arbeiten kritisieren und erweitern diese Definition.

Strassner et al. bemängeln die fehlende inhaltliche Prägnanz dieser Formulierung [228]. So legen die Autoren Wert darauf, zu betonen, dass Kontext sowohl

gemessene Fakten als auch inferiertes Wissen beinhaltet, und dass sowohl aktuelle als auch bereits vergangene Zustände zu den Kontextinformationen einer Entität gehören. Ihre Definition lautet daher:

> „*The Context of an Entity is a collection of measured and inferred knowledge that describe the state and environment in which an Entity exists or has existed.*" [228]

Was hierbei nicht berücksichtigt wird, ist die Möglichkeit, dass Kontext auch in der Zukunft liegende Zustände beschreiben kann: Kalendereinträge verraten, wann der Nutzer sich mit gewissen Personen trifft, historische Bewegungsdaten und aktuelle Routenanfragen verraten, wohin er sich in Zukunft bewegt.

Im Vergleich zur Kontextdefinition von Abowd fällt hierbei jedoch eine aus Sicht der vorliegenden Arbeit störende Einschränkung weg, die auch von Strassner scharf kritisiert wird: Der Hinweis, dass eine Relevanz zur Interaktion des Benutzers mit einer Anwendung bestehen muss, ist vor dem Hintergrund proaktiver, im Hintergrund laufender Dienste schlichtweg nicht zutreffend.

Stattdessen existiert Kontext auch dann und wird von Sensoren ermittelt, wenn der Nutzer überhaupt nicht mit einer Anwendung oder seinem Endgerät interagiert. Insbesondere sind aus Sicht der Privatsphäre auch solche Informationen über die Situation des Nutzers schützenswert, die für die Interaktion mit einer Anwendung keinerlei Bedeutung haben.

Dieser „Interaktions-irrelevante" Kontext, der auf einem mobilen Endgerät u.U. zur Verfügung steht oder akquiriert werden kann, wird von der Definition von Abowd jedoch nicht abgedeckt. Abowd hat ganz klar einen auf Funktionalität und Interaktivität ausgerichteten Blickwinkel. Nachdem die vorliegende Arbeit jedoch nicht auf die Bereitstellung der aus funktionaler Sicht nötigen Kontextinformationen abzielt, sondern auf einen größtmöglichen Schutz dieser Informationen, kann diese Definition nicht verwendet werden.

Sozusagen als Obermenge von Kontext gem. der Definition von Abowd und Dey formalisiert Thomas Strang die Unterscheidung zwischen Kontextinformation, Kontext und *Situation*, um genau diese Abgrenzung von Informationen hinsichtlich ihrer Relevanz für eine Anwendung oder Aufgabe zu ermöglichen [225]: Während Kontext die Menge an Kontextinformationen ist, die – wie bei Abowd – eine Entität beschreiben, die für eine Aufgabe relevant ist, ist eine Situation für ihn die Menge an allen bekannten Kontextinformationen.

Aus der Kombination aller dieser Betrachtungsweisen und dem dieser Arbeit zugrundeliegenden Fokus auf den Schutz der Privatsphärebedürfnisse eines Nutzers, ergibt sich für die folgenden Kapitel o.B.d.A. folgende Definition von Kontextinformationen:

Definition 1:
„*Kontextinformationen sind alle Informationen, die von Sensoren gemessen, von Software inferiert, über das Netzwerk bezogen oder durch Eingabe des Nutzers auf dessen mobilem Endgerät zur Verfügung stehen und dazu beitragen*

können, einzelne Aspekte der Person oder einer früheren, aktuellen oder zukünftigen Situation des Nutzers unabhängig von womöglich stattfindender Interaktion mit dem Endgerät zu beschreiben."

Der Begriff der Situation bezieht sich dabei auf die Definition von Strang [225]. Auf die Tatsache, dass Kontextinformationen auch manuell vom Nutzer eingegeben werden können, wie z.B. seine Identität oder Kalendereinträge, weist wiederum bereits Abowd hin [2], der in der Forschung eine deutliche Tendenz einer ausschließlichen Fokussierung auf implizit ermittelte Kontextinformationen sieht. Vor dem Hintergrund des Privatsphäreschutzes kann damit grundsätzlich jede Kontextinformation ein schützenswertes Datum darstellen. Die Entscheidungshoheit darüber sollte stets dem Nutzer selbst obliegen.

Beispiele für verschiedene Typen von Kontextinformationen sind vielfältig in der Literatur zu finden. Gängige Vertreter – und jene, die in der Praxis auch mit Abstand am häufigsten zum Einsatz kommen – sind *Ort*, *Identität*, *Aktivität* und *Zeit* [2]. Daran hat sich auch in den 15 Jahren seit Veröffentlichung von Abowds und Deys wegweisender Arbeit kaum etwas verändert.

Schmidt et al. [214] weisen jedoch darauf hin, dass es darüber hinaus eine Fülle an weiteren Informationen gibt, die für die Umsetzung kontextabhänger Anwendungen berücksichtigt werden sollten. Dazu zählen *menschliche Faktoren* wie die Gewohnheiten und Interessen des Nutzers, seine Emotion und Vitalwerte, seine *sozialen Beziehungen* und die *räumliche Nähe* zu anderen Personen sowie seine nächsten *Aufgaben und Ziele*. Darüber hinaus Informationen wie nahe Peripheriegeräte bzw. allgemein die *Eigenschaften seiner Umwelt* wie Wetter, Helligkeit, Temperatur oder Umgebungslautstärke, sichtbare Mobilfunkmasten und WLAN-Netzwerke. Hinzu kommen Eigenschaften, die das *Endgerät* selbst betreffen wie die Bildschirmausrichtung, welche Anwendung gerade im Vordergrund läuft oder der verbleibende Akkustand.

Gustarini et al. [109] zeigen, dass auch der gefühlte Grad an *Intimität* eine wichtige Kontextinformation darstellt. Diese den menschlichen Faktoren zuzurechnende Kontextinformation sei ausschlaggebend für die Art, wie ein Nutzer mit seinem Smartphone und einzelnen Anwendungen interagiert.

Für eine Übersicht über verschiedene Klassifikationsmöglichkeiten von Kontextinformationen sei z.B. auf den Survey in [193] verwiesen.

2.1.2 Kontext- und ortsbezogene Dienste

Anders als bei der Definition des Kontextbegriffs herrscht hinsichtlich der Beschreibung kontextabhängiger Systeme weitgehend Einigkeit. Auch hier ist es Dey, der die am weitesten verbreitete Definition liefert:

> „*A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.*" [60]

Im Gegensatz zu seiner Definition von Kontext selbst ist der Hinweis auf die Relevanz der von einer kontextabhängigen Anwendung gelieferten Informationen bzw. geleisteten Dienste an dieser Stelle zweifellos gerechtfertigt. Ein Dienst, der irrelevante Informationen liefert oder gegen den Nutzerwillen arbeitet, verfehlt seine Aufgabe schließlich.

Kontextabhängige Anwendungen adaptieren ihre erbrachten Informationen und Dienstleistungen an die jeweilige Situation des Nutzers. Hierzu gehört nach Dey u.a. die Akquise und Darstellung relevanter Informationen, das automatische Ausführen von Prozessen oder auch das Anreichern von Informationen mit kontextuellen Metadaten, die das Wiederauffinden erleichtern sollen. Je nach Ausprägung des Dienstes werden dafür unterschiedliche Kontextinformationen verarbeitet, die für eine Anwendung relevant sind.

Die Möglichkeiten, wie kontextabhängige Anwendungen die Nutzungserfahrung verbessern oder völlig neuartige Dienste bieten können, sind vielfältig. In einigen Fällen kann die kontextabhängige Diensterbringung lokal auf dem Endgerät durchgeführt werden – aus Sicht der Privatsphäre ist das der Optimalfall. So ist es heute Standard, dass sich der Bildschirm des Smartphones auf Basis von Sensordaten mitdreht, wenn das Gerät entsprechend geneigt wird. Etwas intelligenter muten kontextabhängige Anwendungen wie die *Llama App*[1] an, die z.B. das Lautstärkeprofil des Telefons kontextabhängig regulieren.

Die meisten kontextabhängigen Dienste entfalten ihre volle Stärke jedoch in der Kommunikation mit externen Parteien. So kann ein proaktives Allzweck-Werkzeug wie *Google Now*[2] den Terminkalender des Nutzers und die aktuelle Verkehrslage im Auge behalten, um verspätetes Losfahren zu verhindern. Ein weiteres Beispiel sind Buddy-Tracker, die darüber benachrichtigen, wenn sich einer meiner Freunde in der näheren Umgebung befindet [5]. LiKamWa et al. [164] schlagen ein soziales Netzwerk vor, in dem Nutzer ihre aktuelle Stimmungslage miteinander teilen, die automatisiert erkannt wird.

Kontextbezogene Systeme sind dabei in nicht nur im privaten, sondern auch im professionellen Umfeld zu beobachten. So entstehen kontextabhängige Geschäftsanwendungen wie die *Smart Business and Enterprise Environments*-Plattform *SmartBEEs* [67]. Diese wurde von Dorfmeister et al. im Rahmen einer Kooperation mit T-Systems für eine große Handelskette entwickelt. Die Plattform sammelt kontinuierlich Kontextinformationen der Mitarbeiter und der Umgebung, um auf Basis eines kontextabhängigen Regelwerkes die häufig, aber dynamisch anfallenden Aufgaben im Sinne einer Prozessoptimierung zu priorisieren und an den jeweils geeignetsten Mitarbeiter zu deligieren. Dieser wird über sein mobiles Endgerät darüber informiert, wenn er sich in der Nähe befindet und mit einer Aufgabe beschäftigt ist, die eine Unterbrechung erlaubt.

Eine weitere, relativ junge Klasse kontextabhängiger Anwendungen stellen sog. *kontextzentrische soziale Netze* dar [250]. Hier wird Kontext nicht als Beiwerk gesehen, das die herkömmliche Funktionalitäten online sozialer Netze

[1] `https://play.google.com/store/apps/details?id=com.kebab.Llama`
[2] `https://www.google.com/landing/now/`

(engl. *Online Social Networks*, OSN) ergänzt, sondern als ultimative Quelle für die Herstellung spontaner oder dauerhafter Kanten im sozialen Graph. Dies ermöglicht die Interaktion zwischen Nutzern, die sich gemäß frei wählbarer Kontexttypen in derselben Situation befinden oder befunden haben.

Die am häufigsten verwendete Ausprägung kontextabhängiger Anwendungen sind **ortsbezogene Dienste** (engl. *Location-Based Services*, LBS). Die große Verbreitung leistungsstarker, „ultra-mobiler" Endgeräte mit integriertem GPS-Chip hat maßgeblich zu deren kommerziellen Erfolg beigetragen. Derartige Systeme berücksichtigen mit dem aktuellen Standort des Nutzers meist nur einen einzigen Typ von Kontextinformationen, um ihm – gemäß der obigen Definition – relevante Informationen und Dienste bieten zu können. Dazu zählen sog. *Point-of-Interest*-Suchen (POI) um den Standort des Nutzers, online Routenplanung und Navigation oder ortsbezogene soziale Netze.

Ortsbezogene Dienste können je nach ihrer Funktionalität in verschiedene Kategorien eingeordnet werden (vgl. z.B. [53, 94]). Eine grundsätzliche Unterscheidung ist, ob es sich bei einem LBS um einen privat genutzten Dienst handelt oder um ein professionelles Werkzeug wie das Tracking von Logistik-Vorgängen. Daneben können die einzelnen Klassen entlang vieler Dimensionen verglichen werden. Eine solche Dimension ist z.B. die Häufigkeit einzelner LBS-Anfragen, die von sporadischer Dienstnutzung mit großen Pausen über periodische Anfragen bis hin zu einer kontinuierlichen Übertragung von Standortinformationen an den Dienstanbieter reichen kann.

Ein weiteres Unterscheidungsmerkmal ist der Zweck und relative Zeitpunkt der LBS-Nutzung. So gibt es Dienste, die sich stellvertretend für das Endgerät auf Basis von Netzwerkinformationen um die Ermittlung des aktuellen Standortes kümmern, wohingegen sich andere wie z.B. eine POI-Suche auf die Auslieferung maßgeschneiderter, ortsbezogener Suchergebnisse spezialisieren.

Andere Unterscheidungsmerkmale können sein, ob ein Dienst anonym oder personalisiert genutzt wird, ob es sich um eine interaktive Anwendung handelt, bei der der Nutzer eine unmittelbare Antwort erwartet oder ob es ein Hintergrunddienst ist wie die Klasse der *Location-Based Social Networks* [266] ist, die den Nutzer proaktiv über Ereignisse in seiner Umgebung informieren.

Darüber hinaus gibt es LBS-Anwendungen, die unter den Oberbegriffen des *Participatory Sensing* [50] und *Crowd Sensing* [169] dem einzelnen Nutzer keinen unmittelbaren Mehrwert liefern, sondern im Sinne des Allgemeinwohls Daten ihrer Teilnehmer sammeln, z.B. über Straßenschäden, Lärmbelästigung und Luftverschmutzung oder präzise Informationen über die aktuelle Verkehrslage auf Basis des Standorts und der aktuellen Geschwindigkeit.

Des Weiteren benötigen unterschiedliche LBS für die qualitativ hochwertige Dienstnutzung unterschiedlich hohe Auflösungen von Standortinformationen. Während für den Abruf einer aktuellen Wettervorhersage beispielsweise die Angabe einer Postleitzahl oder eines Städtenamens vollkommen ausreicht, benötigen Routenplaner und Navigationsdienste für eine sinnvolle Diensterbringung eine exakte Standort- und Zielangabe im Ein-Meter-Bereich.

Zu guter Letzt kann auch unterschieden werden, ob es sich um einen herkömmlichen LBS für den Außenbereich oder um eine Anwendung aus dem Feld der *Indoor Location-Based Services* handelt, die ortsbezogene Dienste wie z.B. die Positionierung und Navigation in Gebäuden anbieten [183].

2.1.3 Kontextermittlung auf mobilen Endgeräten

Ein großer Teil der Arbeiten aus dem populären Forschungsfeld der *Context Awareness* mobiler Anwendungen beschäftigt sich mit dem grundlegenden Problem der Kontextermittlung. Letztere stellt naturgemäß den Ausgangspunkt jeder kontextbezogenen Diensterbringung dar und hat die Aufgabe, aus der Vielzahl an zur Verfügung stehenden Datenquellen den Kontext des Benutzers so präzise wie möglich zu erfassen und interpretierbar zu machen.

Die möglichen Kontextquellen reichen von Hardware-Sensoren, strukturierten Daten und manuellen Nutzereingaben bis hin zu komplexen Reasoning-Verfahren, die aus einzelnen Kontextinformationen höherwertige Zusammenhänge schließen können und dadurch weitere Kontextdaten erzeugen oder die Plausibilität der gemessenen Daten prüfen können. Meist werden dabei mindestens zwei Ebenen von Kontext unterschieden, je nachdem, ob eine Kontextinformation direkt messbar war oder sich aus logischer Inferenz über gemessene Kontextinformationen ergibt [193]. So gilt die GPS-Position eines Benutzers z.B. als *low-level* Kontextinformation, wohingegen die räumliche Entfernung zu anderen Nutzern oder die Auflösung der Koordinaten auf eine Adresse oder einen semantischen Ort („Zuhause") sog. *high-level* Kontext darstellen.

Im Folgenden werden einige Beispiele für Verfahren zur Kontexterkennung auf mobilen Endgeräten gegeben. Einen umfassenden Überblick über die Smartphone-basierte Kontextermittlung in Echtzeit bietet der Survey in [125].

2.1.3.1 Aktivitäts- und Situationserkennung

Ein häufig untersuchter Typ von Kontextinformationen ist die Aktivität, die ein Nutzer aktuell ausführt. Grundlegende Aktivitäten, die sehr zuverlässig erkannt werden können, sind Ruhephasen wie Sitzen oder Stehen, Gehen, Rennen, Treppen Auf- und Absteigen [7] oder die Nutzung und Klassifizierung unterschiedlicher Verkehrsmittel [114]. Daneben lassen sich auch komplexere Alltags-Aktivitäten wie Staubsaugen und Zähneputzen [190] oder feingranulare Teilabläufe innerhalb komplexer Gesamtvorgänge aus den Sensordaten des Smartphones extrahieren: In [172] können z.B. 17 verschiedene Aktivitäten innerhalb des Kontexts „U-Bahnfahren" korrekt unterschieden werden.

Der Großteil der Forschungsarbeiten versucht, diese Aktivitäten mit Hilfe von Methoden des maschinellen Lernens aus Sensormesswerten wie z.B. des eingebauten Accelerometers, Gyroskops, Magnetometers oder Mikrofons abzuleiten, welche heute standardmäßig in Smartphones zur Verfügung stehen.

Der grundsätzliche Ablauf der Aktivitätserkennung auf mobilen Endgeräten ist dabei meist derselbe [125]: Zunächst werden über die Sensoren des

Smartphones *gemessene Datenströme* ermittelt, die in einem darauffolgenden *Pre-Processing*-Schritt zu einer Menge an für die Kontexterkennung relevanten *Features* aus Zeit- oder Frequenzdomäne vorverarbeitet werden. Je nach Aktivität, die erkannt werden soll, eignen sich hierfür unterschiedliche Feature-Sets.

Die so entstehenden Feature-Vektoren werden schließlich an Verfahren des maschinellen Lernens zur Klassifikation übergeben. Häufig verwendete Klassifikationsverfahren sind k-nächste-Nachbarn-Suchen, Naive Bayes, Entscheidungsbäume und -wälder oder sog. Support-Vector-Maschinen. Diese Techniken basieren auf dem mitunter aufwändigen „Lernen" eines Klassifikators auf Basis eines annotierten Trainingsdatensatzes, ermöglichen dafür aber die sehr effiziente Klassifizierung von Sensordaten zur Laufzeit, die sich ohne weiteres auf einem heutigen Smartphone durchführen lässt [125].

Im letzten Schritt werden die erkannten Kontextinformationen zur Verwaltung und Reasoning an ein Kontextmodell weitergegeben oder direkt an kontextabhängige Anwendungen und Dienste übermittelt.

Neben der Aktivitätserkennung lassen sich auch weitere Informationen über den Kontext des Nutzers in Erfahrung bringen, die seine aktuelle Situation beschreiben ohne dabei auf bewusste Tätigkeiten oder Bewegungsabläufe abzuzielen. Hierzu zählen neben der Ermittlung biometrischer Daten wie Herzfrequenz und Blutddruck (low-level) z.B. Verfahren, die dazu in der Lage sind, die Stimmungslage und Emotionen des Nutzers (high-level) anhand seiner Interaktionsmuster mit dem Smartphone zu messen [164, 139]. In [164] wird hierbei bei der Unterscheidung von je fünf verschiedenen Stimmungslagen und Aktivitätsleveln eine korrekte Erkennungsrate von 93 % erreicht.

Auch kommerzielle Produkte sind mittlerweile erhältlich, die den eigenen Stresslevel und Gefühlszustand tracken [117]. Diese beruhen auf sog. *Wearables*, die zusätzliche Sensoren für Puls und Hautleitfähigkeit zur Kontexterkennung nutzen, die in heutigen Smartphones i.d.R. noch nicht verbaut sind.

Ein weiterer Typ von Kontextinformationen ist die Nähe zu anderen Benutzern. Wenn diese Nähe naiv auf Basis bekannter Nutzerpositionen berechnet wird (s. nächster Abschnitt), stellt sie eine high-level Kontextinformation dar. Nähe lässt sich aber auch – privatsphäreschonend – ohne die Preisgabe absoluter Standortangaben ermitteln: So beschreibt [173] einen Ansatz, bei dem sich Nähe auf Basis der Ähnlichkeit von ermittelten RSSI-Vektoren mobiler WLAN-Endgeräte in den Umgebungen der Teilnehmer ermitteln lässt.

Die beschriebenen Verfahren lassen sich alle in die Klasse der *opportunistic context recognition* einordnen, in deren Rahmen keine Nutzereingaben nötig sind. Darüber hinaus stellen natürlich auch diese Eingaben eine Quelle für Kontextinformationen dar, bei der viele der erwähnten Klassifikationsaufgaben jedoch wegfallen, da die Daten – wie z.B. ein Kalendereintrag mit Ortsangabe – ja schon in strukturierter Form und mit hoher Präzision vorliegen.

2.1.3.2 Positionsermittlung

Die bislang erfolgreichste und am häufigsten genutzte Ausprägung kontextabhängiger Anwendungen stellen ortsbezogene Dienste dar. Um derartige Angebote zu nutzen, muss das Endgerät des Nutzers seine eigene Position kennen und diese zusammen mit der Spezifikation weiterer Anfrageparameter an den LBS-Anbieter übermitteln. Wie die Positionsermittlung auf einem mobilen Endgerät umgesetzt werden kann, wird im Folgenden erklärt.

2.1.3.2.1 Positionsermittlung im Außenbereich Die gängigste Form der der Outdoor-Positionierung stellt die satellitengestützte Lokalisierung mit Hilfe von *Global Navigation Satellite Systems* (GNSS) dar. Existierende Systeme sind das amerikanische *Global Positioning System* (GPS) oder das russische *GLONASS*. Mit *Galileo* will Europa ein eigenes, auf die europäischen Bedürfnisse abgestimmtes Satellitennavigationssystem etablieren, das ab 2020 zivil nutzbar sein soll [204].

Die GNSS-gestützte Positionierung erfolgt mithilfe der Laufzeitermittlung von Signalen mehrerer sichtbarer Satelliten. Auf Basis bekannter Satellitenpositionen lässt sich eine dreidimensionale Position für das Empfangsgerät berechnen. Die Positionen der Satelliten lassen sich neben dem Zeitpunkt des Absendens den Signalen selbst entnehmen. Um unvermeidbare Fehler in der Uhrensynchronisation des mobilen Endgeräts zu begegnen, sind zur korrekten Positionierung mindestens vier Satelliten notwendig. Prinzipiell würden drei Satelliten genügen, um durch die beiden Schnittpunkte deren durch den ermittelten Abstand gegebenen Sphären einen Standort in Erdnähe zu bestimmen. Da der Empfänger jedoch keine qualitativ hochwertige Uhr besitzt, kann er die Laufzeiten und Abstände nicht korrekt berechnen. Aufgrund des Fehlers in seiner lokalen Zeitmessung kann der Empfänger somit keinen eindeutigen Schnittpunkt vier solcher Sphären ermitteln. Durch schrittweises Nachstellen der lokalen Uhr kann diese Ungenauigkeit korrigiert werden bis sich ein Schnittpunkt ergibt, der als Standort zurückgegeben wird. [247]

Daneben gibt es verschiedene andere Positionierungsverfahren für den Außenbereich. Das Mobilfunknetz ermöglicht eine *zellbasierte Lokalisierung*, die – je nachdem, wo man sich befindet – jedoch relativ ungenau sein kann. Durch die Einteilung einer Zelle in gerichtete Sektoren und die laufzeitbasierte Ermittlung der Distanz eines Endgeräts zum Sendemast kann diese Region allerdings deutlich eingegrenzt werden [188]. Darüber hinaus besteht die Möglichkeit, auf Basis mehrerer Funkmasten eine Triangulation durchzuführen [259].

Eine alternative Herangehensweise besteht in der Verwendung von Datenbanken, die durch *Wardriving* oder *Crowdsourcing* ermittelte Informationen über die ortsabhängige Sichtbarkeit von WLAN-Netzwerken speichern. Hierfür ermittelt das mobile Endgerät einen sog. *Fingerprint* aus sichtbaren Netzwerk-ESSIDs und schickt diesen an einen zentralen Dienstanbieter. Dieser vergleicht den Fingerprint mit seiner Datenbank und liefert – mittlerweile ebenfalls mit hoher Genauigkeit – den Standort des Nutzers zurück. Kommerzielle Anbieter

wie *Skyhook*[3] stellen solche Dienste global zur Verfügung.

Aus Sicht der Privatsphäre sind satellitengestützte Positionierungssysteme vorzuziehen, da hierbei keine externe Partei die Position des Nutzers erfährt, sondern nur sein eigenes Endgerät. Im weiteren Verlauf dieser Arbeit wird daher stets davon ausgegangen, dass solche terminalbasierten Positionierungsmethoden verwendet werden und sich aus dem Vorgang der Standortermittlung selbst somit keine Gefahr für die Privatsphäre des Nutzers ergibt.

2.1.3.2.2 Positionierung im Innenbreich Das gebräuchlichste Verfahren für die Indoor-Lokalisierung ist das sog. *WLAN-Fingerprinting*, bei dem das Endgerät des Nutzers einen Vektor aus den empfangbaren Signalstärken der „sichtbaren" WLAN-Accesspoints erzeugt. Dieser wird mit Hilfe einer sog. *Radio-Map* z.B. gemäß des k-nächste-Nachbarn-Verfahrens auf die plausibelste 3D-Position innerhalb des Gebäudes abgebildet. Daneben existieren *Time-of-Flight*-basierte Positionierungsmethoden auf Basis von WLAN, die nicht die Signalstärken, sondern die Signallaufzeiten zwischen dem Endgerät und mehreren Accesspoints zur Positionierung verwenden. Der Vorteil dieser Systeme ist, dass die meisten Gebäude ohnehin über eine geeignete WLAN-Infrastruktur verfügen, die für die Positionierung verwendet werden kann. Eine umfassende Übersicht über diesen Forschungsbereich bietet der Survey von Liu et al. [166].

Zudem existieren *visuelle Positionierungssysteme*, die auf Basis von Kamerabildern dazu in der Lage sind, den aktuellen Standort eines Benutzers innerhalb eines Gebäudes zu ermitteln [183]. Zu diesem Zweck wird anhand ausgewählter Features die paarweise Ähnlichkeit von zur Laufzeit aufgenommenen Frames mit ortsannotierten Trainingsbildern ermittelt. Wie auch für die Fingerprint-basierte Positionierung wird hierfür eine umfangreiche Datenbank benötigt, die in einer Offline-Phase erzeugt werden muss.

Daneben lässt sich – sowohl für den Innen- und Außenbereich – auch das sog. *Dead-Reckoning* (dt. Koppelnavigation) für die Positionierung oder für die Erhöhung der erreichbaren Positionierungs-Genauigkeit einsetzen. Ausgehend von einer bekannten Startposition wird dabei durch die Zuhilfenahme weiterer Sensoren und ggf. Kartenmaterial versucht, die relativen Bewegungen eines Objekts seit Verlassen der Ausgangsposition nachzuverfolgen. Für Fußgänger werden aus den Accelerometer-Daten einzelne Schritte und die Schrittlänge abgeleitet. Über das Gyroskop können Richtungsänderungen oder über den digitalen Kompass absolute Richtungen erkannt werden [52, 138, 196]. Für Fahrzeuge, deren Bewegungen durch ein Strassennetz beschränkt sind, wird z.B. durch zweimaliges Integrieren der Beschleunigungswerte und der über den Kompass erhaltene Fahrtrichtung die zurückgelegte Strecke auf eine Karte gematcht [21, 86]. Dies funktioniert mitunter sogar ohne Kenntnis der Startposition, wie [111] zeigt.

[3] `http://www.skyhookwireless.com/`

Neben einer möglichst energieeffizienten Kontextermittlung streben all diese Verfahren naturgemäß danach, möglichst hohe Erkennungsraten zu erreichen, um eine qualitativ hochwertige Nutzung kontextabhängiger Dienste zu erlauben. Der Fokus solcher Arbeiten liegt auf Aspekten wie die Reduzierung der Fehlerrate sowie die Maximierung von Präzision, Auflösung und Verfügbarkeit.

Vor dem Hintergrund der Privatsphäre eines Nutzers müssen jedoch andere, teilweise sogar gegensätzliche Zielsetzungen verfolgt werden: So ist es in vielen Fällen der Dienstqualität nicht wirklich abträglich, wenn keine hochqualitativen, detaillierten Kontextinformationen zur Verfügung gestellt werden, sondern u.U. nur solche, die eher vage, ungenau oder sogar falsch sind.

Während eine Qualitätsreduzierung der verwendeten Kontextinformationen oft noch eine ausreichende Dienstqualität ermöglicht, schützt sie zugleich die Privatsphäre des Benutzers. Dieser Schutz der persönlichen Daten steht im Mittelpunkt der vorliegenden Arbeit, weshalb die hierfür in Frage kommenden Mechanismen im Folgenden gesondert betrachtet werden.

2.2 Verwandte Arbeiten zum Schutz der Privatsphäre

Die Beispiele der vorangehenden Abschnitte zeigen, dass es eine Vielzahl an kreativen Einsatzmöglichkeiten von Kontext für die Umsetzung innovativer Anwendungen gibt. Zudem lässt sich eine große Bandbreite an Kontextinformationen direkt über das mobile Endgerät eines Nutzers ermitteln. Gleichzeitig ist der potentieller Verlust sensibler, persönlicher Daten bei der Nutzung derartiger Dienste von substantiellem Interesse. Darauf weist auch Peter Schaar, der ehemalige Bundesdatenschutzbeauftragte im Jahr 2016 hin:

> „*Der Einzelne soll selbst entscheiden (können), was er über sich preisgibt. Wer befürchten muss, dass sein gesamtes Verhalten registriert und in Persönlichkeitsprofilen zusammengefasst wird, kann sich nicht frei entscheiden und entwickeln. Er wird auf die Wahrnehmung mancher Rechte verzichten und Verhaltensweisen meiden, die irgendwelche nachteiligen Folgen für ihn haben könnten, sei es im Privat- oder im Arbeitsleben. Datenschutz ist als Basisgrundrecht der Informationsgesellschaft nicht verhandelbar.*" [211]

Unterschiedliche Disziplinen kennen dabei verschiedene Begrifflichkeiten. So wird aus rechtlicher Sicht zwischen Privatsphäre und Individualsphäre unterschieden. Während die Privatsphäre das Privatleben und den häuslichen Bereich in der Familie meint, fällt das Recht auf informationelle Selbstbestimmung in den Bereich der Individualsphäre. Die Psychologie versteht unter der Persönlichkeitssphäre „*die Gesamtheit des menschlichen Verhaltens und speziell die Gesamtheit aller Handlungen, die zu einem vorhersagbaren Verhalten führen*" [1]. Alle drei Begriffe lassen sich auf den in der vorliegenden

Arbeit angestrebten Schutz von Kontextinformationen einer Person übertragen. Nachfolgend wird ausschließlich Begriff der Privatspähre genutzt, wobei die Definitionen der anderen Sphären stets mit abgedeckt sein sollen.

Mit den unterschiedlichen Möglichkeiten zur Umsetzung der informationellen Selbstbestimmung in kontext- und ortbezogenen Diensten beschäftigen sich die folgenden Abschnitte.

2.2.1 Allgemeine Datenschutzprinzipien

Es wird meist zwischen vier unterschiedlichen Herangehensweisen unterschieden, wie sich der Schutz der Privatsphäre erreichen lässt [70]:

- *Gesetzliche Regelungen* stellen die EU-Direktive 2002/58/EG oder das in Deutschland geltende Grundrecht auf informationelle Selbstbestimmung[4] dar. In beiden Fällen wird u.a. verlangt, dass der Nutzer seine Einwilligung zur Datenerhebung abgeben muss sowie dass der Empfänger der Daten gewissen Sorgfaltspflichten nachkommt.

- *Vereinbarungen* (engl. *Policies) mit dem Dienstanbieter* orientieren sich am geltenden Recht aus dem Land des Dienstanbieters und müssen dem Nutzer bei der Inanspruchnahme des Dienstes zugänglich gemacht werden. Dabei handelt es sich z.B. um die Datenschutzbestimmungen des Anbieters, die angeben, was mit den gesammelten Daten geschieht, wie sie gespeichert und verwendet werden. Der Kunde vertraut dabei dem Dienstanbieter, dass dieser sich an die Vereinbarungen hält.

- Eine *anonyme Dienstnutzung* kann eingesetzt werden, wenn für die Diensterbringung weder die reale Identität des Nutzers noch ein wiedererkennbares Pseudonym bekannt sein muss. Einzelne Daten und Anfragen lassen sich dabei nicht demselben Nutzer zuordnen.

- Die *Verschleierung* von Kontextinformationen ermöglicht eine privatsphäreschonende Verwendung kontextbezogener Dienste, selbst wenn eine anonyme Nutzung nicht möglich ist. Im Gegensatz zur Herstellung von Anonymität lässt sich dieser Prozess rein clientseitig umsetzen.

Entsprechend der gesetzlichen Zustimmungspflicht gestalten sich i.d.R. die Vereinbarungen mit dem Dienstanbieter. Meist ist dieser Pflicht durch einen eizelnen Klick zur Akzeptanz der Datenschutzbestimmungen bei der ersten Dienstnutzung genüge getan. Eine tatsächliche Kontrolle über Umfang und Detailgrad der geteilten Kontextinformationen besteht darüber hinaus nicht.

An dieser Vorgehensweise lassen sich mehrere Schwachstellen ausmachen. Einerseits sieht sich der Nutzer in den meisten Fällen einer binären Entscheidung gegenüber – verweigert man die Zustimmung, lässt sich der Dienst nicht nutzen. Zweitens besteht in großen Teilen der Bevölkerung eine Missinterpretation

[4] BVerfG, Urteil v. 15. Dezember 1983, Az. 1 BvR 209, 269, 362, 420, 440, 484/83

hinsichtlich der Bedeutung von Datenschutzbestimmungen [235]: Nur, dass es eine solche gibt, bedeutet entgegen weitverbreiteter Meinung noch nicht, dass die Daten z.B. nicht an andere Unternehmen weitergegeben werden. Zuletzt besteht – selbst wenn ich einem Anbieter und seinen Bestimmungen vertraue – bei der Preisgabe persönlicher Daten stets die Gefahr eines Datenlecks.

Eine Liste an unverbindlichen Empfehlungen, wie der Datenschutz in der Entwicklung von Software berücksichtigt werden sollte, liefert der *Privacy-by-Design*-Ansatz [40]. Hierfür werden sieben Grundprinzipien eingeführt, an die sich Softwarearchitekten und -entwickler im Idealfall halten sollen. Dazu zählen der *proaktive Schutz der Privatsphäre*, *Datenschutz als Default-Einstellung*, die *Einbettung des Datenschutzes in den Designprozess* und Designentscheidungen stets aus der *Nutzerperspektive* zu treffen. Diese Prinzipien sind jedoch relativ unkonkret und geben keine technischen Lösungen vor [200].

Ein Beispiel für den fairen Umgang mit persönlichen Informationen ist das *Need-to-know*-Prinzip [128]. Hier werden nur jene Daten gesammelt, die für die Diensterbringung absolut unumgänglich sind. Die vorliegende Arbeit beschäftigt sich mit technischen Möglichkeiten zur clientseitigen Umsetzung dieses Prinzips. Hinzu kommen jedoch noch individuelle Privatsphärebedürfnisse eines Nutzers, die es ebenfalls zu berücksichtigen und situationsabhängig gegen Aspekte wie Dienstqualität und -verfügbarkeit abzuwägen gilt.

2.2.2 Privatsphäre in kontextbezogenen Diensten

In diesem Abschnitt soll eine Übersicht über bestehende Ansätze für den Schutz der Privatsphäre in Anwendungen gegeben werden, die ihre Dienste auf Basis unterschiedlicher Kontextinformationen ihrer Nutzer erbringen und daher voraussetzen, dass externe Systeme, die sich nicht unter der Kontrolle des jeweiligen Nutzers befinden, in Besitz dieser persönlichen Daten gelangen.

Wie in Kapitel 2.1 beschrieben, gehören hierzu sowohl personenbezogene Informationen wie seine Identität, der aktuelle Aufenthaltsort des Nutzers, die derzeit ausgeführte Aktivität, sein Stresslevel oder seine Stimmungslage. Andererseits sind hier aber auch Informationen über seine Umgebung oder das verwendete Endgerät betreffend beinhaltet. Welche Anwendung sich gerade im Vordergrund befindet, wie hoch die verbleibende Batterieladung noch ist oder die Geschwindigkeit der momentan verfügbaren Datenverbindung.

Aufgrund der Popularität entsprechender Anwendungen beschäftigt sich sowohl die vorliegende Arbeit als auch ein Großteil der bekannten Literatur speziell mit dem Schutz von Standortdaten. In diesem Zusammenhang beschreiben Beresford et al. den wichtigen Begriff der *Location Privacy* als „die Fähigkeit einer Person, andere Parteien davon abzuhalten, ihre aktuellen oder früheren Aufenthaltsorte zu ermitteln." [26] Etwas detaillierter umschreiben es Duckham et al. als „der Anspruch einer Person, für sich selbst bestimmen zu können, zu welcher Zeit, an welchem Ort, auf welche Weise und in welchem Umfang ihre Standortinformationen an andere kommuniziert werden." [70]

Vor dem Hintergrund des in Deutschland geltenden Grundrechts auf informationelle Selbstbestimmung muss der Nutzer stets dazu in der Lage sein, zu wissen, welche Informationen aktuell über ihn gesammelt werden. Auf Basis dieses Wissens muss er im nächsten Schritt kontrollieren können, welche Typen von Kontextinformationen in welcher Auflösung gesammelt werden dürfen. Dies muss unabhängig davon gelten, ob es sich um kontextabhängige Anwendungen im eigentlichen Sinn handelt, die laut Dey eine Aufgabe oder ein Ziel des Nutzers unterstützen oder wie im Fall vieler Apps um solche, die Informationen sammeln ohne diese für ihre Diensterbringung zu benötigen.

Als eine der ersten Arbeiten zu diesem Thema haben Lederer et al. bereits 2003 die Wichtigkeit von Privatsphäre in kontextabhängigen Anwendungen betont [159]. Auf Basis einer Nutzerstudie kommen die Autoren zu dem subjektiv nachvollziehbaren Ergebnis, dass neben Art und Inhalt der übermittelten Daten insbesondere die aktuelle Situation des Nutzers dafür ausschlaggebend ist, ob man Informationen preisgeben möchte oder nicht.

Anhand der Studienergebnisse stellen die Verfasser darüber hinaus fest, wie wichtig eine Möglichkeit zur manuellen Festlegung dieser Regeln ist. In den verwendeten Beispielszenarien haben sich die Studienteilnehmer zum Großteil für die Erstellung individueller Präferenzen entschieden, obwohl dies mühselig und mit manuellem Konfigurationsaufwand verbunden ist.

Andere Studien kommen zu übereinstimmenden Ergebnissen und stellen die Individualität von Privatsphärepräferenzen sowie den Einfluss verschiedener Kontextinformationen auf die Freigabeentscheidung des Nutzers in den Vordergrund. Dazu zählen u.a. dessen Stimmungslage, die Art der Gesellschaft, in der man sich aktuell befindet sowie die soziale Beziehung des Benutzers zum Empfänger der Informationen [30, 142, 256, 265].

2.2.2.1 Privatsphäre in Ubiquitous Computing-Umgebungen

Einige Arbeiten beschäftigen sich mit der Fragestellung, wie ein Nutzer seine Privatsphäre innerhalb eines *Ubiquitous Computing*-Umfelds schützen kann, in der eine Vielzahl von Infrastrukturkomponenten potentiell private Informationen über ihn ermittelt. Hierzu zählen z.B. Kameras, die über eine Gesichtserkennung die Identität anwesender Personen ermitteln oder anhand visueller Klassifizierungsverfahren die Aktivitäten der Personen erkennen.

Obwohl sich die vorliegende Arbeit mit der privatsphärezentrischen Verwaltung von Kontextinformationen befasst, die auf dem mobilen Endgerät des Nutzers anfallen, sollen diese Ansätze der Vollständigkeit halber ebenfalls erwähnt werden.

Das *Privacy Awareness System (pawS)* [158] ermöglicht es den Sensoren einer intelligenten Umgebung, ihre Policies für den Umgang mit den von ihnen erhobenen persönlichen Daten in Form sogenannter *Privacy Beacons* mitzuteilen. Das System stellt eine Erweiterung des (in der Praxis relativ erfolglosen) *Platform for Privacy Preferences Project* (P3P) des W3C dar. Dort können

Webseiten festlegen, welche Daten sie erheben, an wen sie weitergeleitet werden und zu welchem Zweck. Der Browser eines Nutzers kann dann entscheiden, ob er diese Webseite besuchen möchte oder nicht. Die Rolle des Browsers übernimmt bei *pawS* das mobile Endgerät des Nutzers, das als sog. Privacy Proxy agiert, indem es die von der Infrastruktur versandten Beacons empfängt und gemäß der Nutzerpräferenzen entscheidet, ob eine Dienstleistung in Anspruch genommen werden kann oder nicht. Ein ähnlicher Ansatz samt einer standardkonformen Integration der datenschutzrelevanten Beschreibung solcher intelligenten Infrastrukturkomponenten in die Beacons des 802.11-WLAN-Standards wurde 2015 von Schaub et al. vorgestellt [212].

Hong et al. räumen dem Datenschutz beim Entwurf der *Confab*-Infrastruktur, welche die einfache Erstellung kontextabhängiger Dienste ermöglichen soll, viel Platz ein [122]. *Confab* besteht aus verteilten Speichereinheiten, sog. *Information Spaces*, die jeweils den Kontext einer bestimmten Entität, wie z.B. eines Raumes oder einer Person, verwalten. Es lassen sich dabei durch den Nutzer einfache Regeln anlegen, die den Zugriff auf persönliche Informationen durch die Identität der anfragenden Entität sowie den Zeitpunkt der Anfrage regeln. Darüber hinaus sieht dieser Ansatz vor, alle ausgetauschten Kontextinformationen mit *Privacy Tags* zu versehen, die wie im Rahmen des digitalen Rechtemanagements (engl. *Digital Rights Management*, DRM) an die Daten gekoppelt sind und beschreiben, wie mit den Informationen umzugehen ist. Als Policy-basierter Ansatz setzt dies natürlich voraus, dass sich alle Empfänger von Kontextinformationen tatsächlich an diese Regeln halten.

Zentral aufgebaut ist die *Context Broker Architecture (CoBrA)* von Chen et al. [44]. Ein solcher Broker ist für die Akquise von Kontextinformationen aus einem bestimmten, als *Smart Space* bezeichneten räumlichen Bereich zuständig. Die Erzeugung semantisch angereichterter Kontextinformationen aus den Beobachtungen aller am System registrierten Sensoren wird durch den Einsatz eines ontologiebasierten Kontextmodells und logischer Inferenz ermöglicht.

Der Infrastrukturbetreiber kann mit *CoBrA* auf Basis der *REI Policy Language* [135] Rechte definieren, welche Entitäten auf Sensoren und Aktoren der Umgebung wie z.B. Beamer oder Lichtschalter zugreifen dürfen. Für verschiedene Umgebungen lassen sich zudem durch die Nutzer Regeln für die Freigabe persönlicher Profilinformationen (wie z.B. ihre Privatadresse) an einen Dienst angeben, die von einem zentralen *Privacy Protection Module* überwacht werden [46]. Ob und wie widersprüchliche Regeldefinition erkannt und aufgelöst werden, wird von den Autoren nicht angesprochen, ebenso wie unterschiedliche Granularitäten von Kontext oder die Ausgewogenheit im Informationsfluss zwischen Entitäten nicht berücksichtigt werden.

Henricksen et al. beschreiben die Integration des Konzepts des Kontextbesitzers in die Kontextmodellierung, um dynamisch verwalten zu können, welcher Entität eine bestimmte Quelle von Kontextinformationen, wie. z.B. eine Kamera in einem Besprechungsraum, derzeit „gehört" [116]. Die Akquise, Verwaltung und Freigabe von Kontextinformationen findet dabei durch eine

zentrale Komponente statt, das *Context Management System* (CMS). Der Kontextbesitzer kann Präferenzen angeben, die einer eindeutig identifizierten Entität unter bestimmten Bedingungen Zugriff auf eine Kontextquelle oder -information erlaubt oder verwehrt, die ihm derzeitig zugeordnet ist. Interessant hierbei ist, dass der Besitz von Resourcen dynamisch wechselt. So verliert eine Person z.B. den Besitz der Kamera, wenn sie den Raum verlässt. In diesem Fall kann wieder frei auf die Daten dieses Sensors zugegriffen werden.

Dem Grundgedanken der Ubiquitous Computing-Vision folgend gehen diese Arbeiten jeweils davon aus, dass die Kontextinformationen über einen Nutzer nicht von seinem Endgerät erfasst werden, sondern von Sensoren der Umgebung. Sie versuchen daher zu kontrollieren, welche Daten von externen Quellen erhoben werden und müssen sich dementsprechend auf die Kooperation und Integrität der Infrastrukturbetreiber verlassen.

Derartige Systeme sind als komplementär zu dem in Kapitel 3 vorgestellten Ansatz zu sehen, der sich auf das Szenario konzentriert, in dem das Smartphone selbst Kontextinformationen über seinen Nutzer liefert und diese für die Nutzung kontextabhängiger Dienste mit anderen Komponenten austauschen muss. Auch hierfür gibt es in der Literatur verschiedene Lösungsansätze, die aufgrund der großen thematischen Nähe zu den Inhalten der vorliegenden Arbeit in Kapitel 3.3.2 beschrieben werden.

2.2.2.2 Maßnahmen zum Erhalt der Privatsphäre von Kontextinformationen

In diesem Abschnitt werden etablierte Mechanismen zum Schutz der Privatsphäre im Rahmen kontextbezogener Dienste vorgestellt, die ein Benutzer über sein mobiles Endgerät nutzt, welches gleichzeitig auch die einzige Quelle der über seine Situation zur Verfügung stehenden Kontextinformationen darstellt. Eine grobe Übersicht über die grundlegenden Prinzipien und Herangehensweisen, die dabei zum Einsatz kommen können, geben auch Bokhove et al. [35].

2.2.2.2.1 Qualitätsanpassung von Kontextinformationen Eine Möglichkeit, die Nutzung kontextabhängiger Dienste privatsphäreschonend zu gestalten, ist die Qualität der ermittelten Kontextinformationen in Anlehnung an das *Need-to-know*-Prinzip [128] künstlich zu reduzieren, bevor diese an einen externen Dienstanbieter übermittelt werden. Unter dem Begriff der *Quality-of-Context* (QoC) fassen Sheik et al. [218] unterschiedliche solcher Maßnahmen zusammen, die sich teilweise generisch für beliebige Typen von Kontextinformationen umsetzen lassen. So kann z.B. die *Präzision* numerisch beschreibbarer Kontexttypen durch Runden herabgesetzt werden, Durchschnittswerte oder ein Intervall anstelle eines exakten Wertes angegeben werden.

Die Autoren beschreiben damit eine verallgemeinerbare Form der Kontextverschleierung – die Generalisierung – die sich auf verschiedenste Ausprägun-

gen von Kontext anwenden lässt. Auch die Verschleierung der exakten Aktivitäten eines Benutzers ist auf diese Weise denkbar. Die Aktivität eines Nutzers, der z.B. gerade die Email seines Vorgesetzten liest, kann auf unterschiedlichen Detailstufen wie folgt angegeben werden, wobei der Detailgrad schrittweise abnimmt: `read-e-mail-from-supervisor`, `read-e-mail-from-colleague`, `read-e-mail`, `computer-work`, `office-work` und `working`. Im Gegensatz zu numerischen Werten ist hierfür natürlich eine entsprechende Taxonomie von Aktivitäten nötig, entlang derer die Generalisierung durchgeführt werden kann. Im Rahmen der Kontextmodellierung findet eine solche hierarchische Klassifizierung aller Kontexttypen jedoch schon zur Ermöglichung der Inferenz auf einzelnen Kontextinformationen meist ohnehin statt, sodass dies eine realistische und praktikable Herangehensweise darstellt (vgl. Kapitel 3.3.1.2).

Ein weiteres Mittel zum Schutz der Privatsphäre ist die Anpassung der *Korrektheit* von Kontextinformationen. Anstelle ungenauer Angaben wird hierbei schlicht und ergreifend gelogen. Dies kann in Form von Notlügen (engl. *White Lies*) geschehen [14], z.B. um durch eine falsche Standort- oder Aktivitätsangabe die tatsächlichen Kontextinformtionen zu verheimlichen oder einen Kommunikationspartner bewusst zu täuschen. Es soll an dieser Stelle darauf hingewiesen werden, dass sich diese Herangehensweise an normalem zwischenmenschlichen Alltagsverhalten orientiert. Man nimmt den Telefonhörer nicht ab, obwohl man daneben sitzt oder erzählt dem Partner, man müsse länger arbeiten, um zu verbergen, dass man gerade einen Verlobungsring besorgt.

Intelligentere Methoden zur Generalisierung sowie zur Erzeugung von falschen Angaben für den speziellen Kontexttyp der Standort- bzw. Mobilitätsinformationen werden in Kapitel 2.2.3 umfassend beschrieben und klassifiziert.

Aktualität (engl. *Freshness*) und *zeitliche Auflösung* sind zwei weitere, generisch einsetzbare Verschleierungstechniken. Im ersten Fall wird dabei anstelle der jeweils neuesten Kontextinformation ein früherer Wert zurückgeliefert, der vor einer frei wählbaren Zeitspanne gültig war. Aus Sicht der Privatsphäre wird dadurch verhindert, dass ein Angreifer einen Benutzer in Echtzeit überwachen kann, was bei einigen Typen von Kontextinformationen wie z.B. dem aktuellen Standort ein echtes Sicherheitsbedürfnis darstellen kann.

Im zweiten Fall wird anstelle eines genauen Zeitpunkts, zu dem eine Information erhoben wurde oder Gültigkeit besaß, nur ein grobes zeitliches Intervall an den Kontextempfänger übermittelt. Diese Technik lässt sich freilich nicht für interaktive Dienste einsetzen, bei denen der Nutzer eine kontextabhängige Anfrage stellt und eine unmittelbare Antwort erwartet. Im Rahmen von Crowdsourcing-Kampagnen o.ä. kann dies jedoch ein probates Mittel zur Verschleierung von Mobilitätsinformationen darstellen. Im Gegensatz zur eingangs genannten Generalisierung werden hierbei nicht die Kontextinformationen selbst, sondern deren Metainformationen verändert.

Aus derselben Kategorie an Verschleierungstechniken machen Sheik et al. noch die *Probability-of-Correctness* (PoC) als weitere Stellschraube für den Schutz der Privatsphäre aus [218]. Hierbei wird ähnlich des Prinzips der *Plau-*

sible Deniability versucht, den potentiellen Angreifer durch eine scheinbar geringe Zuverlässigkeit der übermittelten Informationen zu täuschen. Viele Verfahren aus dem Bereich des maschinellen Lernens geben z.B. eine Wahrscheinlichkeit an, mit der ein Klassifikationsergebnis korrekt ist. Diese PoC kann für die Herstellung einer solchen „plausiblen Abstreitbarkeit" künstlich herabgesetzt und als Metainformation mit an den Dienstanbieter übertragen werden. Da in der Sicherheits- und Privatsphäreliteratur jedoch üblicherweise davon ausgegangen wird, dass ein Angreifer die verwendeten Schutzmechanismen kennt, scheint fraglich, wie sich hierbei eine Balance aus qualitativ hochwertiger Dienstnutzung und Privatsphäreschutz erreichen lassen soll.

2.2.2.2.2 Symmetrie im Datenaustausch Ein weiteres Konzept, die Privatsphäre bei der Umsetzung kontextbezogener Dienste zu schützen, ist es, eine Ausgewogenheit im Informationsaustausch zu erwirken. Hierbei wird darauf geachtet, dass jede Partei im selben Maße Informationen über sich selbst preisgeben muss, wie sie Daten von anderen abfrägt. Jiang et al. führen dazu das Prinzip der „minimalen Asymmetrie" ein, das voraussetzt, dass ein Dienstanbieter den Nutzer im Austausch gegen dessen Informationen über jedes Detail der Datenerhebung, -nutzung, -speicherung und -weitergabe informiert [132].

In [145] wird dieses Prinzip auf den symmetrischen Austausch von Status-Informationen (Online, Offline, Verfügbar, Do-not-disturb, etc.) in einer bestehenden Instant-Messenger-Applikationen überführt. Der zentrale Chat-Server sorgt dafür, dass wenn ein Nutzer seinen eigenen Status verbirgt, er auch nicht mehr die Orte und Status seiner Kontaktlisten-Freunde einsehen kann. Insbesondere für Peer-to-Peer-basierte kontextabhängige Anwendungen stellt dies einen interessanten Aspekt dar, der zwischenmenschlichem Kommunikationsverhalten entspricht und Ungleichgewichte im Informationsfluss zwischen zwei Parteien verhindert.

2.2.2.2.3 Anonyme und pseudonyme Dienstnutzung Kontextbezogene Anwendungen, die keine personalisierten Dienste erbringen, sollten aus Sicht der Privatsphäre stets anonym verwendet werden [157]. Dabei wird in der Kommunikation zwischen Endgerät und Dienstanbieter kein Identifier des Nutzers übertragen, sodass sich einzelne Anfragen im Idealfall nicht miteinander in Verbindung bringen lassen. Anonymität wird dabei wie folgt definiert:

> „*Anonymity is the state of being not identifiable within a set of subjects, the anonymity set.*" [195]

Die dafür zudem nötige Kommunikationsanonymität lässt sich i.d.R. nur durch die Existenz eines vertrauenswürdigen Dritten herstellen, um auch netzwerkbasierte Identifier wie die IP-Adresse des Nutzers zu verstecken. Dies kann eine als TTP-definierte zentrale Komponente sein oder ein Netzwerk aus anderen Nutzern, die im Rahmen eines Peer-to-Peer-Ansatzes die Anonymität

des Einzelnen schützen und entsprechende Nachrichten zufällig untereinander weiterleiten, bevor sie den Dienstanbieter erreichen.

Bei einer pseudonymen Dienstnutzung hingegen ist zwar auch nicht die tatsächliche Identität des Nutzers bekannt, wohl aber ein wiederverwendeter Identifikator, das sog. Pseudonym, das in jeder Anfrage enthalten ist und in gewissem Rahmen somit auch personalisierte Dienste ermöglicht.

„*Pseudonymity is the use of pseudonyms as IDs.*" [195]

Es ist jedoch zu beachten, dass Anonymität und Pseudonymität oftmals nur eine trügerische Sicherheit bieten. Lane et al. [156] gelingt z.B. die zuverlässige De-Anonymisierung, d.h. die Wiedererkennung einzelner Nutzer auf Basis anonym zur Verfügung gestellter Sensordaten. Insbesondere auf Standortdaten existieren viele Angriffe, die sehr erfolgreich auf die De-Anonymisierung von Nutzern abzielen. Solche Verfahren und aus der Literatur bekannte Gegenmaßnahmen werden in den Kapiteln 2.2.3 und 4.2 im Detail besprochen.

2.2.2.2.4 Privatsphäreschonende Datenerhebung und Publikation von Datenbanken Einige kontextabhängige Anwendungen und LBS basieren auf der Sammlung, Auswertung und letztendlichen Veröffentlichung von Daten, die über ihre Nutzer gesammelt werden. Beispiele hierfür liefert das *Crowd Sensing*, wie z.B. die Ermittlung von Durchschnittsgeschwindigkeiten auf Strassensegmenten auf Basis der Sensordaten mobiler Nutzer. Um die Privatsphäre der Studienteilnehmer auch im Rahmen solcher Anwendungen zu schützen, können entweder bei der Datenerhebung oder vor der Publikation der Datenbank durch den Datenkurator entsprechende Schutzmaßnahmen ergriffen werden.

PoolView [84] von Ganti et al. ist ein Verfahren zum Schutz der persönlichen Kontextinformationen während der Datenerhebung. Bevor ein Teilnehmer seine Messwerte an den Dienstanbieter schickt, werden diese „geeignet" verfremdet. Im Gegensatz zu den oben genannten Mechanismen der Generalisierung oder bewussten Täuschung muss hierbei jedoch darauf geachtet werden, dass die globale Datenqualität nicht unter der lokalen Verschleierung leidet. Dies wird erreicht, indem jeder Teilnehmer zufälliges Rauschen aus einer passend parametrisierten statistischen Verteilung auf seine tatsächlichen Messwerte hinzu addiert. Die Autoren zeigen für verschiedene Typen von Kontextinformationen mit unterschiedlichen Wertebereichen und Dynamiken – Fahrgeschwindigkeiten und Gewichtsverlauf von Studienteilnehmern – wie die entsprechenden Rauschmodelle berechnet werden können und dass sich ab einer gewissen Mindestnutzerzahl die tatsächlichen Verteilungen sehr genau annähern lassen, ohne dass die von einem Einzelnen beigesteuerten Werte verräterisch sind.

Die nachträgliche privatsphärebezogene Optimierung einer Datenbank verfolgt Latanya Sweeney mit ihrem Prinzip der k-Anonymität [229], das auch umfangreiche Adaptionen für den Schutz der Privatsphäre in ortsbezogenen Diensten erfahren hat (vgl. Kapitel 2.2.3.1). Dabei wird davon ausgegangen, dass der Angreifer beliebiges Hintergrundwissen über die Zielperson(en) wie

z.B. Geburtstdatum oder Wohnort besitzt und dass er weitere Informationen über diese Person aus den Daten ermitteln möchte. Der Grundgedanke der k-Anonymität ist es, die Einträge einer Datenbank vor ihrer Veröffentlichung so zu modifizieren, dass eine korrekte Zuordnung eines Individuums zu einem bestimmten Datensatz maximal mit der Wahrscheinlichkeit $\frac{1}{k}$ möglich sein darf.

Zu diesem Zweck müssen nicht nur alle unmittelbaren Identifikatoren aus jedem Datensatz entfernt oder durch Pseudonyme ersetzt werden, sondern auch sogenannte *Quasi-Identifier* erkannt und ggf. eliminiert werden. Diese entstehen aus der Kombination mehrerer, für sich genommen jeweils unkritischer Informationen, die – wie z.B. im Falle von Geburtsort und -datum – in vielen Fällen doch zur eindeutigen Identifikation eines Individuums führen können. Um die Bedingung der k-Anonymität zu erfüllen, muss für jeden Eintrag eine Verschleierungsmenge existieren, die mindestens $k - 1$ weitere Elemente enthält, auf welche dieselben Quasi-Identifier zutreffen.

Möglichkeiten zur Erreichung des Zustands einer k-anonymen Datenbank sind u.a. erneut die Generalisierung von Daten, wie z.B. das Abschneiden von Postleitzahlen nach einer gewissen Stelle oder das Ausblenden von Attributen, die sich nicht entsprechend umformen lassen. Nur solche Quasi-Identifier, die mindestens k mal in der Datenbank vorkommen, dürfen veröffentlicht werden.

Darauf aufbauende Konzepte stellen die *l-Diversity* [170] sowie die *t-Closeness* [163] dar, die verschiedene Unzulänglichkeiten der k-Anonymität hinsichtlich der Verteilung der sensiblen Attribute nachweisen und lösen. So verhindert die k-Anonymität z.B. keine Homogenitätsattacken – besitzen alle k möglichen Einträge denselben Wert für ein sensibles Attribut, kann der Angreifer diese Information über seine Zielperson eindeutig in Erfahrung bringen. Die l-Diversität fordert daher, dass in jedem Anonymitätsset jedes sensible Attribut mindestens l verschiedene Werte annehmen muss.

Eine alternative Herangehensweise wird bei der *Differential Privacy* [71] verfolgt, die den Schutz einzelner Einträge bei der Beantwortung von Aggregatsanfragen an eine Datenbank fokussiert. Hier werden Hinweise auf die Existenz einzelner Einträge in einer Datenbank durch das Einfügen von zufälligem Rauschen vermieden, das sich an der Werteverteilung der Originaldaten orientiert. Gegeben seien zwei Datenbanken, die sich nur durch das Vorhandensein und Nicht-Vorhandensein eines einzigen Eintrags voneinander unterscheiden (daher die Bezeichnung „differential"). Differential Privacy bewirkt, dass sich die Wahrscheinlichkeiten beider Datenbanken, das jeweils beobachtbare Ergebnis auf eine Aggregatsanfrage zu produzieren, maximal um den Faktor e^{ϵ} unterscheiden. Somit sind kaum Rückschlüsse darauf möglich, ob sich dieser eine Eintrag nun in der Tabelle befand oder nicht.

Ein grundsätzliches Problem der k-Anonymität und ihrer Ableger sowie der Differential Privacy ist jedoch, dass der Nutzer dem Datenkurator stets volles Vertrauen entgegenbringen muss – obwohl dieser selbst das wahrscheinlichste Datenleck darstellt [202]. Ansätze zur clientbasierten Verschleierung, wie in diesem Zusammenhang PoolView, vermeiden dieses Problem.

2.2.2.2.5 Feingranulare Zugangskontrolle Einen entscheidenden Aspekt für den Privatsphäreschutz von Kontextinformationen sehen Bokhove et al. in der Verfügbarkeit von individuell einstellbaren, feingranularen und situationsabhängigen Freigaberegeln [35]. Der Nutzer muss darüber hinaus in der Lage sein, diese Regeln einfach zu revidieren oder zu modifizieren.

Gemäß einer *klassischen Rechteverwaltung* in Computersystemen muss es mindestens möglich sein, lesende Zugriffe auf die eigenen Kontextinformationen für verschiedene Kontextempfänger zu erlauben und zu verweigern. Zudem sollen diese Regeln sowohl *feingranular* sein, d.h. sich auch auf bestimmte Subtypen oder Detailstufen einer Kontextinformation beziehen können als auch selbst *kontextabhängig* definiert werden können. So zeigen u.a. Xie et al. [256], dass die Bereitschaft eines Nutzers, seine derzeitigen Kontextinformationen mit anderen Parteien zu teilen, stark von unterschiedlichsten Einflussfaktoren abhängt. Neben der übermittelten Information selbst sind das insbesondere der jeweilige Zeitpunkt, die Personen, in deren Gesellschaft man sich befindet sowie die eigenen Emotionen und Stimmungslagen. Diverse andere Studien wie z.B. [30, 142, 265] bestätigen diese Beobachtungen und identifizieren zusätzliche Kontextinformationen als weitere relevante Einflussfaktoren.

Eine Schwierigkeit und potentielle Hürde für die Erstellung von Freigaberegeln durch die Benutzer ist in der Benutzerfreundlichkeit zu sehen. So wird kein Nutzer im Vorfeld alle seine Privatsphärebedürfnisse in Regeln fassen. [35] plädiert aus diesem Grund für eine zusätzliche *Just-in-Time*-Freigabe bei eingehenden Anfragen für Kontextinformationen. Diese widerspricht jedoch dem Grundgedanken kontextabhängiger Anwendungen, die ja so viele Aufgaben wie möglich vom Nutzer fernhalten und automatisieren wollen.

Gleichzeitig ist jedoch klar, dass sich ohne Zutun der Nutzer ein sinnvoller, individueller Schutz der Privatsphäre nicht erreichen lässt. Die dafür notwendige Bereitschaft muss den Menschen wohl erst antrainiert werden – so sieht es z.B. auch Hacker-Aktivist Julian Assange:

> „*Ich glaube, die einzig wirksame Verteidigung* [...] *besteht darin, eigene Schritte zum Schutz der Privatsphäre zu unternehmen, denn den Datenkraken, die heute alles abgreifen können, fehlt jeder Anreiz zur Selbstbeschränkung. Man könnte eine historische Analogie zur Verbreitung des Händewaschens ziehen.*" [13]

Verwandte Arbeiten, die sich mit der regelbasierten Freigabe von Kontextinformationen beschäftigen, werden in Kapitel 3.3.2 besprochen.

2.2.2.2.6 Benachrichtigungen und Übersicht über gesammelte Informationen Analog zu den *Privacy-by-Design*-Prinzipien führen Bokhove et al. [35] als weitere Voraussetzung auf, dass der Nutzer jederzeit dazu in der Lage sein muss, die Daten, die ein bestimmter Kontextempfänger über ihn gesammelt hat, einzusehen. Um dies clientseitig zu ermöglichen, müssen alle freigegebenen Kontextinformationen in einer Logdatei mitgeschrieben werden.

Darüber hinaus sollte dem Nutzer die Möglichkeit gegeben werden, dass er Echtzeit-Benachrichtigungen abonniert, wenn ein bestimmter Kontextempfäger auf seine Kontextinformationen zugreift. Ein derartiger Mechanismus ist z.B. auch in der TTP-basierten Plattform zur zentralisierten Kontextverwaltung in [206] enthalten.

Diese Techniken tragen dazu bei, dass sich der Nutzer stets ein Bild davon machen kann, wer welche Informationen über ihn besitzt. Dies erlaubt z.B. eine bewusste Auseinandersetzung mit den bisher aufgestellten Freigaberegeln und ggf. eine passende Modifikation.

Ein weiterer wichtiger Aspekt, der bei der Preisgabe von Kontextinformationen beachtet werden muss, ist, dass sich aus der geeigneten Analyse eines Kontexttyps, der auf den ersten Blick u.U. wenig privatsphärekritisch erscheint, auch Rückschlüsse auf andere Kontextinformationen ziehen lassen, die sehr wohl die Privatsphärebedürfnisse des Nutzers tangieren.

In [111] wird der aktuelle Standort sowie die gefahrene Route eines Fahrzeugs allein anhand der Accelerometer- und Magnetometerwerte eines Smartphones ermittelt. Lex et al. [162] gelingt die korrekte Bestimmung des semantischen Orts (Zuhause, Haus eines Freundes, Restaurant, Arbeitsplatz, etc.) anhand von Accelerometerdaten, WLAN-Verfügbarkeit und dem Ladestatus des Smartphones. Umgekehrt kann die aktuell ausgeführte Aktivität auch aus dem Aufenthaltsort eines Nutzers geschlossen werden [237]. Madan et al. [171] leiten den Grad der politischen Aktivität und politische Einstellungen ihrer Studienteilnehmer aus den Nutzungsdaten ihres Smartphones ab. In [176] ermitteln Michalevsky et al. den Standort eines in Bewegung befindlichen Nutzers anhand von ortstypischen Mustern im Gesamt-Batterieverbrauch seines Smartphones, die durch wechselnde Signalstärken und Entfernungen zu Mobilfunkmasten entstehen und einen charakteristischen Fingerabdruck ergeben.

Es lässt sich daher nicht pauschal sagen, welche Typen von kontextinformationen privatsphärerelevant sind und welche nicht. Besonders kritisch erscheint vor diesem Hintergrund die Tatsache, dass viele dieser Sensordaten (z.B. Accelerometer, Gyroskop, Magnetometer und Batteriestand) auf aktuellen mobilen Betriebssystemen nicht durch entsprechende Rechte, die der Nutzer zuteilen oder verweigern kann, geschützt sind. Die grundlegende, aus dem Einsatz jeglicher Form von Sensortechnik erwachsende Problematik wird von Ackermann et al. [3] plakativ formuliert: „*One person's sensor is another person's spy.*"

Der durchschnittliche Nutzer kann derartige Querverbindungen nicht ziehen und soll sich auch gar nicht damit beschäftigen. Vor diesem Hintergrund ist es viel mehr Aufgabe der Wissenschaft, diese Zusammenhänge zu untersuchen und Mechanismen gegen deren Missbrauch zu entwickeln.

2.2.3 Privatsphäre in ortsbezogenen Diensten

Sowohl in wissenschaftlichen Betrachtungen als auch unter den tatsächlich verfügbaren Anwendungen stellen LBS die häufigste Ausprägung kontextbezogener Dienste dar. Gleichzeitig werden die Aufenthaltsorte und Bewegungsmuster einer Person einhellig als schützenswerte private Informationen angesehen. Dementsprechend beschäftigt sich eine große Zahl an Arbeiten mit dem Schutz der Privatsphäre bei der Nutzung solcher Dienste, die im Rahmen ihrer regulären Diensterbringung den aktuellen Standort ihrer Nutzer abfragen.

Die verschiedenen Architekturen und Algorithmen zur privatsphärekonformen Verwendung ortsbezogener Anwendungen unterscheiden sich entlang mehrerer Dimensionen. So werden nicht nur verschiedene Einsatzszenarien fokussiert, sondern auch unterschiedliche Annahmen über die Kooperationsbereitschaft des Dienstanbieters getroffen. Letzterer stellt dabei meist zugleich selbst den mächtigen Angreifer dar, der naturgemäß über alle Anfragen der LBS-Nutzer Bescheid weiß und versucht, aus diesem Wissen zusätzliche Informationen über seine Nutzer abzuleiten. Ziel dieser Schutzmechanismen ist es daher, dem Nutzer die Verwendung ortsbezogener Dienste unter Einhaltung bestimmter Privatsphäregarantien gegenüber dem Dienstanbieter zu ermöglichen.

Während der Großteil der bekannten Verfahren auf dem Einsatz eines vertrauenswürdigen Dritten basiert, wurden insbesondere in der jüngeren Vergangenheit auch verschiedene clientseitig umsetzbare Mechanismen vorgestellt [251]. Die meisten Schutzmechanismen sind dabei auf sog. Snapshot-Anfragen ausgelegt. Die jeweiligen Autoren gehen entweder bewusst oder unbewusst davon aus, dass einzelne Anfragen durch einen Nutzer zeitlich so weit auseinanderliegen, dass diese nicht miteinander korrelieren. Die Übertragung solcher Verschleierungstechniken auf das Szenario kontinuierlicher LBS ist meist nicht ohne Weiteres möglich, z.B. weil sie aufgrund physikalischer Constraints bzgl. örtlicher und zeitlicher Zusammenhänge unbrauchbar werden und eindeutige Rückschlüsse auf die echten Bewegungen des Nutzers zulassen. Auch lassen sich je nach Komplexität des jeweiligen Verfahrens unterschiedlich starke Angreifer, Lösungsstrategien und Schutzziele ausmachen. Letztere reichen von Kommunikations-Anonymität und Vertraulichkeit bis hin zur Privatheit von Standortinformationen im eigentlichen Sinne und der Vermeidung von ortsbasierten Inferenz-Angriffen. Darüber hinaus gibt es einige spezialisierte Algorithmen, die sich nur für eine bestimmte Teilmenge von LBS einsetzen lassen.

Dass für den Schutz der Privatsphäre in solchen Systemen die bestehende Vielzahl an unterschiedlichen Herangehensweisen gerechtfertigt und nötig ist, zeigt z.B. die Arbeit von Burghardt et al. [39]: Die Autoren vergleichen unterschiedlich komplexe Schutzmechanismen hinsichtlich ihrer Verwendung durch echte Nutzer in deren normalen Alltag. Die Studie kommt u.a. zu dem Ergebnis, dass entsprechende Privacy-Maßnahmen tatsächlich verwendet werden, wenn sie zur Verfügung stehen und dem Nutzer bekannt sind, dass in unterschiedlichen Situationen unterschiedliche Mechanismen eingesetzt werden und kombiniert werden, und dass Nutzer allgemein solche Mechanismen bevorzu-

gen, die keine ständige Aufmerksamkeit oder große mentale Anstrengungen von ihnen verlangen.

Auch John Krumm kommt zu dem Schluss, dass ein einzelner Schutzmechanismus die individuellen Privatsphärebedürfnisse eines Menschen wohl nicht abdecken kann [150]. Die Vielzahl an seitdem neu entwickelten Verfahren zum Schutz von Standortinformationen unterstützt und bestätigt diese These. Entscheidend für die Akzeptanz und Adaption derartiger Schutzmechanismen durch den Endnutzer ist laut Gambs et al. zudem, dass diese Verfahren trotz Datenschutz eine qualitativ hochwertige Dienstnutzung gewährleisten [83].

Um einen umfassenden Überblick über den Stand der Technik zu geben, wird im Folgenden eine Auswahl wissenschaftlicher Arbeiten vorgestellt und kategorisiert, die sich mit dem Schutz der Privatsphäre in LBS beschäftigen. Ansätze wie der Einsatz von Bann-Zonen und die Verzerrung von Standortinformationen werden aufgrund der inhaltlichen Nähe zu den Inhalten von Kapitel 4 dort besprochen.

2.2.3.1 Verfahren basierend auf k-Anonymität

Gruteser et al. [104] übertragen das zuvor beschriebene Prinzip der k-Anonymität auf den Schutz der Privatsphäre in ortsbezogenen Diensten, um Anonymität bei der Formulierung von LBS-Anfragen zu gewährleisten. Die Grundannahme ist hierbei, dass der Dienstanbieter die aktuellen Positionen aller Nutzer kennt oder in Erfahrung bringen kann. Diese Prämisse liegt fast allen Arbeiten zugrunde, die auf den Schutz der Privatsphäre auf Basis der k-Anonymität setzen [95]. Shokri et al. weisen zudem darauf hin, dass der Wert von k keinen Einfluss auf den Grad an Standortanonymität besitzt [220].

Als privat werden hingegen die Nachrichteninhalte der LBS-Anfragen angesehen, wie z.B. die Suche nach einem *Facharzt in meiner Nähe*, da diese Details über die anfragende Person verraten. Zudem betrachten Gruteser et al. erstmalig das Potential von Standortdaten zur Identifizierung von Nutzern: So soll insbesondere verhindert werden, dass eine anonym formulierte LBS-Anfrage allein aufgrund des darin übermittelten Standorts eindeutig einer bestimmten Person zugeordnet werden kann. Letzteres ist möglich, wenn dieser Ort z.B. das Zuhause des anfragenden Nutzers beschreibt oder weil bereits kurz zuvor vom selben Ort eine nicht-anonymisierte Anfrage beobachtet wurde, die als privatsphäretechnisch unbedenklich eingestuft und daher im Original übertragen wurde. Bezogen auf die oben beschriebenen Quasi-Identifier muss der Standort einer LBS-Anfrage daher derart vergröbert werden, dass stets k verschiedene Nutzer als Urheber der Nachricht in Frage kommen.

Für die Umsetzung ihres Systems verwenden die Autoren einen *Anonymization Server*, der die Positionen aller Nutzer kennt und dementsprechend als TTP betrachtet wird. Für diesen und alle weiteren vorgestellten Ansätze wird stets angenommen, dass der Kommunikationskanal zwischen Client und TTP ausreichend geschützt und abhörsicher ist. Neben dem Entfernen von Nutzer-IDs und womöglich verräterischen Netzwerkinformationen ist die-

ser insbesondere für die Verschleierung der Kommunikationsinhalte auf Basis der k-Anonymität zuständig. Um eine hohe Qualität der Dienstnutzung zu gewährleisten, schlagen die Autoren ein einfaches Verfahren zur Erzeugung einer Verschleierungsmenge vor, die einen räumlich möglichst kompakten Bereich abdeckt: Stellt ein Nutzer p eine LBS-Anfrage, wird der gesamte Raum gemäß des Quadtree-Verfahrens [208] hierarchisch aufgeteilt. Befinden sich mehr als k Nutzer in dem Quadranten von p, wird die aktuelle Zelle so lange in vier gleich große Quadranten aufgeteilt, bis das geforderte k unterschritten ist. Befinden sich auf der aktuellen Hierarchiestufe weniger als k Nutzer darin, wird der übegeordnete Quadrant ermittelt. Die resultierende Zelle wird schließlich dem LBS übermittelt, der – selbst, wenn er die aktuellen Standorte aller Nutzer kennt – in vielen Fällen nicht nachvollziehen kann, wer von mindestens k Nutzern diese Suche gestartet hat. Ein Problem, das aus der inhomogenen Verteilung der Nutzerstandorte auftritt, wird weiter unten behandelt.

Als Alternative zur räumlichen Verschleierung, die u.U. sehr große Zonen erzeugt, schlagen Gruteser et al. den zusätzlichen Einbezug der zeitlichen Komponente vor. Dadurch werden LBS-Anfragen mit hoher räumlicher Auflösung ermöglicht, die jedoch so lange hinausgezögert werden, bis k Nutzer an diesem Ort anzutreffen waren. Auch Kombinationen aus räumlicher und zeitlicher Vergröberung werden vorgeschlagen. Grundsätzlich stellt dieses Vorgehen jedoch ausschließlich eine Lösung für zeitpunktunabhängige und vor allem nichtinteraktive Dienste dar, bei denen der Nutzer nicht auf eine Antwort wartet, wie z.B. die Crowdsourcing-basierte Erkennung von Straßenschäden [77], bei der eine mehrstündig verzögerte Datenübermittlung unproblematisch ist.

Basierend auf den grundlegenden Prinzipien dieses Vorreitersystems wurde in der Folge eine ganze Reihe ähnlich ausgerichteter, mit verschiedenen Ergänzungen und Optimierungen versehener Verfahren vorgestellt: Gedik et al. beschäftigen sich bei *Clique Cloak* mit alternativen Strategien für die Erzeugung von Verschleierungszonen sowie Möglichkeiten zur Personalisierung von Privatsphäreeinstellungen. So haben alle Teilnehmer dieses Systems die Möglichkeit, ihren individuellen Bedürfnissen entsprechend sowohl unterschiedliche Werte für k als auch Obergrenzen für die maximal tolerierbare räumliche und zeitliche Vergröberung ihrer Anfragen festzulegen. Analog zu [104] gehen die Autoren davon aus, dass die Erzeugung des Anonymitätssets sowohl entlang der räumlichen Dimension als auch durch zeitliche Verzögerung von Anfragen erreicht werden kann.

Im Gegensatz zu der ersten Arbeit wird hier jedoch vorausgesetzt, dass ein Nutzer nur dann Teil eines Anonymitätssets sein kann, wenn er selbst eine Anfrage stellt. Die Anonymitätsmenge wird nicht auf Basis aller Nutzer erzeugt, sondern nur auf der Teilmenge, die aktuell eine LBS-Anfrage stellen. Grundsätzliche Bedingung für die erfolgreiche Erzeugung einer Verschleierungsregion ist daher, dass sich mindestens k Anfragen finden lassen müssen, deren Ursprung jeweils innerhalb der im Rahmen der übrigen Anfragen angegebenen maximalen räumlichem Toleranz liegt. Gelingt es, k solche Nachrichten zu

finden, wird das minimal umgebende Rechteck (MUR) um die Ausangspunkte der beteiligten Anfragen gelegt und alle Anfragen an den LBS übermittelt. Anstelle der echten Nutzer-IDs und der realen Standorte übermittelt die zentrale *Message Perturbation Engine* anonymisierte Anfragen an den LBS, die das soeben ermittelte MUR als Ortsangabe enthalten. Lassen sich hingegen nicht ausreichend viele Nachrichten in der Umgebung eines anfragenden Nutzers finden, wird die Anfrage nach Ablauf einer vom Nutzer wählbaren Zeitspanne Δt erfolglos verworfen. Das System besitzt die positive Eigenschaft, dass sich bei der Erstellung einer erfolgreichen Anonymitätsmenge für eine Nachricht gleich $k-1$ weitere Anfragen automatisch mit verschleiern und beantworten lassen.

Gleichzeitig weist *Clique Cloak* aufgrund der maximal erlaubten räumlich-zeitlichen Abweichungen bei der Standortverschleierung schon bei geringen Parameterbelegungen wie $k = 5$ und kleinen Anfrageintervallen in den Experimenten der Verfasser nur geringe Erfolgsraten von ca. 50 % auf. Ein weiterer Nachteil dieses Verfahrens ist in der MUR-basierten Zonenerstellung zu sehen. Einem Angreifer, der vorab nicht über die exakten Standorte der Nutzer informiert ist, lernt bei der Verwendung dieses Vorgehens eindeutige Hinweise auf die genauen Aufenthaltsorte, da sich diese für den Extremfall von $k = 2$ zwangsweise alle auf den Grenzen der Verschleierungszone befinden müssen. Für größere Werte von k trifft dies in abgeschwächter Form ebenfalls zu.

Anders als *Clique Cloak* führen Mokbel et al. [177] in *Casper* mit A_{min} einen zusätzlichen, vom Nutzer individuell einstellbaren Parameter neben k ein, der keinen maximalen Toleranzwert, sondern eine Mindestanforderung an die Größe der Fläche der Verschleierungsregion stellt. Bei der wie in [104] baumbasierten Auswahl passender Verschleierungszonenen wird daher nicht nur darauf geachtet, dass sich in der entsprechenden Zelle k Nutzer befinden, sondern zudem auch eine gewisse Mindestgröße eingehalten wird. Darüber hinaus konzipieren die Autoren den sog. *Privacy-Aware Query Processor*, der beim LBS-Anbieter installiert wird und dafür sorgt, dass dieser überhaupt dazu in der Lage ist, Anfragen wie z.B. eine POI-Suche auf Basis k-anonym verschleierter Regionen zu verarbeiten, anstelle der eigentlich üblichen punktbasierten Anfragen. Dieses Zusatzmodul sorgt auch dafür, dass dem Nutzer nicht eine unmittelbare Antwort präsentiert wird, sondern eine Liste an Antwortkandidaten, die je nach Privatsphäreeinstellungen des Nutzers beliebig groß werden kann, dafür aber so erstellt wird, dass sie stets das optimale Ergebnis beinhaltet. Der Client selbst wählt anhand seiner tatsächlichen Position schließlich die für ihn zutreffende Antwort aus dieser Kandidatenmenge aus. Ein konzeptioneller Nachteil dieses Systems ist damit allerdings, dass neben der Existenz einer vertrauenswürdigen TTP zusätzlich die umfangreiche Kooperation des LBS-Anbieters vorausgesetzt wird.

In [48] stellen Chow und Mokbel eine weitere Ergänzung ihres Anonymisierungsservers vor, bei dem der Nutzer nun zwei verschiedene Parameter k_l und k_q angeben kann. Damit wollen die Autoren eine explizite Unterscheidung zwischen *Location Privacy* und *Query Privacy* ermöglichen, da sie nicht da-

von ausgehen, dass die Nutzerstandorte dem Angreifer ohnehin bekannt sind. Stattdessen soll nun auch der Standort selbst den Privatsphärebedürfnissen des Nutzers entsprechend geschützt werden. Die Autoren unterscheiden zu diesem Zweck zwischen zwei verschiedenen Nutzungsmodi: Im Rahmen von *Public location with private query* werden $k_l = 1$ und $k_q > 1$ gesetzt. Für Nutzer, die kein Problem mit der Freigabe ihres genauen Standorts haben, jedoch nicht möchten, dass der Dienstanbieter die von ihnen formulierten Anfragen eindeutig zuordnen kann, bietet sich dieser Modus an. Im alternativen Modus *Private location with private query* soll auch der Standort selbst verschleiert werden. Die beiden Parameter können dabei individuell und unabhängig von einander gesetzt werden. Dies ermöglicht eine größere Flexibilität bei der Angabe von Privatsphärepräferenzen, da der Nutzer unterschiedlich große Anonymitätsmengen für Standortupdates und LBS-Anfragen definieren kann.

Unter der für eine sinnvolle Anonymisierung notwendigen, stets vorausgesetzten Bedingung $k_l \leq k_q$ ermittelt der Anonymisierungsserver im Folgenden sog. *robuste* Verschleierungszonen. Unabhängig vom gewählten Modus werden diese so erzeugt, dass sich auch unter Kenntnis der genauen Standorte eines Nutzers und bei der wiederholten Formulierung von Anfragen an den LBS-Anbieter die Zuordnung von Queries zu Nutzern nicht nachvollziehen lässt. Um die Identität eines Nutzers, der kontinuierliche Anfragen an einen LBS schickt, über die Zeit hinweg zu schützen, merkt sich der Anonymisierungsserver das initiale Set an Nutzern, die sich zum Zeitpunkt der ersten Anfrage in der Anonymitätsmenge des Nutzers befunden haben. Bei jeder weiteren Anfrage werden die Grenzen des Verschleierungsregion den aktuellen Standorten all dieser Nutzer sukzessive angepasst, um die Größe des in Frage kommenden Anonymitätssets konstant bei k_q zu halten. Dies verhindert zwar effektiv die Identifikation eines Nutzers durch das Tracking einer kontinuierlichen Anfrage, führt aufgrund der Nutzermobilität aber zu stetig anwachsenden Zonen und im selben Maße verschlechterter Dienstqualität. Bettini et al. skizzieren mit der *historical k-Anonymity* [29] ein ähnliches Verfahren, das nicht nur einzelne kontinuierliche Anfragen, sondern alle LBS-Anfragen eines via Pseudonym identifizierten Nutzers auf diese Weise schützen soll. Aus Sicht der Privatsphäre sind derartige Ansätze den vorigen Verfahren überlegen, da sie die Korrelation aufeinanderfolgender Anfragen berücksichtigen. Problematisch bleibt jedoch die Rolle der TTP, v.a. wenn diese wie in [29] selbst die historischen Anfragen und Standorte aller Teilnehmer speichert.

Ein weiterer Nachteil von [104] und auch [177] ist, dass LBS-Anfragen aufgrund der bekannten Nutzerpositionen u.U. verräterisch sein können und sich eindeutig ihrem Urheber zuordnen lassen. Dieses Phänomen wird durch die inhomogene Verteilung der Nutzerstandorte im Raum hervorgerufen. Teilnehmer, die sich in kaum besuchten Regionen aufhalten werden zu Ausreißern, die charakteristisch große Verschleierungszonen erzeugen. Auch hierfür wurden unterschiedliche Lösungsansätze präsentiert. So stellt z.B. [48] die Forderung nach dem *k-sharing* von solchen Verschleierungsregionen auf. Hierfür wird voraus-

gesetzt, dass eine erzeugte Verschleierungszone nicht nur die Standorte von k Nutzern umfassen muss, sondern dass diese Zone auch für k verschiedene Nutzer verwendet wird. Die dafür notwendige Gruppierung von Nutzern führt der Anonymisierungsserver ad-hoc bei Eingang einer LBS-Anfrage durch, indem er Nutzer entweder einer bestehenden Gruppe $\mathcal{G}$ zuordnet, die seine aktuelle Position enthält oder eine neue Gruppe erzeugt, indem er die $k-1$ dem Nutzer am nächsten liegenden, bis dato ebenfalls ungruppierten Teilnehmer ermittelt. Im Gegensatz zu [104] wird fortan der Umriss der von dieser Gruppe an Benutzern besuchten Region für alle diese Nutzer als Verschleierungszone an den LBS übermittelt. Obwohl die Autoren unmittelbar auf das oben beschriebene Problem abzielen, scheint es fraglich, ob eine derartige Gruppenerstellung das Problem der Ausreißer-Positionen wirklich löst. Immerhin würde auch bei diesem Verfahren ein abseits positionierter Nutzer eine charakteristische und völlig andere Zone erzeugen als ein Teilnehmer, der sich in einer hochfrequentierten Umgebung befindet. An dieser Form der deterministischen Gruppenerzeugung auf Basis kürzester Distanzen ist zu kritisieren, dass hierbei durch Reverse-Engineering direkte Rückschlüsse auf eine u.U. deutlich eingegrenzte Kandidatenmenge möglich sind, die als Ausgangspunkt dieser Gruppe und damit auch als Urheber der LBS-Anfrage in Frage kommen.

Da in der Sicherheits- und Datenschutz-Literatur üblicherweise davon ausgegangen wird, dass der Angreifer die verwendeten Verfahren und Parametrisierungen kennt, muss eine solche Bestimmbarkeit oder Eingrenzbarkeit des anfragenden Nutzers anhand der übermittelten Regionsgrenzen verhindert werden. Effektiven Schutz vor derartigen Angriffen – weiterhin unter Verwendung der k-Anonymität als grundlegender Schutzmechanismus – bietet das *Hilbert Cloak*-Verfahren von Kalnis et al. [136]. Der hierbei eingesetzte Algorithmus stellt zum einen wie zuvor sicher, dass für mindestens k verschiedene Nutzer dieselbe Verschleierungszone an den LBS übermittelt wird. Zudem sollen die erzeugten Regionen erneut möglichst klein sein, um eine hohe Dienstqualität zu erlauben. Darüber hinaus sorgt er jedoch auch dafür, dass jeder der darin enthaltenen Nutzer auch tatsächlich mit derselben Wahrscheinlichkeit genau diese Zonenumrisse erzeugen würde.

Um dieses Ziel zu erreichen, verfolgen die Autoren eine Strategie, die die *Reziprozität* von Verschleierungszonen gewährleistet: Da eine Ermittlung der optimalen reziproken Partitionierung NP-schwer ist [136], legen die Autoren zunächst ein geeignet feinmaschig gewähltes Gitternetz über den kompletten vom Anonymisierungsserver abgedeckten Raum. Aufbauend auf diesem Grid wird anschließend die sog. *Hilbert-Kurve* [118] eingesetzt, um eine Abbildung des zweidimensionalen Raums, in dem sich die Teilnehmer des Systems bewegen auf eine eindimensionale Darstellung zu reduzieren.

Die Verwendung der Hilbert-Kurve hat die einer Erzeugung flächenmäßig möglichst kleiner Verschleierungszonen zuträgliche Eigenschaft, dass für einen Großteil der Punkte, die im zweidimensionalen Raum nah beisammen liegen, dies auch im eindimensionalen noch gilt. Die Nutzer werden schließlich entspre-

chend der Reihenfolge, in der sie auf der Kurve liegen, aufsteigend in einem B+-Baum indexiert. In [92] verallgemeinern die Autoren dieses Prinzip noch für die Erstellung reziproker Anonymitätssets in gängigen spatialen Indexstrukturen wie R-Bäumen oder Quadtrees. Abhängig vom jeweils geforderten Wert für k werden alle Nutzer nun gleichmäßig in *Buckets* der Länge k eingeteilt. Um die Privatsphäre einzelner Nutzer nicht zu gefähren, kann dabei das letzte Bucket je nach Gesamtzahl der Teilnehmer bis zu $2k - 1$ Mitglieder haben. Geht eine LBS-Anfrage von einem bestimmten Nutzer ein, wird dessen Indexposition ermittelt, das MUR der diesem Bucket zugehörigen Nutzer gebildet und als Verschleierungszone an den LBS übermittelt. Der *Hilbert Cloak*-Ansatz löst damit nicht nur effektiv das Problem der inhomogenen Verteilung der Nutzerstandorte, sondern lässt sich dank der Dimensionsreduktion auf nur einen einzelnen Wert pro Benutzer aus Sicht des Anonymisierungsservers auch überaus effizient für eine Vielzahl an gleichzeitigen Teilnehmern umsetzen. Auch dieses Verfahren setzt jedoch spezielle, rechenintensive Mechanismen zur Anfragebearbeitung voraus, die der LBS-Anbieter zur Verfügung stellen muss, um auch regionsbasierte Anfragen korrekt beantworten zu können.

Im Rahmen von *PrivacyGrid* [18] erweitern Bamba et al. die bislang rein k-Anonymitäts-basierte Verschleierung um das ebenfalls aus dem Bereich der Datenbankforschung stammende Prinzip der l-Diversität. Damit soll einer markanten Schwachstelle der bisherigen Verfahren begegnet werden. Unter der Annahme, dass der Angreifer die Nutzerstandorte nicht im Vorfeld kennt, müssen diese ebenfalls geschützt werden. Schließlich gibt ein Nutzer ein ähnliches Maß an persönlichen Informationen preis, wenn er nach der nächsten Facharztpraxis sucht oder von dort eine „harmlose" LBS-Anfrage stellt. Das Problem liegt darin begründet, dass es sich bei dem Standort eines Nutzers zugleich um einen Quasi-Identifier und um ein sensibles Attribut handelt. Das Prinzip der k-Anonymität ist daher genau dann wirkungslos für den Schutz von Standortinformation, falls sich alle k Nutzer an demselben (semantischen) Ort aufhalten wie z.B. ein Krankenhaus, ein Bürogebäude oder eine Schule.[5] Als Lösung führen Machanavajjhala et al. das Prinzip der l-Diversität ein [170]. In der einfachsten Form wird dabei gefordert, dass alle sensiblen Attribute innerhalb einer Anonymitätsmenge mindestens l verschiedene Werte annehmen müssen. Übertragen auf den Schutz von Standortinformationen bedeutet dies, dass die k Mitglieder der Anonymitätsmenge so gewählt werden müssen, dass sich diese an mindestens l verschiedenen Orten aufhalten und somit kein unmittelbarer Rückschluss auf die Standorte eines oder gar aller Teilnehmer möglich sind. In [258] werden komplexere Formen der l-Diversität betrachtet, die auf Seite des Angreifers auch Hintergrundwissen über die statistische Verteilung der Inhalte von LBS-Anfragen von verschiedenen Standorten aus annehmen.

Der gesamte vom System abgedeckte Raum wird dabei erneut in ein Git-

[5] Analog verhält es sich bei der Veröffentlichung einer medizinischen Datenbank: Leiden alle k Nutzer an derselben Krankheit, war die Elimination der Quasi-Identifier wirkungslos für den Schutz der Privatsphäre der Betroffenen.

ternetz eingeteilt. Für jede Zelle dieses Grids wird die Anzahl der darin enthaltenen mobilen Objekte, d.h. die Nutzer, die sich dort aufhalten, dynamisch verwaltet. Zudem muss zur Umsetzung der l-Diversität für jede Zelle bekannt sein, wie viele statische Objekte, also verschiedene semantische Orte in der Zelle existieren. Neben den Parametern k und l können die Nutzer auch hier individuelle Obergrenzen für die maximal tolerierte räumliche und zeitliche Vergröberung ihrer LBS-Anfragen angeben. Die Verwendung der gridbasierten Aufteilung des Raumes ergibt dabei natürlich nur Sinn, wenn die von den Teilnehmern tolerierbare räumliche Vergröberung um ein Vielfaches größer als die Seitenlänge der Gitterzellen ausfällt. Aus der Kombination dieser vier Vorgaben erzeugt der zentrale *Anonymization Service* gemäß unterschiedlicher Algorithmen rechteckige Verschleierungsregionen, die sowohl k Nutzer als auch l verschiedene Orte abdecken. Leider werden in [18] jedoch keine Angaben gemacht, wie solche semantischen Orte ermittelt werden können oder was als sinnvoller Wert für l erachtet werden kann. Dementsprechend wird der Einfluss der l-Diversität auch nicht in den durchgeführten Experimenten evaluiert. Es ist zu erwarten, dass die erzeugten Verschleierungsregionen dadurch größer werden und die Anzahl an nicht erfüllbaren Anfragen zunimmt.

Insgesamt führt auch dieser Ansatz zu einem gewissen Anteil an Anfragen, die aufgrund nicht erfüllbarer Bedingungen verworfen werden oder die erst nach einer gewissen zeitlichen Verzögerung übermittelt werden können, was für interaktive LBS ungeeignet ist. Darüber hinaus setzt auch der *PrivacyGrid*-Ansatz die direkte Kooperation des LBS-Anbieters voraus: Dieser muss in Form von spezialisierten Dienstschnittstellen und durch die Implementierung geeigneter, verhältnismäßig rechenaufwändiger Lookup-Mechansimen die korrekte Beantwortung regionsbasierte Anfragen, um dem Nutzer in der Liste der Antwortkandidaten auch stets das optimale Ergebnis mitliefern zu können.

Während die bisherigen Ansätze das abgedeckte Gebiet jeweils als Freifläche betrachten, wurden auch einige Verfahren vorgestellt, die die Umgebung als Straßengraph modellieren [152, 182, 242]. Diese unterscheiden sich im Wesentlichen jedoch nur durch die nun graphbasierten Verfahren zur Ermittlung der k nächsten Benutzer und Verfahren zur effizienten graphbasierten POI-Suche anstelle der euklidischen Distanz von den anderen Verfahren.

Alle der bisher vorgestellten Ansätze zur Herstellung von k-Anonymität bei der LBS-Nutzung kommen nicht ohne die Existenz eines vertrauenswürdigen Dritten aus, der die genauen Positionen aller Nutzer kennt. Um diesen grundsätzlichen Kritikpunkt an TTP-basierten Lösungen zu vermeiden, stellen verschiedene Arbeiten dezentrale Ansätze zur k-Anonymität in ortsbasierten Diensten auf Basis einer direkten Kommunikation der Teilnehmer untereinander und einer verteilter Erzeugung von Verschleierungszonen vor [49, 91, 264, 126]. Auch Chow et al. [48] verweisen darauf, dass sich ihr Ansatz im Rahmen einer solchen dezentralen Architektur umsetzen ließe.

Kommunikationsanonymität wird in diesen Szenarien dadurch erreicht, dass ein zufälliger Peer aus der ermittelten Gruppe von k Nutzern die verschleierte

Anfrage an den LBS-Anbieter weiterleitet. Durch das im Rahmen Peer-to-Peer-basierter Funknetze grundsätzlich auftretende Problem begrenzter Sende- und Empfangsreichweiten können Nutzer in wenig besuchten Gebieten überhaupt keine verschleierten LBS-Anfragen stellen. Aufgrund der lokal begrenzten Sicht auf die verfügbaren Nutzer stellt sich in dicht besiedelten Bereichen erneut das Problem der zurückrechenbaren Regionsgrenzen. Diese erlauben u.U. Rückschlüsse auf den Urheber der LBS-Anfrage, der sich in diesem Fall im Zentrum der übermittelten Verschleierungszone befindet.

Das eigentliche Problem, einer fremden Partei Informationen über den eigenen Standort mitteilen zu müssen, wird aus Sicht der Privatsphäre nur verlagert: Anstatt einem zentralen Server, der als vertrauenswürdig definiert wird, muss im Falle von Peer-to-Peer-basierten Lösungsansätzen einer Menge an unbekannten Nutzern vertraut werden. Hierbei besteht z.B. die Gefahr der Kollaboration bösartiger Nutzer mit dem LBS-Anbieter oder der „Verschwörung" mehrerer Teilnehmer gegen einen einzelnen Nutzer. Ob diese Vervielfachung der möglichen Angreifer eine echte Verbesserung darstellt, scheint fraglich. Hu et al. [126] begegnen diesem Problem durch die ausschließliche Verwendung von Informationen über die mittels Funkreichweite ermittelte paarweise Nähe unter den teilnehmenden Peers anstelle exakter Koordinaten und den Einsatz von Techniken aus dem Bereich der *Secure-Multiparty Computation* [96].

Der Einsatz der k-Anonymität für den Datenschutz in LBS geht darüber hinaus mit einigen elementaren Nachteilen einher: Vor dem Hintergrund der angestrebten Kommunikationsanonymität ist grundsätzlich keiner dieser Ansätze dazu in der Lage, eine pseudonyme Dienstnutzung zu ermöglichen. Damit wird auch die Klasse der personalisierten ortsbezogenen Dienste automatisch ausgeschlossen, da z.B. ein Dienst, der einen Nutzer proaktiv über Freunde in seiner Nähe informiert, derartige Pseudonyme für seine Diensterbringung zwangsläufig benötigt. Auch lassen sich solche LBS nicht nutzen, die wie die Online-Routenplanung und -Navigation auf exakte Standortinformationen angewiesen sind. Zudem geht bis auf [18] keiner dieser Ansätze davon aus, dass der Angreifer über Hintergrundwissen wie die Verteilung semantischer Orte auf der Karte verfügt.

Xu et al. argumentieren, dass die Wahl eines passenden Wertes für k schwer fällt, da es wenig intuitiv ist, ein abstraktes Bedürfnis wie der Wunsch nach Privatsphäre in eine Zahl zu übersetzen. Sie schlagen in [257] deshalb eine „gefühlsbasierte" Herangehensweise vor: Anstatt einen Zahlwert anzugeben, definiert der Nutzer hier einen öffentlichen Platz, der ihm als Ortsangabe ausreichend anonym erscheint. Die TTP ermittelt daraufhin anhand der Popularität dieses Platzes die entsprechende Parameterbelegung und wählt die Anonymitätsmenge fortan auch für andere Orte dementsprechend aus.

In [59] argumentiert Dewri gegen die Verwendung von Verschleierungsregionen. Er schlägt stattdessen eine Koordinatentranslation der Standorte von k benachbarten Nutzern vor. Dewri geht davon aus, dass ein Angreifer ohnehin grobes Vorwissen über die Aufenthaltsorte von Nutzern besitzt, das sich

ebenfalls in Flächen ausdrückt. Die Verwendung eines Anonymizers und die Angabe von MUR-basierten Verschleierungszonen bei allen bisher vorgestellten Verfahren ermöglichen demnach ausschließlich die Herstellung von Kommunikationsanonymität, nicht jedoch den Schutz der Standorte. Das Problem hinsichtlich der Standort-Privatsphäre entsteht dadurch, dass die erzeugte Verschleierungszone die grob bekannten Nutzerstandorte u.U. nicht vollständig überdeckt, sondern Schnittflächen bildet, mit deren Hilfe der Angreifer sein Wissen über die Aufenthaltsorte der Nutzer genauer eingrenzen kann.

Dewris Berechnung der k-anonymen Koordinaten-Transformation leitet sich von dem Prinzip der *Differential Privacy* [71] ab und sorgt dafür, dass jeder der k Nutzer mit fast derselben Wahrscheinlichkeit diese Koordinate erzeugt haben kann. Zu diesem Zweck ermittelt die TTP den maximalen Abstand r zwischen den k gewälten Nutzern, der als Skalenparameter einer Laplace-Verteilung verwendet wird. Aus dieser Verteilung werden getrennt für die X- und Y-Koordinaten der Standorte Werte gezogen, aus denen sich die Translation des eigentlichen Punktes ergibt. Dies gewährleistet, dass sich die Wahrscheinlichkeiten, dass zwei verschiedene Punkte mit maximalen Abstand r den Verschleierungspunkt erzeugt haben, maximal um den Faktor e^{ϵ} unterscheiden. ϵ ist dabei ein Parameter, der systemweit angegeben wird.

Im Gegensatz zu allen anderen Verfahren wird hiermit also keine Verschleierungsregion, sondern nur eine intelligent gewählte Koordinate an den LBS übertragen. Dies hat den Vorteil, dass der Angreifer keine zusätzlichen Informationen über die Aufenthaltsorte der Nutzer lernen kann und dass der LBS keine spezialisierten Schnittstellen zur Bearbeitung von regionsbasierten Anfrage zur Verfügung stellen muss. Ein Nachteil ist jedoch, dass hierbei nun nicht mehr das korrekte Ergebnis der LBS-Anfrage garantiert werden kann.

Größter Kritikpunkt ist auch hier der Einsatz der TTP selbst, die allwissend über die Aufenthaltsorte all ihrer Nutzer informiert ist und daher ein lukratives Angriffsziel darstellt. In [258] wird zudem in Frage gestellt, ob Nutzer tatsächlich dazu bereit sind, einen solchen *Location Server* kontinuierlich mit aktuellen Standortaktualisierungen zu versorgen, die sowohl Energie als auch Bandbreite verbrauchen, obwohl sie selbst gerade keine Anfrage stellen. Auch scheint fraglich, wer eine solche Komponente aus welchem Interesse heraus betreiben sollte oder warum man diesem Anbieter prinzipiell mehr Vertrauen entgegen bringen sollte als einem LBS selbst. So hat sich – sicherlich begünstigt durch die enorme technologische Weiterentwicklung der Rechenleistung mobiler Endgeräte selbst – kein solcher vertrauenswürdiger Location Server etabliert. Stattdessen kommunizieren die auf mobilen Endgeräten installierten LBS-Anwendungen stets direkt mit den Servern des Anbieters, ohne dass dabei eine vertrauenswürdige dritte Partei vermittelnd eingreifen kann.

Aus diesem Grund werden im Folgenden alternativ ausgerichtete Schutzmechanismen für ortsbezogene Dienste vorgestellt, die u.a. die Vermeidung einer TTP oder den Einbezug von Hintergrundwissen in den Vordergrund stellen.

2.2.3.2 Verwendung von Dummy-Locations

Einen clientseitig umsetzbaren Mechanismus zum Schutz von Standortinformationen in ortsbezogenen Diensten stellt der Einsatz sog. *Location Dummies* dar. Diese Herangehensweise folgt in ihrem Kern einer ähnlichen Definition von Privatsphäre und greift – wenn auch mit anderer Zielsetzung und abweichender Definition des Anonymitätssets – ebenfalls das Prinzip der k-Anonymität auf. Allerdings wird hierbei nicht der Urheber einer LBS-Anfrage unter k realen Nutzern versteckt, sondern der tatsächliche Standort des Nutzers geheim gehalten, während seine Identität z.B. über ein Pseudonym bekannt ist. Grundlegende Annahme ist hierbei, dass der Angreifer nicht im Vorfeld über die aktuellen Aufenthaltsorte seiner Teilnehmer Bescheid weiß, sondern diese ausschließlich aus den übermittelten LBS-Anfragen ableiten kann. Im Falle eines typischen LBS-Anbieters, der nicht der Netzwerk-Provider selbst oder ein physischer Angreifer ist, der einzelne Nutzer verfolgt, ist diese Betrachtungsweise überaus realistisch. Zusätzlich setzen alle folgenden Verfahren voraus, dass der Angreifer nicht anhand von Seitenkanälen auf den echten Standort des Nutzers schließen kann, wie z.B. bei der Ortung der IP-Adresse des mobilen Endgeräts.

Um den Aufenthaltsort des Nutzers zu verschleiern, werden im Rahmen einer LBS-Anfrage nun $k-1$ Dummy-Positionen generiert, für die zusätzlich zu der Anfrage mit dem echten Nutzerstandort ebenfalls Queries formuliert und an den LBS übermittelt werden. Somit soll auch hier die Wahrscheinlichkeit, dass der Angreifer den tatsächlichen Aufenthaltsort eines Nutzer korrekt ermitteln kann, auf maximal $\frac{1}{k}$ begrenzt werden. Sobald das Endgerät des Nutzers die Antworten auf die k Anfragen erhält, kann es aufgrund der alleinigen Kenntnis des tatsächlichen Standorts die Kandidatenmenge entsprechend filtern und dem Nutzer die optimale Antwort präsentieren. Im Gegensatz zu den zuvor beschriebenen Systemen sind derartige Verfahren unabhängig von den aktuellen Aufenthaltsorten anderer Nutzer und lassen sich ohne den Einsatz eines vertrauenswürdigen Dritten umsetzen.

Eines der ersten, vergleichsweise einfachen, dummybasierten Verfahren für sporadische LBS-Anfragen stammt von Duckham und Kulik [69]. Hierbei werden die Dummy-Positionen als Knoten im Strassennetz ausgewählt. Diese müssen nicht zusammenhängend sein und je mehr solche Punkte ausgewählt werden, desto größer stufen die Autoren den Grad an Privatsphäre ein. Bei ungeschickter Dummy-Wahl wird jedoch vermutet, dass diese angegriffen werden kann und es daher ausgefeilter Verfahren zur Dummy-Erzeugung bedarf.

Andere Systeme versuchen explizit auch die privatsphärekonforme Nutzung kontinuierlicher LBS durch einen mobilen Benutzer über mehrere, miteinander korrelierte Standortupdates hinweg zu ermöglichen. Vor diesem Hintergrund müssen nicht nur einzelne Dummy-Positionen, sondern ganze Dummy-Routen erzeugt werden, die in sich plausibel und nicht als Fälschung erkennbar sind.

Den ersten dummybasierten Ansatz zum Schutz der Privatsphäre in solchen kontinuierlichen ortsbezogenen Diensten präsentieren Kido et al. in [144]. Hierbei werden bei der initialen Formulierung einer LBS-Anfrage durch den Be-

nutzer vom Algorithmus $k-1$ zufällige Dummy-Positionen auf der gesamten vom LBS abgedeckten Fläche ausgewählt. Bei der Erstellung solcher Dummy-Routen muss verhindert werden, dass der LBS-Anbieter echte Pfade eines Nutzers und verschiedene Dummy-Trajektorien auf triviale Weise voneinander unterscheiden kann. Insbesondere müssen zusammenhängende Trajektoren entstehen, bei denen aufeinanderfolgende Standorte jeweils so nah beisammen liegen, dass sie von der zuvor übermittelten Position in der seit dem letzten Update vergangenen Zeitspanne auch tatsächlich erreicht werden können.

Zur plausiblen Fortführung der simulierten Standortangaben schlagen die Autoren zwei unterschiedliche Verfahren vor. Dies wird durch den Einsatz der *Moving in a Neighborhood*-Strategie erreicht, die den nächsten Dummy zufällig in der Nachbarschaft des zuvor verwendeten erzeugt. Beim *Moving in a Limited Neighborhood*-Verfahren gehen die Autoren davon aus, dass das Endgerät des Nutzers – ohne zu spezifizieren, wie dies umgesetzt werden kann oder welchen Einfluss dies auf die Privatsphäre der Nutzer hat – über die tatsächlichen Standorte der anderen Nutzer informiert ist. Hierbei werden zunächst wie zuvor die Folgepositionen zufällig aus der Nachbarschaft des zuletzt gesendeten Updates ausgewählt. Zudem wird nun jedoch darauf geachtet, dass stets solche Dummies erzeugt werden, die in wenig stark besuchte Regionen fallen. Trifft dies auf den aktuellen Kandidaten nicht zu, wird eine neue zufällige Position erzeugt. Damit soll verhindert werden, dass sich die gefälschten Standortverläufe aufgrund der großen Zahl an echten Bewegungsmustern in dieser Region durch ihre Zufälligkeit und ihr damit von der Norm abweichendes Verhalten als simulierte Trajektorie zu erkennen zu geben. Bei jeder neuen Anfrage an den LBS werden schließlich sowohl der echte aktuelle Standort des Nutzers als auch die derart fortgeführten Dummy-Positionen an den Dienstanbieter übermittelt.

Insgesamt gestalten sich diese Verfahren relativ naiv und berücksichtigen bei der Erzeugung von Dummy-Trajektorien weder das zugrundeliegende Straßennetz noch ein realistisches Bewegungsmodell. Stattdessen werden auf einer als Freifläche angesehenen Karte gemäß eines zufälligen Prozesses, der ausschließlich von der zuletzt erzeugten Dummy-Position abhängt, neue gefälschte Standortangaben erzeugt. Dank der verwendeten Algorithmen liegen diese zwar in der Nähe des zuletzt übermittelten Punktes, können jedoch unwahrscheinliche Orte (z.B. Wege abseits befahrbarer Gebiete oder Routenendpunkte in unbewohnten Regionen) und unnatürliche Trajektorien (z.B. mehrfaches Oszillieren zwischen zwei Richtungen) erzeugen, die einem Angreifer mit Kartenwissen in vielen Fällen nicht standhalten können und sich unmittelbar als Fälschung erkennen lassen. Ist ein Angreifer wie z.B. der LBS-Anbieter dazu in der Lage, alle übermittelten Standortupdates eines Nutzers zu sammeln, müssen hierfür beispielsweise nur jene Trajektorien entfernt werden, die solche unrealistischen Punkte oder Bewegungsmuster beinhalten, um mit hoher Präzision auf den verbleibenden echten Pfad des Nutzers zu schließen.

Ein ausgefeilteres Vorgehen, das diesen Schwachstellen entgegenwirkt, schla-

gen Shankar et al. mit *SybilQuery* [216] vor. Hier werden sowohl die $k-1$ initialen Dummy-Positionen als auch die Zielpunkte der zu erzeugenden Trajektorien so gewählt, dass deren Umgebungen verkehrstechnisch ähnliche Eigenschaften wie die der Endpunkte der tatsächlich vom Nutzer zurückgelegten Strecke aufweisen. Zudem sorgt der Algorithmus dafür, dass die euklidische Distanzen zwischen den echten Start- und Zielpositionen des Nutzers sowie zwischen den jeweils künstlich erzeugten Routenendpunkten – abgesehen von einer maximal tolerierbaren Abweichung – vergleichbar sind, um Dummy-Routen mit einer zur Originalstrecke möglichst ähnlichen Fahrtdauer erzeugen zu können. Im Gegensatz zu [144] werden die Dummy-Positionen daher nicht anhand des unrealistischen *Random Walk*-Bewegungsmodells erzeugt, sondern orientieren sich an der Annahme, dass ein Mensch von einem Startpunkt aus i.d.R. mehr oder weniger direkt sein nächstes Ziel ansteuert. Die $k-1$ aktuellen Location-Dummies werden daher jeweils durch Interpolation entlang der entsprechenden Verbindungsrouten erzeugt. Damit soll erreicht werden, dass Dummy-Trajektorien generiert werden, die der tatsächlichen Route des Nutzers statistisch ähneln und sich zudem nicht auf Basis trivialer Angriffe von realen Mustern menschlicher Mobilität unterscheiden lassen.

Um dies zu erreichen, besteht *SybilQuery* aus drei verschiedenen Modulen und teilt die Karte für die Auswahl von künstlichen Routenendpunkten in ein gleichmäßiges Gitternetz auf. Auf Basis historischer Verkehrsstatistiken werden im Rahmen eines Preprocessing-Schritts für jede Zelle des entstandenen Grids verschiedene mobilitätsrelevante Eigenschaften berechnet, wie die tageszeitabhängige Verkehrsdichte oder die Wahrscheinlichkeit, dass ein Trip der Länge *len* in dieser Zelle beginnt.

Vor Beginn einer Fahrt wird vom Benutzer verlangt, dass dieser wie bei einer Navigationsanwendung den Zielort seines aktuellen Trips angibt. Durch Hinzunahme des aktuellen Aufenthaltsort stehen damit sowohl Start und Ziel zur Verfügung als auch die mittels der euklidischen Distanz zwischen diesen beiden Punkten abgeschätzten Länge *dist* der im weiteren Verlauf zu versteckenden Trajektorie. Diese Werte werden als Eingabeparameter an das erste Modul, den *Endpoint Generator* übergeben. Dieser wählt aus dem gesamten abgedeckten Gebiet $k-1$ künstliche Startzellen *src'* aus, die eine ähnliche Verkehrsdichte aufweisen wie die Zelle *src*, in der sich der tatsächliche Startpunkt befindet und von denen mit ähnlicher Wahrscheinlichkeit Trips der ungefähren Länge *dist* ausgehen. Anschließend wird eine Zielzelle *dst'* im Radius *dist* um *src'* ausgewählt, die eine ähnliche Verkehrsdichte wie die tatsächliche Zielzelle *dst* aufweisen. Im nächsten Schritt müssen schließlich noch realistische, exakte Endpunkte innerhalb der gewählten Dummy-Zellen gefunden werden. Hierfür wird zunächst ein zufälliger Punkt ermittelt, der mit Hilfe eines online Geocoding-Dienstes auf die nächstgelegene Adresse gemappt wird, die fortan als plausibler Endpunkt der Dummy-Route dient.

Das zweite Modul ist der *Path Generator*, der jeweils die kürzeste Route zwischen den $k-1$ soeben ermittelten Paaren künstlicher Endpunkte erzeugt.

Für die Pfaderzeugung zwischen den Dummy-Endpunkten greifen die Autoren auf einen kommerziellen online Routenplaner zurück, der die kürzeste Strecke berechnet. Während des Trips simuliert der *Query Generator* für jede echte LBS-Anfrage je eine Dummy-Position entlang jeder der $k-1$ künstlich erzeugten Trajektorien. Dabei wird in Abhängigkeit von der seit dem Start vergangenen Zeit t und der Gesamtdauer der jeweiligen Dummy-Route der Punkt ermittelt, der durch Projektion der während t mit durchschnittlicher Geschwindigkeit zurückgelegten Strecke auf den entsprechenden Pfad entsteht. Zusammen mit dem echten aktuellen Standort des Benutzers werden diese interpolierten Location-Dummies als Verschleierungsset an den Dienstanbieter übertragen, der nicht unterscheiden kann, auf welcher Route sich der Nutzer tatsächlich bewegt.

Während die Autoren von *SybilQuery* durch die Verwendung historischer Verkehrsstatistiken vor allem Wert auf die Auswahl plausibler Routenendpunkte legen, erscheint die Erzeugung der eigentlichen Dummy-Positionen durch lineare Interpolation entlang der künstlichen Trajektorie relativ naiv. Genau diesen Aspekt untersucht Krumm im Detail für die Erstellung von *Realistic Driving Trips* [149], wobei der Autor ebenfalls davon ausgeht, dass die tatsächlich vom Nutzer zurückgelegten Wege durch die zusätzliche Übermittlung plausibler Dummy-Routen verschleiert werden sollen. In dieser Arbeit werden die GPS-Trajektorien über 250 Studienteilnehmern hinsichtlich unterschiedlicher Eigenschaften analysiert, um anhand der realen Abfolge von gemessenen Standorten eines mobilen Nutzers ein probabilistisches Modell für die Erzeugung möglichst realistischer Fake-Routen zu erstellen sowie wichtige Einflussfaktoren zu identifizieren und geeignete Parameterbelegungen zu finden. Im Gegensatz zu *SybilQuery* werden hierbei nicht nur plausible Start- und Endpunkte gewählt, sondern zudem Dummy-Trajektorien mit zufälligen Abweichungen von der kürzesten Route erzeugt, unterschiedliche Fahrtgeschwindigkeiten simuliert sowie das in den Originaldaten beobachtbare typische Rauschen von GPS-Messungen auf die interpolierten Positionen addiert.

Um einen realistischen, d.h. nicht optimalen Streckenverlauf zu generieren, werden die Kantengewichte des Straßengraphen im Folgenden zufällig variiert. Durch die Verwendung des A*-Algorithmus [112] werden schließlich die kürzesten Pfade zwischen den erzeugten Routenendpunkten ermittelt, die aufgrund der zufällig veränderten Kantengewichte aus Sicht des Angreifers jedoch nicht mehr die optimale und damit eine ggf. als Fälschung identifizierbare Route darstellen. Abhängig von den Eigenschaften der Kanten vor und hinter einem Knoten entlang der gefundenen Route (Strassenbelag, Umgebung, Kreisverkehr, etc.) werden auch die Geschwindigkeiten der statistischen Verteilung aus dem Datensatz entsprechend zufällig gezogen und zwischen den Knoten linear interpoliert. Somit steht nun für jeden Punkt entlang der Dummy-Routen eine realistische Fahrtgeschwindigkeit fest, mit deren Hilfe schließlich mit einer Frequenz von 1 Hz gefälschte Standortangaben und Zeitstempel für die gesamte Strecke berechnet werden. Im letzten Schritt wird auf jeden interpolierten

GPS-Punkt normalverteiltes Rauschen mit einer aus den Originaldaten ermittelten Standardabweichung $\sigma = 7.65\ m$ addiert, um diese tatsächlich wie echte Messdaten wirken zu lassen.

Aus der Kombination von [216] und [149] ergäbe sich somit ein umfassendes System zur Erzeugung einer Menge an realistischen Dummy-Routen, um die tatsächlichen Bewegungen eines mobilen Nutzers darin zu verstecken. Auf einem höheren Abstraktionslevel gehen Bindschaedler et al. [32] darüber hinaus von einem mächtigen Angreifer aus, der auf umfangreiche Modelle der Übergangswahrscheinlichkeiten zwischen verschiedenen Orten entlang routinemäßiger menschlicher Mobilitätsmuster zurückgreifen kann und dabei zudem die Semantik der besuchten Orte wie Arbeitsplatz oder Wohnhaus kennt. Die Autoren zeigen, dass ein solcher Angreifer Trajektorien aus rein zufällig gewählten Stütz- bzw. Endpunkten mit deutlich höherer Wahrscheinlichkeit als Fälschung erkennen kann als solche, die sich ebenfalls an diesem Wissen orientieren.

Aus diesem Grund wird das Prinzip der semantischen Ähnlichkeit zwischen zwei Trajektorien vorgeschlagen. Dabei werden solche Dummy-Trajektorien erzeugt, die zwar keinerlei geographische Ähnlichkeit mit der echten Route des Nutzers aufweisen, dafür aber dieselbe Semantik bzgl. Reihenfolge, Ankunftszeit und Aufenthaltsdauer an Orten verschiedener semantischer Klassen aufweisen. Aufgrund der hohen Komplexität einer derartigen Dummy-Erzeugung und der dafür benötigten Menge an echten Nutzertrajektorien, aus denen es die möglichen Semantiken zu lernen gilt, wird die Erstellung der Fake-Routen in einer Art Offlinephase an eine zentrale Komponente ausgelagert, die somit wieder eine TTP darstellt. Ist der Client hingegen einmal in Besitz einer ausreichenden Zahl zu ihm passender Dummy-Trajektorien, kann die online Erzeugung der Dummy-Positionen entlang dieser semantisch realistischen Fake-Routen während der LBS-Nutzung erneut clientseitig erfolgen.

Ein offensichtlicher Nachteil aller bislang vorgestellten Dummy-Verfahren ist jedoch, dass diese explizit davon ausgehen, dass der Angreifer keinerlei Hintergrundwissen über einen einzelnen Benutzer besitzt. Verfügt er jedoch über grobe Informationen wie z.B. die ungefähre Wohnadresse, ist er unmittelbar dazu in der Lage, exakte Rückschlüsse auf die aktuelle Position des Nutzers sowie die von ihm besuchten Ziele zu ziehen. Befindet sich beispielsweise eine Adresse, in deren Umgebung der Angreifer die Wohnung des Nutzers vermutet, neben $k-1$ entlegenen Startpunkten in der Verschleierungsmenge, können letztere direkt als unplausibel aussortiert werden. Da von jedem Startpunkt wiederum nur eine einzige Route ausgeht, kann der Angreifer somit auch präzise das Ziel des aktuellen Trips in Erfahrung bringen. Analog verhält es sich, wenn der Angreifer ein übliches Ziel des Nutzers grob kennt, wodurch rückwärts auf den echten Startpunkt geschlossen werden kann.

Um genau diesem Problem zu begegnen, gehen Do et al. [63] davon aus, dass der Angreifer ohnehin schon umfangreiches Hintergrundwissen über den Nutzer besitzt, welches er z.B. von Webseiten und aus sozialen Netzen beziehen kann. Aus diesem Grund orientieren sich die Autoren bei der Dummy-

Erzeugung an den möglichen Informationen, die ein Angreifer über den Kontext der Anfrage oder über den Nutzer haben könnte. So werden die möglichen Dummy-Standorte zunächst so sortiert, dass sie einem Angreifer im Kontext der aktuellen Anfrage möglichst plausibel erscheinen sollen. Hierfür beziehen die Autoren allgemeingültiges Hintergrundwissen mit ein, z.B. dass ein Restaurant zur Mittagszeit einen wahrscheinlicheren Aufenthaltsort darstellt als andere Lokalitäten. Zusätzlich werden noch jene Orte stärker gewichtet, die der Nutzer zu der jeweiligen Tageszeit tatsächlich schon besucht hat und die dem Angreifer daher besonders wahrscheinlich erscheinen sollen.

Die Autoren beschränken ihre Arbeit jedoch ausdrücklich auf sporadische LBS-Anfragen und gehen daher davon aus, dass ein Nutzer ortsbezogene Dienste so selten benutzt, dass die aufeinanderfolgenden Standorte nicht miteinander korrelieren und keinen in sich stimmigen Trip ergeben müssen. Darüber hinaus werden durch dieses Verfahren natürlich auch die wichtigen Orte des Nutzers unmittelbar verraten und stattdessen nur der aktuelle Aufenthaltsort unter diesen Orten verschleiert. Ein Angreifer, der bislang wenig über sein Ziel weiß, wird somit automatisch mit den entsprechenden Profil-Informationen über den Nutzer wie Wohnung, Arbeitsplatz, etc. ausgestattet. Insgesamt verfolgt dieser Ansatz somit lediglich den Schutz des aktuellen Standorts, wofür andere Ziele wie die Verhinderung einer Profilerstellung geopfert werden.

Ein weiteres Problem, das dieser Ansatz aufweist, ist der Schutz neuer und für einen Nutzer bisher „unwahrscheinlicher" Aufenthaltsorte. Da neben dem tatsächlichen Standort stets solche Dummy-Positionen ausgewählt werden, die zur aktuellen Tageszeit wahrscheinlich sind oder vom Nutzer regelmäßig besucht werden, gibt es nur eine mögliche Erklärung, warum es ein ansonsten unwahrscheinlicher Aufenthaltsort in das Verschleierungsset geschafft hat: Es muss der Ort sein, an dem sich der Nutzer gerade tatsächlich aufhält. Bei der Evaluation in [63] wird dieser Aspekt nicht berücksichtigt, da der simulierte Angreifer dort die Strategie verfolgt, den auf Basis seines Hintergrundwissens über den Nutzer am wahrscheinlichsten erscheinenden Ort als echten Aufenthaltsort anzunehmen. Bei den durchgeführten Experimenten mit unterschiedlichen Werten von k zwischen 2 und 5 gelingt es dem simulierten Angreifer jedoch auch so bereits jeweils in mindestens 55 % der Fälle, den echten Aufenthaltsort des Nutzers korrekt aus der Verschleierungsmenge zu identifizieren.

Eine alternative Herangehensweise bei der Auswahl von Dummy-Positionen für ebenfalls sporadische, unkorrelierte LBS-Anfragen schlagen Niu et al. in [189] vor. Hier werden die Dummy-Standorte so ausgewählt, dass sie eine möglichst ähnliche Aufenthaltswahrscheinlichkeit wie der tatsächliche Standort des Nutzers aufweisen, wodurch das soeben beschriebene Problem der verräterischen Zusammensetzung der Verschleierungsmenge umgangen wird. Auf der anderen Seite berücksichtigt dieser Ansatz wiederum kein kontextuelles Hintergrundwissen wie die Uhrzeit der Anfrage, sodass dieser Ansatz in der Evaluation in [63] beim Schutz regelmäßig besuchter Orte sogar noch etwas schlechter abgeschneidet. Eine Kombination dieser beiden Verfahren erscheint

daher für solche Szenarien vielversprechend, in denen zwar nicht die Profilerstellung verhindert, aber der aktuelle Standort verschleiert werden soll.

Insgesamt geht der Einsatz solcher dummybasierten Verschleierungstechniken mit mehreren Vorteilen einher. Der größte Pluspunkt ist in der Vermeidung einer TTP zu sehen, die bei den im vorangehenden Kapitel vorgestellten Verfahren stets einen Flaschenhals hinsichtlich der Privatsphäre und Performanz des Systems darstellt. Stattdessen wird der Schutz der Privatsphäre rein clientseitig erreicht, indem neben dem eigentlichen Standort des Nutzers unter Berücksichtigung unterschiedlichen Hintergrundwissens zufällige Positionen als mögliche Query-Ausgangspunkte an den Dienstanbieter übermittelt werden. Demzufolge ist die privatsphärekonforme Dienstnutzung hierbei auch nicht davon abhängig, dass sich zum Zeitpunkt der Anfrage eine ausreichend hohe Anzahl an weiteren Benutzern in der Umgebung des anfragenden Nutzers befindet. Somit ist in jeder Situation ohne Wartezeit die unmittelbare Anfrageformulierung möglich, wodurch auch interaktive LBS problemlos und mit den gewohnten Reaktionszeiten genutzt werden können. Nachdem zudem der tatsächliche Standort des Nutzers jeweils im erzeugten Anonymitätsset enthalten ist, kann wie zuvor garantiert werden, dass auch das optimale Ergebnis stets in den Ergebniskandidaten des LBS vorkommt. Im Gegensatz zu den zuvor beschriebenen Verfahren auf Basis der k-Anonymität lässt sich dies zudem ohne die explizite Kooperation des LBS-Anbieters erreichen, da hierbei ausschließlich mehrere standardmäßige, punktbasierte Anfragen versendet werden. Es wird also nicht wie zuvor vorausgesetzt, dass der Anbieter spezielle Dienstschnittstellen und Algorithmen zur korrekten Beantwortung regionsbasierter LBS-Anfragen zur Verfügung stellt. Zuguterletzt ermöglicht die Verwendung der dummybasierten Standort- und Trajektorienverschleierung in gewissem Umfang auch eine pseudonyme bzw. personalisierte Dienstnutzung.

Gleichzeitig weisen diese Verfahren auch einige Nachteile auf. Eine erhebliche Schwachstelle der dummybasierten Standortverschleierung ist, dass die tatsächlichen Anfragen – und damit die exakten Aufenthaltsorte und Bewegungsmuster eines Nutzers – stets im Original in der Verschleierungsmenge enthalten sind. Die große Schwierigkeit liegt daher, wie anhand der Komplexität der beschriebenen Verfahren zu sehen ist, jeweils in der Erzeugung plausibler Dummy-Standorte und in sich stimmiger Trajektorien. Dabei müssen nicht nur die gewählten Start- und Zielpunkte einer Trajektorie plausibel sein, sondern auch der Verlauf der zwischen den Endpunkten zurückgelegten Strecke. Darüber hinaus wird bei der Verwendung von Location-Dummies durch den bewussten Verzicht auf eine TTP keine Kommunikationsanonymität erreicht – so ist für den LBS-Anbieter stets ersichtlich, von welchem Teilnehmer eine Anfrage stammt. Besitzt der Angreifer grobes Wissen über die typischen Orte oder Regionen eines Nutzers, gestaltet es sich für ihn einfach, falsche Standortangaben mit hoher Wahrscheinlichkeit von echten zu unterscheiden.

Darüber hinaus muss der Einsatz von Location-Dummies als kontinuierlicher Prozess angesehen werden, der früher verwendete Dummies nicht ver-

gessen darf, sondern konsistent weiterführen muss. Werden Dummy-Orte bei jeder LBS-Anfrage – egal ob diskret oder kontinuierlich – zufällig neu erzeugt, besteht die Gefahr einer simplen Frequenzanalyse durch den Angreifer, der über die Zeit hinweg die echten Standorte, die wiederholt in den Anfragen auftauchen, von nur einmalig beobachteten Dummies unterscheiden kann. Aufgrund der in vielen Studien gezeigten Regelmäßigkeit und Vorhersagbarkeit menschlicher Bewegungsmuster stellt dies ein ernstzunehmendes Problem dar. In *SybilQuery* wird zu diesem Zweck ein Caching der häufigsten LBS-Anfragen des Nutzers durchgeführt und frühere Dummies zur Verschleierung desselben Standorts wiederverwendet.

Peddinti et al. [192] weisen jedoch auf ein weiteres Problem hin, das trotz dieses Cachings bei der unabhängigen Verschleierung einzelner Trips entsteht. In ihrer Analyse des *SybilQuery*-Verfahrens gelingt es den Autoren – abhängig von den Fähigkeiten des jeweils simulierten Angreifers – einen Großteil der echten Trajektorien zu erkennen. Ein schwacher Angreifer ohne jegliches Hintergrundwissen ist für den zur Analyse verwendeten Datensatz von Taxifahrten und unter Einsatz überaus simpler Angriffstechniken immerhin dazu imstande, knapp 43 % der echten Routen korrekt zu klassifizieren, obwohl der gewählte Wert von $k = 5$ dies eigentlich nur zu 20 % ermöglichen sollte. Dies liegt daran, dass die für aufeinanderfolgende Nutzerbewegungen künstlich erzeugten Startpositionen oft weit von den zuvor verwendeten Zielen entfernt liegen und aus globaler Sicht somit keine plausiblen Trajektorien ergeben. Eine naheliegende Gegenmaßnahme ist es, die zuvor verwendeten Routenendpunkte auch wieder bei der Erzeugung der neuen Startpositionen zu berücksichtigen. Ein Angreifer mit Hintergrundwissen über frühere Trips eines Nutzers ist sogar dazu in der Lage, 93.67 % aller echten Trips korrekt zu erkennen. Zur Lösung dieses Problems schlagen Peddinti et al. ähnlich wie [63] vor, die üblich vom Nutzer besuchten Orte als Endpunkte für Dummy-Trajektorien zu verwenden, räumen gleichzeitig jedoch ein, dass dem Angreifer dadurch noch fehlende Informationen über das Bewegungsprofil des Nutzers direkt präsentiert werden.

Ein weiterer Nachteil der dummybasierten Standortverschleierung ist, dass sich der Grad an Privatsphäre bei all diesen Verfahren ausschließlich linear mit dem dadurch verbundenen Kommunikationsaufwand erhöhen lässt. Ein hohes Maß an Anonymität, d.h. viele mögliche Aufenthaltsorte, bedeutet unweigerlich die Formulierung vieler Anfragen an den LBS. In demselben Zusammenhang können diese Verfahren auch keine Unterscheidung zwischen schützenswerten und nicht-privatsphärerelevanten Standortangaben treffen. So kann zum Schutz vor der Profilerstellung über einen Benutzer auf Basis ortsbasierter Inferenzangriffe z.B. davon ausgegangen werden, dass das Wissen eines Angreifers über einen Aufenthalt an einem bestimmten Ort die Privatsphäre des Nutzers gefährdet, wohingegen der Weg zwischen zwei Orten deutlich weniger privatsphärekritisch erscheint.

Durch das massive Einschleusen simulierter Daten und künstlicher Trajektorien lassen sich solche Verfahren zudem nicht für den Bereich der partizipa-

tiven Sensornetze einsetzen [150]. Ein Beispiel hierfür stellt die Ermittlung der aktuellen Verkehrslage dar, wie sie Anbieter von verkehrsadaptiven Routing- und Navigationsanwendungen heute durch die kontinuierlichen Standortupdates ihrer mobilen Nutzern erheben. Werden an solche Systeme für jede echte Route $k-1$ anhand historischer Verkehrsstatistiken probabilistisch erzeugte Dummy-Trajektorien übermittelt, wie [149] oder [216] es vorschlagen, ergeben sich daraus zwei verschiedene Probleme. Entweder, der LBS-Anbieter ist nicht dazu in der Lage, Fake-Routen von echten zu unterscheiden – dann beeinträchtigen diese die Dienstqualität, da die Verlässlichkeit der Verkehrsdaten durch die Vielzahl an simulierten Routen eingeschränkt wird [50].

Ein LBS-Anbieter oder ein Angreifer versucht daher, Dummy-Routen als solche zu erkennen, was in der Praxis extrem zuverlässig und quasi in Echtzeit funktioniert [115]. Derartige *Sanitation*-Verfahren basieren z.B. darauf, Fälschungen über Outlier-Detection oder Konsistenzprüfungen zu erkennen [119]. Angenommen, eine Trajektorie bewegt sich mit der dort üblicherweise typischen Geschwindigkeit eine Kante entlang, während der Dienstanbieter aufgrund anderer Fahrer bereits die derzeit tatsächlich erreichbare Durchschnittsgeschwindigkeit kennt und über Störungen informiert ist. In diesem Fall wird die Dummy-Route unmittelbar als Fälschung erkannt und trägt nichts zur Privatsphäre des Nutzers bei.

2.2.3.3 Verwendung von Ankerpunkten

Neben den dummybasierten Verschleierungstechniken gibt es eine Reihe weiterer clientseitig umsetzbarer Mechanismen zum Schutz von Standortinformationen im Rahmen der LBS-Nutzung, die den tatsächlichen Aufenthaltsort des Nutzers nicht exakt – oder situationsabhängig gar nicht – preisgeben. Meist muss hierfür jedoch ebenfalls ein höherer Kommunikationsaufwand in Kauf genommen werden oder Einbußen in der Dienstqualität hingenommen werden. Auch diese Verfahren gehen davon aus, dass nicht die Identität des Nutzers das schützenswerte Attribut darstellt, sondern die von ihm besuchten Standorte.

Eine Ausprägung derartiger Schutztechniken stellt der Landmarken-basierte *Space Twist*-Ansatz von et Yiu al. dar [262]. Dieser ermöglicht die privatsphäreschonende Ermittlung der k nächsten POIs in der Umgebung des Nutzers, welcher durch geeignete Parametrisierung des Algorithmus zudem über den erreichbaren Grad an Privatsphäre und den dafür nötigen Kommunikationsaufwand abwägen kann. Als Maß für die Privatsphäre definieren die Autoren den durchschnittlichen Abstand aller Punkte z in der resultierenden Verschleierungsregion Ψ zum tatsächlichen Aufenthaltsort des Nutzers. Je größer diese Fläche ausfällt, desto höher der erreichte Grad an Privatsphäre.

Im Gegensatz zu den dummybasierten Privatsphäre-Mechanismen wird hierbei zu keiner Zeit der reale Ort der LBS-Anfrage q übertragen, sondern stattdessen eine gefälschte Ortsangabe q'. Dieser sogenannte Ankerpunkt wird einmalig an den Dienstanbieter übermittelt. Auf Basis einer anbieterseitig implementierten, inkrementellen *k-Nearest-Neighbor*-Suche (kNN) werden in der

Folge so lange POIs mit zunehmender Distanz zu diesem Ankerpunkt an das Endgerät des Nutzers übermittelt, bis die für die tatsächliche Nutzerposition q gültige Antwort garantiert in der Antwortmenge enthalten ist.

Bei diesem Vorgehen entsteht ein kreisförmiger *Supply Space* um q', der mit jeder weiteren Antwort des Servers wächst. Diese Fläche beschreibt die Region, in der bereits alle POIs bekannt sind. Um q herum schrumpft parallel dazu der ebenfalls kreisförmige *Demand Space*, der die k derzeit bekannten nächsten Nachbarn zum tatsächlichen Ort der LBS-Anfrage enthält. Besitzt ein neu übermittelter POI eine geringere Distanz zu q als einer der bisherigen nächsten Nachbarn, wird die Liste an kNNs entsprechend aktualisiert. Die korrekte Antwort auf die Anfrage des Nutzers ist gefunden, sobald der *Demand Space* vollständig vom *Supply Space* überdeckt wird. Tritt dieser Zustand ein, sendet der Client eine `stop`-Nachricht an den LBS, um die unnötige Übermittlung weiterer POI-Antworten zu beenden.

Die euklidische Distanz zwischen q und q' kann somit als ungefähres Maß für den erreichbaren Grad an Privatsphäre betrachtet werden. Je näher diese Punkte beisammen liegen, desto kleiner ist die Fläche, in welcher der tatsächliche Nutzerstandort liegen muss. Befinden sich die Punkte hingegen in großer Entfernung zueinander, werden viele POI-Antworten benötigt, um die Anfrage des Nutzers korrekt beantworten zu können. Umso größer fällt jedoch auch der *Supply Space* der Anfrage aus. Aufgrund der deterministischen Reihenfolge der übermittelten POIs lässt sich unter Kenntnis von q', k und der Anzahl an benötigten POI-Antworten m die Region Ψ, in der sich der Nutzer befinden muss, innerhalb dieser Fläche jedoch noch weiter eingrenzen. Um Garantien bezüglich des erreichten Grades an Privatsphäre geben zu können, schlagen die Autoren daher vor, das Senden der `stop`-Nachricht so lange hinauszuzögern, bis die Größe der Fläche Ψ den Privatsphärebedürfnissen des Nutzers entspricht. Diese Zusicherung hinsichtlich der Größe der Fläche Ψ geht auf Kosten zusätzlichen Kommunikationsaufwands in Form von POI-Anfragen, jedoch ist es dem LBS-Anbieter somit unmöglich zu entscheiden, welche der übermittelten POIs tatsächlich die nächsten Nachbarn des Nutzers darstellen.

Space Twist lässt sich somit ohne die Existenz einer TTP umsetzen und garantiert das korrekte Ergebnis. Dabei verursacht es jedoch merklichen Kommunikationsoverhead. So müssen hierfür abhängig von den Privatsphärebedürfnissen des Nutzers und der POI-Dichte bis zu mehrere tausend Einträge für eine Anfrage übertragen werden [262]. Außerdem wird die Kooperation des Dienstanbieters vorausgesetzt, welcher die inkrementelle Suche nächster Nachbarn ermöglichen muss. Der Ansatz eignet sich nur für die Verwendung im Rahmen der POI-Suche und berücksichtigt kein mögliches Hintergrund- oder Kartenwissen eines Angreifers über die Verteilung möglicher Orte in Ψ. Stattdessen gehen die Autoren explizit davon aus, dass jeder Ort $q_c \in \Psi$ dieselbe Wahrscheinlichkeit besitzt, der tatsächliche Standort des Nutzers zu sein.

2.2.3.4 Weitere Ansätze zum Schutz von Standortinformationen

Darüber hinaus gibt es noch eine Reihe weiterer Techniken, welche die Privatsphäre von Standortdaten bei der Nutzung ortsbezogener Dienste auf Basis anderer Schutzmaßnahmen gewährleisten. Diese sollen an dieser Stelle ebenfalls erwähnt werden, um die Fülle der unterschiedlichen Ansätze zu zeigen, lassen sich jedoch nicht in die vorangehenden Kategorien einordnen.

2.2.3.4.1 Caching Eine weitere Möglichkeit stellen Caching-basierte Techniken dar, wie sie Amini et al. mit *Caché* vorstellen [4]. Die Autoren beschreiben ein System, bei dem der aktuelle Aufenthaltsort des Nutzers dadurch geschützt wird, dass LBS-Informationen nicht zum Zeitpunkt der Dienstnutzung, sondern intervallbasiert im Voraus heruntergeladen werden.

Entlang des Spektrums von hochaktuellen Echtzeit-LBS wie der verkehrsadaptiven Routenplanung bis zu relativ statischen Diensten wie einer Fahrplan- oder POI-Suche positioniert sich *Caché* in etwa in der Mitte und eignet sich für Dienste, deren Daten sich über die Zeit hinweg zwar ändern, die Änderungsrate jedoch einigermaßen niedrig ist. Der Nutzer gibt dabei an, für welche Regionen einer in Gridzellen aufgeteilten Karte er in Zukunft Informationen abrufen möchte. Der LBS-Anbieter erfährt jedoch nicht, zu welchem Zeitpunkt sich der Benutzer dort aufhalten wird oder welche genauen Orte er in dieser Region besucht. Der Client lädt daraufhin in einem ebenfalls durch den Nutzer spezifizierten Update-Intervall die POI-Informationen für alle ausgewählten Regionen herunter und hält diese lokal vor. Formuliert der Nutzer online eine ortsbezogene Anfrage, gibt *Caché* die gespeicherten Daten zurück, anstatt den LBS mit der aktuellen Position des Nutzers zu versorgen.

Um die praktische Einsetzbarkeit ihres Systems zu bewerten, untersuchen die Autoren verschiedene Typen ortsbezogener Dienste und vergleichen die prozentuale Übereinstimmung der aus dem Cache geladenen Antworten mit denen, die er Dienst in Echtzeit zurückgeben würde. Solche LBS, deren Inhalte sich kaum ändern, eignen sich gut für einen derartige lokale Speicherung. Darüber hinaus bietet dieses System den Vorteil, dass es rein clientseitig umsetzbar ist und keine Kooperation des Dienstanbieters benötigt. Systeme, die auf Echtzeit-Informationen beruhen oder wie Buddy-Tracker den Standort der Nutzer selbst benötigen, lassen sich damit jedoch nicht nutzen. Dieses System schützt somit insbesondere den Zeitpunkt, an dem sich ein Nutzer in einer bestimmten Region aufhält, nicht jedoch die Region selbst.

2.2.3.4.2 Private Information Retrieval Unter dem Titel *Anonymizers are not Necessary* beschreiben Ghinita et al. in [90] ein Client-Server-basiertes Verfahren zur privatsphäreschonenden Nutzung ortsbezogener Dienste. Auf Basis des *Private Information Retrieval*-Prinzips (PIR) legen die Autoren besonderen Wert auf die Vermeidung einer zentralen, allwissenden TTP zur Herstellung von Kommunikationsanonymität und Privatsphäre.

Der Ansatz basiert wie einige kryptographische Verfahren auf der *Quadratic Residuosity Assumption* (QRA), die besagt, dass es rechentechnisch überaus aufwändig ist, herauszufinden, ob eine Zahl a ein quadratischer Rest modulo $N = q_1 \times q_2$ ist, wobei q_1 und q_2 große Primzahlen sind [154]. Anders als bei früheren PIR-Implementierungen muss dabei nicht der gesamte Datenbestand zwischen Server und Client kommuniziert werden. Im Gegensatz zu den k-Anonymitäts-basierten Ansätzen werden keinerlei Informationen über den Standort des Nutzers offenbart – auch nicht in Form einer verzerrten oder vergröberten Positionsangabe. Stattdessen wird der Standort gemäß der QRA-Annahme verschlüsselt an den LBS-Anbieter übertragen, der darauf mit einer entsprechend kodierten Antwort reagiert, die ähnlich eines Public-Key-Verfahrens nur vom Client korrekt entschlüsselt werden kann. Zudem gehen die Autoren von einer anonymen Dienstnutzung aus. Dem Anbieter ist es – unter Ergreifung entsprechender Maßnahmen durch den Client wie die Nutzung eines Kommunikationsproxys – somit nicht möglich, auf den Urheber einer Nachricht zu schließen. Dementsprechend ist dieses Verfahren auch per se immun gegen Angriffe auf Basis der räumlichen Korrelation aufeinanderfolgender Queries und muss daher keine besonderen Maßnahmen zum Schutz von kontinuierlichen LBS-Anfragen treffen.

Ein grundsätzliches Problem aller PIR-basierten Verfahren ist der immense Kommunikations- und Rechenaufwand auf Seiten des Clients und des Servers. So werden im Rahmen dieses Verfahrens als Antwort auf eine POI-Anfrage an eine Datenbank mit n Einträgen, $\sqrt{n}$ POIs vom Server an den Client zurückgeliefert. So müssen abhängig von der Gesamtzahl an POIs sehr viele Daten übertragen werden. Die POI-Suche kann zudem nur anhand des Standorts erfolgen und kann nicht z.B. über die Art des POIs – also ob Restaurant oder Apotheke – genauer spezifiziert werden. Darüber hinaus wird stets die umfangreiche Kooperation des Dienstanbieters vorausgesetzt, der entsprechende Module und Rechenkapazitäten zur Verarbeitung der PIR-basierten Anfragen bereitstellen muss. Andere Typen ortsbezogener Dienste als die POI-Suche werden nicht unterstützt.

2.2.3.4.3 Privatsphäreschonende Nähe-Erkennung Eine spezielle Klasse ortsbezogener Dienste stellen Buddy-Tracker sowie die sog. *Location-based Social Networks* (LBSN) [266] dar. Dabei wird ein Nutzer eines entsprechenden Dienstangebots vom LBS-Server über Freunde in seiner Nähe benachrichtigt, um spontane Treffen zu ermöglichen oder es werden Nutzern Personen in der Nähe mit ähnlichen Interessen vorgeschlagen. Wie u.a. Bilogrevic et al. [31] zeigen, sind LBS-Anbieter allgemein dazu in der Lage, soziale Beziehungen zwischen Teilnehmern ihres Dienstes auch unmittelbar aus den Trajektorien und Aufeinandertreffen der Nutzer zu inferieren.

Bei den bisher vorgestellten Einsatzszenarien waren alle für die ortsbezogene Diensterbringung relevanten Informationen öffentlich und befinden sich wie z.B. die Koordinaten von POIs im Besitz des LBS. Im Gegensatz dazu

muss nun stets die Privatsphäre aller Teilnehmer geschützt werden – sowohl vor dem Dienstanbieter selbst als auch vor anderen Teilnehmern. Vor diesem Hintergrund gehen viele Verfahren insbesondere davon aus, dass ein Nähe-Test im Fall eines negativen Ausgangs beiden Parteien so wenig Informationen wie möglich über den Standort des Gegenübers preisgeben soll [198].

In der einfachsten Implementierung empfängt ein Server in regelmäßigen Abständen die aktuellen Standortinformationen aller Nutzer. Um eine personalisierte Diensterbringung zu ermöglichen, kennt der Server dabei die Freundeslisten aller Nutzer und prüft kontinuierlich, ob sich zwei befreundete Teilnehmer innerhalb eines spezifizierten Radius zueinander befinden. Der Dienstanbieter stellt in diesem Fall somit eine TTP dar, die stets über die Aufenthaltsorte aller Nutzer informiert ist. Eine Möglichkeit dies zu verhindern stellt Werner mit der Einführung sog. *Locagrams* vor [249], die auf asymmetrischer Verschlüsselung aller Standortinformationen mit dem öffentlichen Schlüssel des intendierten Kommunikationspartners beruhen. Der LBS-Anbieter tritt hierbei nur noch als Kommunikationsplattform auf, die Nachrichten zwischen befreundeten Nutzern weiterleitet. So kann verhindert werden, dass Unbefugte die Standorte eines Nutzers in Erfahrung bringen, der Kommunikationspartner lernt jedoch stets den genauen Aufenthaltsort.

Ruppel et al. vermeiden dies durch den Einsatz eines zentralen Servers, der die paarweisen Distanzen zwischen den Nutzern berechnet [205]. Um jedoch dem LBS-Anbieter nicht die genauen Standorte aller Nutzer wissen zu lassen, wenden alle Teilnehmer dieselbe, distanzerhaltende Koordinatentransformation in Form einer zweidimensionalen Rotation und Translation auf den eigenen Standort an, bevor diese Information an den LBS gesendet wird. Die Transformationsparameter werden out-of-band unter den Dienstteilnehmern verteilt. Da es sich hierbei jedoch nicht um ein kryptographisches Verfahren handelt, wird die Effektivität der Koordinatentransformation zum Schutz der Privatsphäre bezweifelt [222] und argumentiert, dass sich das verwendete Mapping zurückrechnen lässt.

Šikšnys et al. schlagen in [221, 222] zwei ähnliche Systeme mit derselben Architektur vor. Anstelle von Distanzen berechnet der Server hierbei jedoch, ob sich ein Nutzer in einem von einem anderen Teilnehmer als Polygon definierten *Vicinity*-Gebiet aufhält. Dabei wird die Karte in ein beliebig feingranulares Grid aufgeteilt und alle Gitterzellen mit einem Index versehen. Alle Dienstnutzer teilen sich hierbei denselben symmetrischen Schlüssel, mit dem der Index der aktuellen Aufenthaltszelle sowie die von den Vicinity-Regionen überdeckten Zellen verschlüsselt. Der Server empfängt diese Daten und kann einfach testen, ob sich zwei verschlüssselte Indices gleichen ohne dabei etwas über den Ort selbst herauszufinden. Bei all diesen Ansätzen besteht jedoch die Gefahr, dass der Dienstanbieter in Besitz der Transformationsparameter oder des globalen Schlüssels gelangt, z.B. weil er sich selbst als Nutzer ausgibt und somit direkt mit dem Schlüsselmaterial versorgt wird. In diesem Fall lassen sich alle Schutzmaßnahmen trivial rückgängig machen, ohne dass die Nutzer

des Dienstes dies mitbekommen.

Zhong et al. schlagen drei Protokolle, *Louis, Lester and Pierre* [267] zur privatsphäreschonenden Umsetzung der Nähe-Erkennung auf Basis homomorpher Verschlüsselung vor. Hierbei hat wie in [249] jeder Nutzer sein persönliches Schlüsselpaar $\mathcal{K}_U$, jedoch wird auf die homomorphen Kryptosysteme Pailler und CGS97 zurückgegriffen. Diese erlauben die korrekte Berechnung des verschlüsselten Additionsergebnisses auf den verschlüsselten Daten. Im ersten Schritt verschlüsselt Alice ihren Standort mit ihrem eigenen öffentlichen Schlüssel $\mathcal{K}_{A^+}$. Bob erhält diese Daten und kann durch Verschlüsselung seines eigenen Standorts mit $\mathcal{K}_{A^+}$ die verschlüsselte Distanz berechnen. Er schickt dieses Ergebnis z.B. zurück an Alice. Die drei Protokolle unterscheiden sich hinsichtlich des Einsatzes eines vertrauenswürdigen Dritten sowie des Nachrichtenaufkommens und den jeweils garantierten Privatsphäreeigenschaften.

Selbst bei den Verfahren, die wie [267] nur die zwei beteiligten Parteien binär über den Ausgang eines Nähe-Tests informieren, besteht jedoch das Problem, dass ein neugieriger Nutzer seinen eigenen Standort fälschen kann, um herauszufinden, ob sich eine Person an einem bestimmten Ort aufhält oder nicht. Aus diesem Grund führen Narayanan et al. [187] die Verwendung fälschungssicherer *Location Tags* in die Nähe-Erkennung ein, die sich unvorhersagbar aus orts- und zeitabhängigen Umgebungseigenschaften zusammensetzen, wie z.B. der derzeit beobachtbare WLAN- und Bluetoothdaten oder Audiofingerprints. Der grundlegende Gedanke ist, dass sich diese Tags nicht erzeugen lassen, wenn man sich nicht tatsächlich an diesem Ort aufhält, diese gleichzeitig aber keinen Hinweis auf den Ort selbst geben. Die Erkennung räumlicher Nähe zwischen zwei Benutzern mit einem solchen Ortsbeweis wird damit auf das Problem des *Private Equality Testing* [165] sowie der *Private Set Intersection* [80] reduziert, die bei positivem Ausgang jeweils die Nähe zum Kommunikationspartner beweisen, andernfalls aber nichts über dessen aktuelle Position verraten.

Vergleichbar zu diesem Ansatz ist das *ProbeTag*-Verfahren von Maier, Schauer und Dorfmeister, das unter Verwendung der MAC-Adressen von mobilen WLAN-Endgeräten in der Umgebung zuverlässig zeitabhängige Location Tags erzeugt und feingranulare Nähe-Tests ermöglicht [173].

2.2.3.4.4 Angreifbarkeit anonymisierter Trajektorien-Daten Während sich die bisher vorgestellten Ansätze primär jeweils auf den Echtzeit-Schutz von Standortinformationen fokussieren, gibt es auch einige Arbeiten, die sich mit der individuellen Privatsphäre und Anonymität ganzer Trajektorien-Datensätze beschäftigen. Diese Arbeiten gehen Fragen nach, welche Informationen sich aus Sicht eines LBS-Anbieters oder eines Angreifers aus den historischen Ortsinformationen aller Nutzer extrahieren lassen, bzw. welche Schutzmaßnahmen vor der Veröffentlichung solcher Daten getroffen werden sollten.

Über die privatsphärebezogene Wirkungslosigkeit der alleinigen Verwendung von Pseudonymen, bei der ein Nutzer lediglich über einen zufälligen Identifier bekannt ist, berichten Zang et al. [263]. Aus den *Call Details Records* (CDR)

von 25 Mio. Handynutzern extrahieren die Autoren die Top N Orte aller Nutzer und zeigen, dass diese selbst bei einer Vergröberung der Standortangabe auf das umgebende Postleitzahlgebiet noch für 35 % der Nutzer eine eindeutige Identifizierung ermöglichen. In Kombination mit Hintergrundwissen des Angreifers z.B. über Wohn- und Arbeitsort eines Nutzers die gesamte Trajektorie offengelegt. Selbst bei derart sporadisch erhobenen Standortinformationen und unter grober Standortverschleierung kann mehr als $\frac{1}{3}$ aller Teilnehmer eindeutig identifiziert werden, womit das Ziel der Anonymisierung beim Einsatz von Pseudonymen klar verpasst wird.

Gruteser und Hoh [105] zeigen zudem, dass auch die vollkommen anonyme Preisgabe von Standortdaten dieselben Gefahren birgt. Selbst ohne Existenz eines wiedererkennbaren Pseudonyms pro Nutzer lassen sich einzelne Standortupdates zu ganzen Trajektorien verknüpfen. Das Problem entsteht hierbei aus einem oft vorliegenden Missverhältnis aus Nutzerdichte und Update-Intervall. So lassen sich z.B. aufgrund der Vorhersagbarkeit, Zielgerichtetheit und Regelmäßigkeit menschlicher Mobilität einzelne Datenpunkte mit Hilfe des *Multi Target Tracking* in vielen Fällen wieder zu zusammenhängenden Trajektorien kombinieren und zur De-Anonymisierung verwenden. Genau wie die auf k-Anonymität basierten Schutztechniken funktioniert diese Technik also nur, wenn zu jeder Zeit eine ausreichend hohe Nutzerdichte besteht.

Hoh et al. [120] entwickeln daher ein Verfahren, das vor der Veröffentlichung eines solchen Datensatzes eine rigorose Filterung der Datenpunkte vornimmt. Die Autoren führen das Maß der *Mean Time to Confusion* (MTTC) ein, das angibt, über welchen maximalen Zeitraum sich einzelne Positionsupdates eindeutig zu einer Trajektorie zusammenfügen lassen dürfen. Zudem wird ein entropiebasiertes Maß für die Konfidenz eines Angreifers über die korrekte Verbindung von Datenpunkten verwendet. Nur wenn sich nach Ablauf der maximal erlaubten MTTC zum selben Zeitpunkt eine ausreichend hohe Zahl an Datenpunkten von anderen Teilnehmern in der Nachbarschaft eines Punktes befinden, die aufgrund ihres Abstands und relativen Lage zur selben Trajektorie gehören könnten, ist die Verwirrung des Angreifers möglich. Andernfalls wird die Preisgabe des nächsten Datenpunkts verweigert. Bei der privatsphärebezogenen Aufbereitung des Datensatzes können für beide Kriterien Schwellwerte angegeben werden. Nur solche Datenpunkte verbleiben in dem zu veröffentlichenden Datensatz, die beide Bedingungen erfüllen. Zwar sinkt somit die Anzahl an verfügbaren Datenpunkten, die Verknüpfbarkeit einzelner Punkte zu einer Trajektorie, die später eine potentielle De-Anonymisierung ermöglicht, wird dafür effektiv unterbunden.

In eine ähnliche Richtung geht der Ansatz von Gao et al. [85], die das Konzept der *Mix-Zones* (vgl. Kapitel 4.3.1) aufnehmen und Trajektorien künstlich splitten, indem Regionen festgelegt werden, in denen alle Datenpunkte aller Teilnehmer konsequent ausgeblendet werden. Aus Sicht des Angreifers bewegen sich viele Trajektorien in eine solche Zone und verlassen diese auch wieder. Da jedoch nicht bekannt ist, welche Teiltrajektorien zusammengehören, wird

auf diese Weise für Unsicherheit auf Seite des Angreifers gesorgt.

Das problematische Szenario, in dem ein Angreifer eine gewisse Menge an Hintergrundwissen über einzelne Nutzer besitzt, untersuchen z.B. Gambs et al. [82]. Es wird angenommen, dass der Angreifer einzelne Teilstücke von Trajektorien kennt und anhand dieser Informationen versucht, das Pseudonym des Nutzers herauszufinden, um somit alle seine Bewegungen nachvollziehen zu können. Der Angreifer extrahiert hierbei die besuchten Orte einer Trajektorie und erzeugt daraus eine sog. *Mobility Markov Chain* (MMC), welche die Übergangswahrscheinlichkeiten des Nutzers zwischen den einzelnen Orten modelliert. Für den Angriff wird schließlich die Ähnlichkeit dieses Modells zu den MMCs aller im Datensatz enthaltenen Nutzer ermittelt, um den Erzeuger der Modells zu identifizieren. Bei den durchgeführten Experimenten bestätigt sich einmal mehr die Hypothese, dass die Mobilität eines Menschen als eine Art Signatur betrachtet werden kann, die ihn eindeutig identifiziert. Die korrekte Identifizierung gelingt in bis zu 45 % der Fälle.

Um diesem Problem entgegenzuwirken, filtern Terrovitis et al. [232] verräterische Teiltrajektorien vor der Veröffentlichung des Datenbestands, sodass es keine eindeutige Abbildung von Teilsegmenten auf einen Eintrag gibt. Eine Trajektorie besteht in diesem Szenario aus der geordneten Liste an Orten, an denen sich ein Nutzer nacheinander aufgehalten hat. Die Autoren gehen davon aus, dass das Vorwissen jedes möglichen Angreifers in Form einer lückenhaften Projektion einer Trajektorie genau bekannt ist – beispielsweise, weil er einen Teil der Daten über Nutzer selbst beigesteuert hat. Je mehr der gespeicherten Trajektorien auf das bekannte Muster des Angreifers passen, desto höher der Grad an Privatsphäre. Vor diesem Hintergrund werden verräterische Teilsegmente einfach so lange aus einzelnen Trajektorien entfernt, bis diese bzgl. des Musters des Angreifers mit ausreichend vielen anderen Datenbankeinträgen übereinstimmt. Um die Aussagekraft der gefilterten Datenbank zu maximieren, werden die Trajektorien derart gekürzt, dass sie in ihrem geographischen Verlauf möglichst wenig Abweichung zur Originaltrajektorie aufweisen. Natürlich besteht dieses Verfahren nur gegen solche Angreifer, die tatsächlich genau über das angenommene Vorwissen verfügen. Weicht das tatsächliche Wissen davon ab, verliert das Verfahren seine Wirksamkeit.

Monreale et al. [178] verfolgen dieselbe Strategie, abstrahieren dabei jedoch von geographischen Orten und nehmen an, dass der Angreifer semantisches Vorwissen über einen Nutzer hat. So weiß er zwar nicht, in welcher Region sich das Ziel zu welchem Zeitpunkt aufgehalten hat, aber er kennt die Semantik und Reihenfolge einiger vom Nutzer besuchter Orte. Um derartige Angriffe zu verhindern, führen die Autoren das Prinzip der *c-Safety* ein, das voraussetzt, dass sich in einem Datensatz mit semantischen Trajektorien keine Einträge befinden dürfen, für die ein beliebiges, zeitlich geordnetes Set an semantischen Orten eindeutig ist. Existieren solche Trajektorien, werden einzelne Orte durch semantische Generalisierung verschleiert. Zu diesem Zweck wird in [178] eine Ontologie semantischer Orte vorgeschlagen, die einzelne Individuen wie ein

bestimmtes Krankenhaus zunächst auf dessen genauen Typ (Klinik) und falls weiterhin nötig auf noch allgemeinere Oberklassen (Gesundheit) abbildet. Somit wird wie beim Prinzip der k-Anonymität verhindert, dass Abfolgen semantischer Orte einen Quasi-Identifier darstellen. Dieser Ansatz ist stärker als [232], da keine Annahmen über das genaue Vorwissen eines Angreifers bekannt sein müssen. Nicht gewachsen ist dieser Ansatz jedoch einem Angreifer, der geographisches Vorwissen über das Vorkommen bestimmter semantischer Orte in einem Gebiet besitzt und die oben vorgenommene Generalisierung somit einfach rückgängig machen kann, um an die Originaltrajektorie zu gelangen.

Huo et al. [127] schützen die Privatsphäre der Nutzer, indem die Bereiche, an denen ein Aufenthalt stattgefunden hat, aus der jeweiligen Trajektorie herausgeschnitten werden. Die Autoren wenden zunächst ein Clustering-Verfahren auf die GPS-Daten aller Nutzer an, um alle Aufenthaltsorte (engl. *stays*) aus den kontinuierlichen Trajektorien zu extrahieren [268]. Da die meisten Orte wiederholt oder von mehreren Benutzern besucht werden, lassen sich charakteristische Eigenschaften für diese Orte berechnen wie ihre Popularität, durchschnittliche Verweildauer und mittlere Ankunftszeit. Orte mit ähnlicher Semantik weisen laut den Autoren auch Ähnlichkeiten hinsichtlich dieser Charakteristika auf. Gemäß dieser Eigenschaften werden schließlich räumlich zusammenhängende Zonen erzeugt, die eine konfigurierbare Zahl an solchen Orten beinhalten, die sich möglichst unähnlich sind. Hierfür wird erneut ein Clustering-Verfahren eingesetzt und als Distanzmaß eine Metrik eingeführt, die räumliche Nähe und semantische Unähnlichkeit als Nähe interpretiert. Abschließend werden die Stays der einzelnen Trajektorien durch die entsprechenden Zonen ausgeblendet, indem alle Punkte gelöscht werden, die vor oder nach einem Stay in der Zone liegen, die diesen Aufenthaltsort somit covert. Damit soll erreicht werden, dass durch den Verlauf der Trajektorie mehrere, semantisch möglichst unterschiedliche Aufenthaltsorte möglich sind. Ein Nachteil dieses Verfahrens ist, dass der Angreifer keine Kontextinformationen abgesehen von den Koordinaten der Trajektoriendatenbank erhalten darf. Wäre er in Besitz der Zeitstempel der einzelnen Datenpunkte, könnte er durch das auf Unähnlichkeit ausgerichtete Clustering doch wieder mit großer Genauigkeit auf den richtigen Aufenthaltsort schließen, z.B. wenn eine Bar und eine Grundschule das Anonymitätsset bilden. Darüber hinaus berücksichtigt dieses Verfahren keine Karteninformationen über die Konnektivität der einzelnen Orte. Verläuft durch eine erzeugte Zone z.B. ein Fluß, kommen als Aufenthaltsorte nur die Gebäude auf der der Trajektorie zugewandten Seite in Frage.

Anders verfaren Naghizade et al. [186], die die genauen Orte von Aufenthalte nicht durch großflächiges Ausblenden erreichen, sondern durch das Einfügen von Alibi-Trajektorien. Vor dem Veröffentlichen einer Trajektorie wird daher geprüft, ob sich der Nutzer an Orten aufgehalten hat, die aus dessen Sicht privatsphärekritisch sind. Ist dies der Fall, werden die Datenpunkte der Trajektorie in der zeitlichen Nachbarschaft des Aufenthals so verfälscht, dass ein Aufenthalt an einem benachbarten, privatsphäreunbedenklichen Ort ent-

steht. Dieses Verfahren ist damit mit den dummybasierten Ansätzen verwandt. Problematisch erscheint auch hier die Erzeugung plausibler, nicht als solche erkennbarer Trajektorienfälschungen sowie – z.B. im Rahmen partizipativer Sensornetze – der negative Einfluss auf die Qualität und Korrektheit der entstehenden Datenbasis.

Insgesamt zeigt sich anhand dieser Arbeiten, dass selbst bei einer anonymen LBS-Nutzung ohne weitere Schutzmaßnahmen davon ausgegangen werden muss, dass der Dienstanbieter dazu in der Lage ist, einzelne Anfragen über die Zeit hinweg einem Nutzer zuzuordnen. Die hierfür vorgestellten Lösungsansätze gehen davon aus, dass der gesamte Datenbestand, d.h., die gesammelten Trajektorien aller Nutzer, zum Zeitpunkt der Datenverschleierung durch die TTP bekannt ist. Für den Echtzeit-Schutz von Standortinformationen bei der LBS-Nutzung lassen sich diese daher nicht einsetzen, v.a. wenn dieser zur Vermeidung einer TTP-Komponente clientseitig umgesetzt werden soll.

2.3 Zusammenfassung

In diesem Kapitel wurde zunächst der grundelegende Kontextbegriff definiert. Ferner wurden die die Grundlagen kontext- und ortsbezogener Dienste erläutert. Im Anschluss wurde eine kurze Übersicht über Möglichkeiten zur clientseitigen Kontext- und Positionsermittlung gegeben.

Die im Rahmen kontextabhängiger Dienste über Sensoren erhobenen Kontextinformationen stellen stets persönliche Daten dar. Gemäß rechtlicher, philosophischer und moralischer Grundsätze sollte der Nutzer daher stets darüber informiert sein, welche Daten von ihm abgefragt werden, von wem, und die Entscheidungsmöglichkeit haben, in welchem Umfang er das zulassen möchte.

Aus diesem Grund wurde im Anschluss eine aktuelle Übersicht über bestehende Konzepte zum Schutz der Privatsphäre in kontextabhängigen und insbesondere ortsbezogenen Diensten gegeben. Letztere stellen die am weitesten verbreitete und daher in wissenschaftlichen Arbeiten auch am meisten untersuchte Ausprägung kontextbezogener Dienste dar.

Aufbauend auf diesen aus der Literatur gewonnenen Erkenntnissen wird im nächsten Kapitel ein umfassendes System zur feingranularen, clientseitigen Verwaltung von Kontextinformationen vorgestellt.

3 Privatsphärezentrische Verwaltung von Kontextinformationen

Der massiven Verbreitung und raschen technologischen Weiterentwicklung von mobilen Endgeräten ist es zu verdanken, dass die Vision ubiquitärer Anwendungen mehr und mehr Einzug in die Realität hält. Viele Forschungsarbeiten beschäftigen sich damit, wie sich solche kontextabhängigen Dienste skalierbar und performant für Millionen von Nutzern, Sensoren und Aktoren umsetzen lassen. Daneben existieren bereits seit einigen Jahren kommerzielle Angebote, die ihren Service auf Basis einzelner Kontextaspekte wie dem aktuellen Standort ihrer Nutzer erbringen. Alle solchen Ansätze – sowohl die akademischen Visionen als auch die kommerziellen Produkte – basieren dabei auf der Ermittlung, Interpretation und Verteilung von Kontextinformationen. So werden hardware- und softwareseitig stets neue Verfahren entwickelt, wie sich der Nutzerkontext kontinuierlich, energiesparend und zuverlässig ermitteln lässt.

Egal, ob es sich um personalisierte ortsbezogene Dienste, kontextbezogene Geschäftsanwendungen oder gesellschaftlich relevante Systeme wie eine *Smart City* handelt: Bei der Umsetzung derartiger Dienste darf der Schutz der Privatsphäre der teilnehmenden Benutzer nicht zu kurz kommen. In diesem Kapitel wird daher ein privatsphäreorientierter Blickwinkel auf die Verwaltung von Kontextinformationen eingenommen, der die Entscheidungshoheit des Nutzers in den Vordergrund stellt. Es steht kein bestimmter Typ kontextbezogener Anwendungen im Fokus, sondern die Suche nach einer allgemein einsetzbaren Lösung für die feingranulare und situationsabhängige Durchsetzung individueller Privatsphärebedürfnisse. Dem Nutzer soll so eine flexible und effektive Kontrolle über seine persönlichen Daten angeboten werden, mit der er festlegen kann, wann ein Dienst oder eine andere Person an welche Informationen über ihn gelangen darf.

3.1 Vorveröffentlichungen

Die in diesem Kapitel beschriebenen Konzepte wurden bereits im Rahmen eines internationalen Konferenzbeitrags [66] sowie in einem Journalartikel [65] vorveröffentlicht.

Der Aufbau und die Inhalte des Kapitels folgen dabei grob dem Journalartikel. Die Auseinandersetzung mit den einzelnen Teilaspekten und den Details

der Umsetzung sowie die Beurteilung des vorgestellten Systems finden in der vorliegenden Arbeit jedoch grundsätzlich sehr viel detaillierter statt.

3.2 Bereitstellung und Verwaltung von Kontextinformationen

In diesem Kapitel wird ALPACA, ein umfassendes System zur clientseitigen Verwaltung von Kontextinformationen präsentiert, das die Privatsphärebedürfnisse des Nutzers in den Vordergrund stellt und zu diesem Zweck verschiedenste Schutzmechanismen integriert. Der Ansatz ist generisch für verschiedene Ausprägungen kontextbezogener Dienste einsetzbar: Einerseits für die Umsetzung der visionären, hochkomplexen Anwendungen, in denen Kontextinformationen aus allen erdenklichen Quellen gesammelt und für die Erbringung hochwertiger Dienste zusammengeführt und interpretiert werden. Andererseits für die schon heute weit verbreiteten mobilen Apps, die über die standardisierten Programmierschnittstellen der mobilen Betriebssysteme längst die Kontextinformationen des Nutzers abfragen, um darauf basierend ebenfalls einen gewissen Grad an Kontextbezug zu ermöglichen.

Um den ersten Teil dieser Zielsetzung zu erreichen, definiert das System eigene, spezialisierte Schnittstellen und stellt sie kontextabhängigen Anwendungen zur Verfügung. Durch die damit verbundenen Gestaltungsmöglichkeiten kann ALPACA seinen vollen Funktionsumfang nutzen und auch komplexe Kontextanfragen beantworten, die z.B. auch vom Kontext des Anfragenden abhängen. Zur Erreichung des zweiten Aspekts lässt sich das System daneben aber auch transparent in bestehende mobile Betriebssysteme integrieren, um dem Nutzer effektive und situationsabhängige Kontrolle über den Zugriff auf persönliche Kontextinformationen durch beliebige, heute verfügbare Smartphone-Anwendungen bieten zu können.

Anhand bestehender Plattformen zur Umsetzung komplexer kontextbezogener Dienste werden im Folgenden die typischen Anforderungen an ein generisch einsetzbares System zur Verwaltung von Kontextinformationen zusammengetragen.

Grundlegende Voraussetzung für die Realisierung kontextbezogener Anwendungen sind funktionaler Natur, wie die Akquise, Aufbereitung und der Austausch entsprechender Informationen mit anderen Entitäten. Während die eigentliche Diensterbringung durch die Erfüllung dieser Anforderungen überhaupt erst ermöglicht wird, ist für die breite Akzeptanz solcher Dienste – zumindest unter einem beträchtlichen Teil der Nutzer – jedoch noch ein weiterer Aspekt unabdingbar: Der zuverlässige Schutz persönlicher Daten, zu denen viele der anfallenden Kontextinformationen zweifelsohne zählen, muss gewährleistet sein, damit sich die Benutzer in solchen intelligenten Umgebungen selbstbestimmt, sicher und nicht überwacht fühlen.

Unter Berücksichtigung bestehender Arbeiten zum Schutz der Privatsphäre in kontextbezogenen Anwendungen werden die rein funktionalen Anforderungen entsprechend ergänzt, um die Grundlage für ein allgemein einsetzbares, *privatsphärezentrisches Kontextmanagementsystem* zu bilden. Ziel dieses Ansatzes ist es, dem Nutzer umfassende Kontrollmöglichkeiten für die Freigabe, Granularität und Validität persönlicher Kontextinformationen zu gewähren.

3.2.1 Funktionale Anforderungen

Plattformen für die Umsetzung kontextbezogener Anwendungen beinhalten typischerweise vier aufeinanderfolgende Stadien der Kontextverwaltung, um den Kontextbezug von Diensten zu ermöglichen [174, 193]: *Kontextakquise* und *Kontextmodellierung*, der komponentenübergreifende *Austausch von Kontextinformationen* sowie deren *Zusammenführung und Auswertung*. In diesem letzten Schritt wird durch Kombination und logische Interpretation mehrerer Kontextinformationen aus unterschiedlichen Quellen die eigentliche Logik der Diensterbringung umgesetzt.

Sowohl die Ermittlung von Kontextinformationen auf mobilen Endgeräten als auch die geeignete Modellierung dieser Daten zur gemeinsamen Nutzung durch unterschiedlichste Anwendungen und Dienste stellen äußerst populäre Forschungsgebiete dar. Die Teilgebiete der Kontext-, Situations- und Aktivitätserkennung konzentrieren sich dabei wie in Kapitel 2.1.3 beschrieben auf die Frage, wie sich z.B. mit Techniken des maschinellen Lernens aus den von den rohen Sensormesswerten des mobilen Endgeräts extrahierten Features semantisch angereichterte Kontextinformationen ermitteln lassen. Neben den Fragen, wie zuverlässig, zeitnah sowie rechen- und energieeffizient diese Erkennung funktioniert, spielt vor allem die Erweiterbarkeit einer Plattform hinsichtlich neu hinzukommender Sensoren und Algorithmen oder die Erkennung fehlerhafter Sensordaten hierbei eine wichtige Rolle [60].

Für die Weiterverarbeitung der so ermittelten „Kontextatome" werden Verfahren benötigt, mit deren Hilfe sich Kontext möglichst universal verständlich darstellen lässt, um den späteren Austausch zwischen verschiedenen Komponenten oder Systemen zu ermöglichen. Hierfür wird insbesondere in den akademischen Ansätzen eine formale Kontextmodellierung verwendet, um zwischen verschiedenen Systemen ein identisches Verständnis der für die Diensterbringung relevanten Informationen zu ermöglichen. In der Literatur werden dazu verschiedene Varianten vorgeschlagen, wobei sich jene Ansätze, welche die Modellierung von Kontextinformationen auf Basis eines ontologiebasierten Ansatzes lösen, als am ausdruckstärksten herausgestellt haben [226].

Der Austausch von Kontextinformationen beschreibt die Weitergabe dieser semantisch angereicherten Kontextinformationen vom Kontexterzeuger an bestimmte Empfänger [15]. Auch in diesem Schritt gibt es unterschiedliche Aspekte zu beachten, wie z.B. die Wahl geeigneter Kommunikationsprotokolle oder der grundsätzliche Systemaufbau. Um Kommunikationseffizienz, Skalier-

barkeit und Interoperabilität zu gewährleisten, wird dabei meist auf zentrale Serverkomponenten gesetzt, die z.B. *Context Broker* [174, 143] oder *Context Management System* [116] genannt werden. Auch gilt es festzulegen, welche Kommunikationsparadigmen eingesetzt werden bzw. in welcher Form die Erzeuger und Verbraucher von Kontextinformationen miteinander Daten austauschen. Hierbei sind sowohl rein anfragebasierte Umsetzungen denkbar als auch proaktive Lösungen oder hybride Ansätze, die je nach Art der kontextabhängigen Diensterbringung unterschiedlich geeignet sind.

Im Rahmen der Auswertung von Kontextinformationen gilt es, die eigentliche Logik eines kontextabhängigen Dienstes zu implementieren. Je nach Anwendung müssen dabei unterschiedliche Zusammenhänge und Situationen durch die Kombination aller zur Verfügung stehenden Kontextinformationen erkannt werden und intelligente Entscheidungen getroffen werden, um Nutzer mit maßgeschneiderten Informationen auszustatten oder selbsttätig Maßnahmen durch die zielgerichtete Steuerung von Aktoren zu ergreifen. Dieser Prozess wird meist mittels logischer Inferenz auf Basis der modellierten Kontextinformationen und dem Einsatz spezialisierter Regelsysteme umgesetzt.

3.2.2 Anforderungen aus Sicht der Privatsphäre

Während die bislang genannten Anforderungen aus funktionaler Sicht unverzichtbar für die Umsetzung kontextbezogener Anwendungen sind, muss darüber hinaus die Dimension der Privatsphäre und des Datenschutzes beachtet werden. Diese Themen stehen im Zentrum der vorliegenden Arbeit und beschäftigen sich mit all jenen Aspekten, die direkt oder indirekt den Schutz persönlicher Daten der teilnehmenden Nutzer betreffen.

Obwohl u.a. in [174] als nötiges Querschnittsthemen beschrieben, werden Datenschutz und Privatsphäre in bestehenden Ansätzen – wenn überhaupt – meist nur nachgelagert betrachtet. Laut [193] sollten Mechanismen zum Schutz der Privatsphäre aber nicht nur auf Anwendungsebene umgesetzt werden, sondern müssen entlang der gesamten Kette von der Ermittlung und Modellierung bis hin zur Verwaltung und Verteilung von Kontext berücksichtigt werden, um effektiv zu sein und eine breite Akzeptanz seitens der Nutzer zu erreichen.

Auch in kommerziellen Systemen wirken die Mechanismen zum Schutz dieser persönlichen Daten oft aufgesetzt, statt integriert, und bieten wenig Möglichkeiten zur Feinjustierung. Im Gegensatz dazu wird in dieser Arbeit versucht, alle Schritte bei der Akquise, Modellierung und Verwaltung von Kontextinformationen privatsphärezentrisch zu gestalten. Wie später zu sehen ist, ermöglicht dies u.a. ein hohes Maß an Flexibilität und Ausdrucksstärke bei der Definition individueller Freigaberegeln.

Aus einer privatsphärezentrischen Perspektive heraus lässt sich folgende Faustregel formulieren: *Je weniger persönliche Informationen das mobile Endgerät des Benutzers verlassen, desto besser.* Im Rahmen dieser Arbeit wird daher argumentiert, dass die absolute Kontrolle über die zur Verfügung ste-

henden Kontextinformationen stets beim Nutzer verbleiben sollte. Idealerweise gibt es daher neben dem Kontexterzeuger keine weitere Entität, die unkontrolliert auf dessen Kontextdaten zugreifen kann. Um dies effektiv gewährleisten zu können, sollte die grundsätzliche Verwaltung und Zugangskontrolle von Kontextinformationen nicht auf einem – in vielen bestehenden Lösungen zur vertrauenswürdigen Partei deklarierten – zentralen Context Broker stattfinden, sondern bereits direkt auf dem Endgerät [122, 22, 8].

Um die Benutzung unterschiedlicher Ausprägungen kontextbezogener Anwendungen zu ermöglichen, besteht i.d.R. jedoch die Notwendigkeit zur Herausgabe gewisser Details des eigenen Nutzungskontext an andere Teilnehmer oder eine solche zentrale Komponente des Dienstanbieters. Das erklärte Ziel lautet daher, bereits auf dem Endgerät des Nutzers effektiv festlegen zu können, welche Informationen unter gewissen Bedingungen an bestimmte Parteien weitergegeben dürfen und welche nicht. Natürlich sind solche Ansätze aber stets komplementär zu einer zentralen Zugangskontrolle auf diesen Context Brokern zu sehen und können diese nicht völlig ersetzen.

Um dem Nutzer bei der Verwendung kontextbezogener Dienste echte Kontrolle über seine persönlichen Informationen zu ermöglichen, werden die funktionalen Anforderungen aus dem vorangehenden Abschnitt im Folgenden ergänzt. Bokhove et al. geben in [35] eine kompakte Übersicht über Möglichkeiten und Anforderungen hinsichtlich der Privatsphäre in kontextabhängigen Diensten. Einzelne Teilaspekte finden sich jeweils auch in den verwandten Arbeiten [185, 206, 133, 132, 34, 218, 8, 122, 255, 254, 145] wieder (vgl. Kapitel 3.3).

Unter Berücksichtigung der genannten Gesichtspunkte muss die Verwaltung von Kontextinformationen auf einem mobilen Endgerät aus Sicht der Privatsphäre die folgenden Anforderungen erfüllen:

1. Der Nutzer muss stets die **Kontrolle über die Verarbeitung persönlicher Informationen** haben. Er muss festlegen können, welche Daten herausgegeben werden, einsehen können, welche Informationen durch welche Parteien abgefragt wurden, und sich darüber informieren können, was mit seinen persönlichen Daten geschieht.

2. Es muss ein effektiver Mechanismus für privatsphärezentrische Verwaltung von Kontextinformationen implementiert sein. Dies gilt es in Form von **situationsabhängigen Freigaberegeln** zu realisieren [24]. Diese Regeln müssen dazu in der Lage sein, den gesamten Nutzerkontext, vorhandene Metainformationen und ggf. auch den Kontext der anfragenden Entität zu berücksichtigen. Entscheidungen hinsichtlich der Freigabe von Kontextinformationen sollen hierbei nicht nur positiv oder negativ ausfallen können, sondern unter Umständen auch zur Preisgabe von unwahren, vagen oder verzerrten Informationen führen.

3. Zudem ist auf die **Vollständigkeit und Konsistenz** dieser Regeln zu achten, d.h., es darf nicht dazu kommen, dass Kontextanfragen aufgrund

lückenhafter oder widersprüchlicher Regeln nicht eindeutig beantwortet werden können. Solche Konfliktsituationen müssen zuverlässig erkannt und aufgelöst werden. Auch ist auf die **zeitliche Konsistenz** aufeinanderfolgender Antworten an einen Kontextempfänger zu achten.

4. Insbesondere bei der Umsetzung von Peer-to-Peer-basierten kontextabhängigen Anwendungen muss eine **Ausgewogenheit** hinsichtlich der zwischen den verschiedenen Peers ausgetauschten Informationen erreicht werden. Es muss sichergestellt werden, dass alle Teilnehmer eines solchen Systems im selben Maße Informationen über sich selbst preisgeben müssen, wie sie über andere erfahren wollen.

5. **Allgemeingültigkeit** sorgt dafür, dass eine Lösung zur Kontextverwaltung generisch für unterschiedliche Typen von Kontextinformationen und Diensten eingesetzt werden kann. Um dies zu gewährleisten, müssen z.B. unter den in Kapitel 2.2.2 vorgestellten Verschleierungsmechanismen jene identifiziert werden, die sich nicht nur auf eine spezielle Domäne von Informationen anwenden lassen, sondern für viele Ausprägungen von Kontext verwendet werden können.

6. Durch eine flexible **Erweiterbarkeit** kann zusätzlich erreicht werden, dass sich darüber hinaus auch spezialisierte Schutzmechanismen für einen bestimmten Typ von Kontextdaten in das System einbinden lassen. Somit können auch bei der für eine Dienstnutzung notwendigen Preisgabe von Kontextinformationen komplexe Schutzziele verfolgt werden, die sich mit den allgemein anwendbaren Verschleierungstechniken nicht umsetzen lassen.

7. Durch die **Berücksichtigung von privatsphärerelevanten Aspekten als Querschnittsthema** kann sichergestellt werden, dass bereits bei der Akquise und Modellierung von Kontextinformationen den Privatsphärebedürfnissen des Nutzers angemessen Rechnung getragen wird. Außerdem kann nur so sichergestellt werden, dass alle im weiteren Verlauf womöglich relevanten Metainformationen mit erfasst werden.

8. Es muss eine möglichst große **Benutzerfreundlichkeit** angestrebt werden, um ein Gleichgewicht zwischen der einfachen Bedienbarkeit und Ausdrucksstärke der Privatsphärekontrolle zu finden. Zu diesem Zweck wird eine intuitive Konzeptualisierung der Privatsphärebedürfnisse eines Nutzers benötigt, die auch automatische Entscheidungen ermöglicht und unnötige Unterbrechnungen und Konfigurationen vermeidet.

Im nächsten Kapitel werden zunächst verwandte Themengebiete und existierende Ansätze zur Verwaltung von Kontextinformationen vorgestellt und hinsichtlich dieses Anforderungskatalogs bewertet. Im Anschluss daran wird in mehreren Schritten ein neuer und umfassender Lösungsvorschlag für die Umsetzung all dieser Anforderungen präsentiert.

3.3 Grundlagen und verwandte Arbeiten

In diesem Abschnitt werden die Grundlagen der Kontextverwaltung erläutert und ein Überblick über verwandte Arbeiten gegeben, die sich ebenfalls mit den Themen Modellierung, Verteilung und dem Schutz von persönlichen Informationen in kontextbezogenen Diensten beschäftigen. Die grundsätzliche Definition des Kontextbegriffs, so wie er in der vorliegenden Arbeit verwendet wird, fand bereits in Kapitel 2.1 statt.

Bei der Modellierung können je nach Einsatzzweck unterschiedliche Ziele beobachtet werden, wobei meist die Herstellung eines gemeinsamen Verständnisses sowie die Möglichkeit zur logischen Inferenz auf den modellierten Fakten im Mittelpunkt stehen. Hinsichtlich der Verwaltung von Kontextinformationen kann man zwischen TTP-basierten Architekturen und einigen wenigen, clientbasierten Ansätzen unterscheiden. Einige dieser Plattformen betrachten dabei bereits Aspekte wie Privatsphäre und Datenschutz und bieten unterschiedliche Lösungskonzepte an, die im Folgenden ebenfalls beschrieben werden. Abschließend werden aktuelle, praxisnahe Lösungen vorgestellt, die sich konkret mit der Verbesserung der Datenschutzeinstellungen und der Kontrolle über Kontextinformationen auf heutigen mobilen Betriebssystemen beschäftigen.

3.3.1 Modellierung von Kontextinformationen

Frühe kontextbezogene Systeme wie das *PARCTAB*-Projekt [245] berücksichtigen jeweils nur bestimmte Typen von Kontextinformationen und sind für einen speziellen Anwendungsfall konzipiert. Eine einfache Wiederverwendung von Kontextinformationen für neue Anwendungen oder die Abbildung logischer Zusammenhänge zwischen verschiedenen Informationen ist durch die enge Vermaschung von Kontextermittlung, Informationsweitergabe und Logik der Diensterbringung nicht möglich. Aufgrund einer fehlenden Formalisierung lässt sich auch nicht automatisch mit Hilfe logischer Regelwerke neues, höherwertiges Wissen aus gemessenen Kontextinformationen schließen oder die Konsistenz der ermittelten Daten sicherstellen.

Um die Entwicklung kontextabhängiger Anwendungen zu erleichtern, findet daher üblicherweise ein Form der Kontextmodellierung statt, bei der die Anwendungsentwicklung von der eigentlichen Akquise und Verwaltung von Kontextinformationen entkoppelt ist [193]. Eine interessante Beobachtung machen Bolchini et al., die konstatieren, dass Kontextmodelle, die versuchen alle möglichen Aspekte von Kontext für jedes erdenkliche Anwendungsszenario zu modellieren, für den Entwickler meist kaum noch handhabbar sind [36]. Stattdessen plädieren sie dafür, genau zu überlegen, welche Funktionalitäten für eine bestimmte Anwendung tatsächlich benötigt werden und ein spezialisiertes Modell zu wählen, das die eigenen Anforderungen besonders gut erfüllt.

Bettini et al. untersuchen verschiedene Modellierungsansätze und formulieren mehrere Anforderungen, die ein Kontextmodell berücksichtigen sollte, um für eine Vielzahl von Anwendungen einsetzbar zu sein [28]. Dazu zählen u.a. die

Heterogenität von Kontextinformationen sowie die Abbildbarkeit von Zusammenhängen zwischen diesen Informationen, der Umgang mit der Unsicherheit von sensorisch ermittelten Kontextinformationen und Möglichkeiten zur logischen Inferenz. Wird eine große Anzahl von Individuen innerhalb des Modells verwaltet, befürchten die Autoren Performance-Engpässe bei der Verwendung von Ontologien und schlagen daher die Verwendung eines hierarchischen, hybriden Kontextmodells vor. Die Konsistenzprüfung des Modellzustands wird jedoch nur von der höchsten, ontologiebasierten Ebene ermöglicht, die als einzige den dafür nötigen Formalisierungsgrad bietet.

Strang et al. untersuchen in ihrem Survey ebenfalls verschiedene Formen der Kontextmodellierung [226], die von einfachen *Key-Value*-Paaren über Markupbasierte und objektorientierte Modelle bis zu hin zu ontologiebasierten Ansätzen reichen. Die Autoren kommen zu der Erkenntnis, dass sich ontologiebasierte Kontextmodelle aufgrund ihrer Möglichkeiten zur verteilten Validierung und zum Schließen neuen Wissens aus modellierten Informationen am besten für diese Aufgabe eignen. So kann durch die Konsistenzprüfung des Modells z.B. sichergestellt werden, dass sich eine Person nicht zur selben Zeit in zwei verschiedenen Räumen aufhalten kann, wodurch sich u.a. die Qualität der kontextabhängigen Diensterbringung erhöhen lässt.

Vor dem Hintergrund dieser Empfehlungen, des großen Verbreitungsgrades von Ontologien bei der Kontextmodellierung und insbesondere wegen der Möglichkeiten zur Konsistenzprüfung des Modellzustands wird auch im Rahmen der vorliegenden Arbeit eine ontologiebasierte Kontextmodellierung gewählt. Im nächsten Abschnitt werden daher die allgemeinen Grundlagen von Ontologien sowie der im weiteren Verlauf für die Implementierung verwendeten Ontologiesprache näher erläutert.

3.3.1.1 Formalisierung von Wissen mit Ontologien

Der Kontext eines Nutzers kann als Wissen über dessen aktuelle Situation verstanden werden [28]. Es liegt somit nahe, bestehende Ansätze zur formalen Repräsentation von Wissen auch für die Modellierung von Kontextinformationen einzusetzen, wie z.B. das Konzept der Ontologien.

Der Begriff „Ontologie" ist der Philosophie entlehnt und beschreibt dort die Lehre vom Seienden. In der Informatik kann unter einer Ontologie „*die explizite Spezifikation einer Konzeptualisierung*" verstanden werden [103]. Eine solche Konzeptualisierung wird an selber Stelle definiert als „*eine abstrakte, vereinfachte Sicht der Welt, die man zu irgendeinem Zweck darstellen möchte*".

Eine derart formale Darstellung von Wissen kann aus unterschiedlichen Beweggründen heraus erfolgen. Ein Grund kann der Wunsch nach einem sog. *shared understanding* sein, also die Herstellung eines formalisierten, maschineninterpretierbaren Verständnisses von Konzepten und Zusammenhängen. Ein weiterer Vorteil ist wie erwähnt in der Möglichkeit zum logischen *Reasoning* auf den formal modellierten Informationen zu sehen. Mit Hilfe des Reasonings lässt sich aus explizit modellierten Fakten implizites Wissen ableiten. Gleich-

zeitig kann damit die Konsistenz einer Ontologie überprüft werden, z.B. ob alle Restriktionen und Wertebereiche einer Relation eingehalten werden.

Ontologiesprachen orientieren sich zu diesem Zweck meist an der Familie der sog. Beschreibungslogiken (engl. *description logics* (DL)), die selbst wiederum ein entscheidbares Fragment der Prädikatenlogik erster Stufe darstellen [123]. Die Repräsentation von Wissen findet dabei auf zwei unterschiedlichen Ebenen statt, deren jeweilige Bedeutung sich sich mit der Unterscheidung zwischen Klassen und Objekten innerhalb objektorientierter Programmiersprachen vergleichen lässt. Während eine Klasse die Eigenschaften und Fähigkeiten einer bestimmten Sache beschreibt, handelt es sich bei einem Objekt um eine einzelne, konkrete Instanz dieser Klasse. Analog dazu besteht eine Ontologie auf konzeptueller Ebene aus *Konzepten* oder *Klassen* (TBox), während die Instanzen eines Konzepts als sog. *Individuen* konkrete Ausprägungen des in der TBox beschriebenen Domänenwissens darstellen (ABox).

Sowohl Konzepte als auch Individuen lassen sich mittels sog. *Properties* miteinander verbinden, um Beziehungen und Zusammenhänge zwischen den jeweiligen Elementen zu modellieren. Es kann zwischen *Objekt-* und *Datatype-Properties* unterschieden werden. Erstere verbinden Individuen mit anderen Individuen der Ontologie, letztere können für die Zuordnung von Individuen und Datenwerten, d.h. Wertliteralen verwendet werden. Zudem lassen sich bei einer Property Definitions- und Wertebereiche bzw. erlaubte Datentypen angeben. So kann z.B. festgelegt werden, dass der Wertebereich einer Property `hasMother` nur Individuen enthalten darf, die dem Konzept `Female` angehören.

Im Rahmen der logischen Inferenz wird versucht, auf Basis der in einer Ontologie hinterlegten Fakten implizites Wissen abzuleiten. Durch logische Schlussfolgerungen bezüglich der Klassen und Individuen eines Modells können somit neue Erkenntnisse entstehen, die nicht explizit angegeben wurden. So kann z.B. aus den beiden Fakten, dass das Individuum Alice ein Mensch und die Mutter von Bob ist, ihre Zugehörigkeit zum Konzept `Female` abgeleitet werden. Zugleich lassen sich sowohl die Elemente der TBox als auch der ABox auf Konsistenz prüfen, um ein Modell zu validieren.

Ontologien weisen im Vergleich zu anderen Formen der Wissensrepräsentation einige Besonderheiten auf, auf die es im weiteren Verlauf zu achten gilt. Ein wichtiger Aspekt ist die sog. *Open World Assumption*, die sich grundsätzlich anders auswirkt als das *Closed World*-Verständnis, das z.B. Datenbanken zugrunde liegt. Die damit verbundenen unterschiedlichen Interpretationsweisen lassen sich anhand eines einfachen Beispiels erklären: Angenommen, als einziger Fakt wäre die Tatsache bekannt, dass Alice ein Android-Smartphone besitzt. In einer Closed World wird davon ausgegangen, dass Fakten, die nicht hinterlegt wurden, auch nicht zutreffen. Die Antwort auf die Frage, ob Alice auch ein iPhone besitzt, würde hierbei demnach „nein" lauten. Unter der Open-World-Assumption hingegen wird angenommen, dass nicht hinterlegte Fakten schlichtweg noch nicht bekannt sind. Die Antwort auf dieselbe Frage wäre also „unbekannt". Vor dem Hintergrund der verteilten Erstellung und

Komponierbarkeit von Ontologien gilt auch nicht die sog. *Unique Names Assumption*. Zwei unterschiedlich benannte Individuen können daher ein und dieselbe Entität darstellen oder zwei Konzepte dieselbe Sache beschreiben. Sollen gewisse Konzepte als disjunkt oder Individuen als paarweise verschieden gekennzeichnet werden, muss dies explizit ausgedrückt werden.

Eine Sprache, die für diese explizite Spezifikation von Wissen häufig zum Einsatz kommt, ist die vom W3C spezifizierte *Web Ontology Language* (OWL) [239]. Eine in OWL geschriebene Ontologie wird als RDF-Graph [241] dargestellt. Das *Resource Description Framework* (RDF) ist ebenfalls eine W3C-Empfehlung, welche die Darstellung von Informationen mit Hilfe eines aus *Subjekt-Prädikat-Objekt*-Tripletts bestehenden Graphen definiert. Die Serialisierung dieser Informationen kann z.B. auf Basis von XML umgesetzt sein:

```
1  <Subject rdf:ID="subject_name">
2    <predicate rdf:resource="#Object" />
3    ...
4  </Subject>
```

Wie durch die drei Punkte angedeutet, können hierbei auch in einem Schritt für ein Subjekt mehrere Prädikat-Objekt-Aussagen angegeben werden.

OWL existiert in drei verschiedenen Varianten. *OWL Lite* ermöglicht die Erstellung einfacher Taxonomien und Klassifizierungshierarchien, kennt aber z.B. keine Definition disjunkter Konzepte [238]. *OWL DL* basiert auf dem Ansatz der Beschreibungslogiken und bietet dadurch ausdrucksstarke Möglichkeiten zur Erstellung von Ontologien. Insbesondere ist OWL DL äquivalent zur entscheidbaren Beschreibungslogik $\mathcal{SHOIN}(D)$ [124] und erlaubt somit u.a. die Verwendung von inversen Properties, Kardinalitätsrestriktionen und Datatype-Properties. *OWL Full* verzichtet auf einige dafür notwendige Einschränkungen und verliert dadurch die Eigenschaft der Entscheidbarkeit. In einer Untersuchung von knapp 1300 Ontologien kommen Wang et al. zu dem Ergebnis, dass nur 61 tatsächlich Gebrauch von dem Funktionsumfang von OWL Full machen, während die übrigen mit den Konstrukten von OWL Lite oder DL auskommen [243]. 2009 wurde als Erweiterung von OWL DL die aktuelle Version OWL 2 vorgestellt, die in einigen Aspekten um mehr Ausdrucksstärke ergänzt wurde, aber immer noch entscheidbar ist [240]. OWL 2 entspricht der Beschreibungslogik $\mathcal{SROIQ}(D)$ [123], und erlaubt nun – wie anhand von $\mathcal{Q}$ ersichtlich ist – u.a. auch die Definition qualifizierter Kardinalitätsrestriktionen. Aufgrund dieser Fähigkeit wird OWL 2 auch im Rahmen der vorliegenden Arbeit für die Kontextmodellierung verwendet.

3.3.1.2 Bestehende Kontextmodelle

Die *CONtext ONtology* (*CONON*) von Wang et al. setzt sich aus mehreren Teil-Ontologien zusammen [244]. Die sog. *Upper Ontology* beschreibt grundlegende Informationen wie Personen, Orte, Geräte und Aktivitäten und wird von domänenspezifischen Ontologien ergänzt, die einzelne Umgebungen wie

z.B. das Zuhause, den Arbeitsplatz oder das Fahrzeug eines Nutzers charakterisieren. Die möglichen Aktivitäten eines Nutzers teilen sich in geplante (z.B. Kalendereinträge) und von Sensormesswerten abgeleitete Aktiväten auf. Im Rahmen dieses Modells werden keine Datenschutzaspekte berücksichtigt. Stattdessen soll das Kontextmodell v.a. das Reasoning von komplexen Zusammenhängen auf Basis der zur Verfügung stehenden Sensoren ermöglichen und die einfache Erstellung kontextabhängiger Anwendungen und Prototypen erlauben. Die Umsetzung des Modells findet mit OWL statt, wodurch mit Hilfe von nutzerdefinierten Inferenzregeln höherwertige Kontextinformationen aus einfachen Beobachtungen geschlossen werden können, um darauf aufbauend z.B. das Verhalten von Geräten in der Umgebung des Nutzers anzupassen.

Korpipää et al. stellen eine Ontologie für die Kontextmodellierung auf einem mobilen Endgerät vor [146, 147]. Auch hierbei spielen Datenschutz und Privatsphäre keine Rolle. Stattdessen stehen u.a. die Ausdrucksstärke des Modells im Vordergrund sowie Möglichkeiten zur logischen Inferenz, zur einfachen Erweiterbarkeit und zur flexiblen Darstellung möglichst vieler, heterogener Kontextinformationen. Zu jeder Kontetxinformation werden hier der Kontexttyp, der aktuelle Wert, Konfidenz, Ursprung und Zeitpunkt gespeichert. Die Modellierung folgt einer Baumstruktur, deren Detailgrad sich zu den Blättern hin erhöht und beinhaltet Kontexttypen, welche die Umgebung, die Nutzer- und die Geräteaktivität charakterisieren.

Das Kontextmodell *MUSIC* von Reichle et al. [199] zielt darauf ab, sowohl die Entwicklung kontextabhängiger Dienste als auch den Austausch von Kontext zwischen verschiedenen Systemen zu vereinfachen. Zu diesem Zweck integrieren die Autoren eigens eine Ebene in ihr Modell, die sich um die automatische Serialisierung von Kontext z.B. als XML oder JSON kümmert. Genau wie das *ASC*-Modell von Strang et al. [227] geht dieses Kontextmodell davon aus, dass derselbe Typ von Kontextinformation von verschiedenen Quellen ermittelt werden kann und daher – bei gleicher semantischer Bedeutung – unterschiedliche Darstellungsformen haben kann. Aus diesem Grund ermöglicht das Modell z.B. die Speicherung von Zeitpunkten auf unterschiedliche Arten, wie z.B. als Triplett aus Tag, Montag und Jahr oder in einem beliebigen anderen Datumsformat. Zur Umrechnung von einer Darstellungsart in eine andere sehen sowohl das *ASC*-Modell als auch *MUSIC* die Bereitstellung passender *Inter-Representation-Operations* vor. Genau wie *CONON* ist *MUSIC* als Kombination aus einer allgemein einsetzbaren Top-Level-Ontologie sowie domänenspezifischer Erweiterungen ausgelegt. Die Autoren nennen die Berücksichtigung von Privatsphäreaspekten bei der Kontextmodellierung als mögliches Ziel, wählen für ihr Kontextmodell jedoch unbedenkliche Anwendungsszenarien und räumen entsprechenden Konzepten daher keinen Platz ein.

Chen et al. schlagen die die *Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA)* vor [47]. Diese besteht aus mehreren Teilen, die sich unmittelbar an der Umsetzung konkreter Einsatzszenarien orientieren: Mit einem Fokus auf die Realisierung eines intelligenten Besprechungsraums stellt

COBRA-ONT von Chen et al. eine Familie von Ontologien für kontextabhängige Umgebungen dar [44]. Die Umsetzung der Ontologien geschieht auf Basis von OWL und beinhaltet die Modellierung von örtlichen Zusammenhängen, Personen und Software-Agenten sowie den Standort und die Aktivität einer Person. *SOUPA* beinhaltet eine eigene *Policy*-Ontologie, mit deren Hilfe Inferenz über die Rechte von Agenten bezüglich der Ausführung bestimmter Aktionen innerhalb einer solchen intelligenten Umgebung betrieben werden kann. Während der Fokus auf der Rechteverwaltung bzgl. des Zugriffs auf Objekte der intelligenten Umgebung liegt und auch keine verschiedenen Granularitäten von Kontext berücksichtigt werden, kann ein Nutzer damit immerhin einfache Zugriffsregeln definieren, wer auf seine Informationen zugreifen darf.

Ohne auf die Umsetzung eines Kontextmodells Bezug zu nehmen, schlagen Lederer et al. zum Schutz der Privatsphäre eine Einteilung von Kontextinformationen in vier Granularitätsstufen vor, die von exakten Informationen über zwei Vergröberungsstufen bis hin zu „unbekannt" reicht [159]. Es scheint fraglich, ob sich mit einer so einfachen Kategorisierung ein echte Balance zwischen Privatsphäre und Dienstqualität erreichen lässt. Aus diesem Grund gehen Wishart et al. von einer Form der Kontextmodellierung aus, die eine homogene Eingruppierung verschiedener Typen von Kontextinformationen auf vordefinierte Granularitätsstufen erlaubt [254]. Die Anzahl an verschiedenen Detailstufen ist hierbei jedoch nicht wie in [159] auf einen bestimmten Wert fixiert, sondern abhängig vom jeweiligen Kontexttyp. Nutzer können damit beispielsweise festlegen, dass ein bestimmter Empfänger von Kontextinformationen grundsätzlich nur Detailstufe x einsehen darf. Für die technische Umsetzung dieser verschiedenen Detailebenen schlagen die Autoren für die Kontexttypen Ort und Aktivität die Verwendung von hierarchischen Taxonomien vor, die in einer Baumstruktur jeder Darstellung einer Kontextinformation eindeutig eine übergeordnete, weniger detaillierte Darstellung zuweisen [255]. Ähnlich gehen Chen et al. vor, beschränken sich dabei aber auf die hierarchische Modellierung möglicher Standortangaben in einem Gebäude [45].

Es existiert eine Vielzahl weiterer, anwendungsspezifischer Kontextmodelle. Umfassende Vergleiche bestehender Ansätze finden sich z.B. in [16, 23, 193, 194]. Während alle diese Übersichtsarbeiten Sicherheit und Privatsphäre als entscheidende Faktoren für den Erfolg kontextbezogener Anwendungen ansehen, kommen sie einhellig zu demselben Schluss, dass diesbezüglich erheblicher Nachholbedarf besteht. So merken z.B. Bellavista et al. 2012 kritisch an, dass sich der Großteil der Forschungsarbeiten im Bereich der kontextabhängigen Dienste insbesondere mit der Modellierung und Bereitstellung von Kontext beschäftigt und sicherheitsrelevante Aspekte hintanstellt – was nun ein Hürde für die Akzeptanz und Verbreitung kontextabhängiger Dienste darstellt [23].

Aus diesem Grund werden im nächsten Abschnitt bekannte Arbeiten aus der Literatur hervorgehoben, die sich explizit mit dem Schutz der Privatsphäre in kontextbezogenen Diensten beschäftigen.

3.3.2 Privatsphäre in kontextabhängigen Anwendungen

Neben der Akquise und Modellierung von Kontext spielt vor allem die Verwaltung und Verteilung dieser Informationen eine entscheidende Rolle. Perera et al. untersuchen 50 kontextabhängige Systeme auf die Berücksichtigung von sicherheits- und privatsphärerelevanten Aspekten und kommen zu dem Ergebnis, dass nur elf der untersuchten Ansätze überhaupt entsprechende Vorkehrungen aufweisen [193]. Während sich viele Arbeiten also hauptsächlich mit der Skalierbarkeit und Performanz kontextabhängiger Systeme beschäftigen, werden im Folgenden solche vorgestellt, die sich dem Thema Privatsphäre aktiv annehmen und entsprechende Schutzmechanismen vorsehen.

3.3.2.1 TTP-basierte Kontextverwaltung

Die meisten Umsetzungen kontextabhängiger Anwendungen basieren auf dem Einsatz einer zentralen Server-Komponente, die in Form einer Middleware die Akquise und Verwaltung von allen zur Verfügung stehenden Kontextinformationen ihrer Nutzer übernimmt [223, 185, 206, 34, 218, 145].

LocServ [185] stellt einen sog. Location Provider dar, der als Proxy zwischen den Nutzern ortsabhängiger Dienste und den Diensten selbst vermittelt. Die Nutzer können Regeln festlegen, zu welchen Tageszeiten bestimmte Dienste oder andere Nutzer ihren Standort auf einer frei wählbaren Detailstufe abfragen dürfen. Weitere Typen von Kontextinformationen werden nicht berücksichtigt, auch wird nicht angesprochen, wie inkonsistente Regeln ggf. erkannt oder aufgelöst werden können. In der Literatur finden sich eine ganze Reihe weiterer solcher zentralen Location Server, die meist noch weitere Schutzziele wie z.B. die Herstellung von k-Anonymität bei der Weiterleitung einer Anfrage an einen ortsbezogenen Dienst ermöglichen. Solche TTP-basierten Systeme wurden bereits in Kapitel 2.2.3 beschrieben.

Sacramento et al. stellen mit dem *Context Privacy Service (CoPS)* [206] eine Erweiterung der *Mobile Collaboration Architecture* (MoCA) [207] vor. *CoPS* ermöglicht die Angabe von Freigaberegeln, die individuell oder systemweit definiert werden können und von der zentralen Komponente, dem *Context Service* überwacht werden. Die Regeln erlauben die Auswahl bestimmter Detailstufen der freigegebenen Kontextinformationen, lassen sich jedoch nur unter Bezug auf die Identität der abfragenden Entität festlegen. Die Definition kontextabhängiger Regeln ist nicht möglich. Um verschiedenen Nutzertypen gerecht zu werden, kann zwischen einem Blacklist- und einem Whitelistverfahren ausgewählt werden. Auf die Ausgewogenheit des Informationsflusses zwischen zwei Nutzern wird nicht geachtet, wobei ein Nutzer immerhin angeben kann, dass er benachrichtigt oder immer gefragt werden möchte, wenn auf seinen Kontext zugegriffen wird. Zur Auflösung widersprüchlicher Freigaberegeln betrachten die Autoren die Spezifität der Regeln und wählen die detaillierteste aus.

Die *Context Privacy Engine (CPE)* [34] von Blount et al. stellt eine Erweiterung der klassischen rollenbasierten Zugangsverwaltung dar. Die Autoren

erweitern hierfür den ACL-Ansatz (engl. *Access Control List*) um zusätzliche kontextabhängige Einschränkungen. Die Regeln, die sich innerhalb von *CPE* erstellen lassen, können sich dabei sowohl auf die aktuelle Situation des Nutzers und die Identität der anfragenden Entität als auch auf deren eigenen Kontext beziehen. Die Freigaberegeln, die erneut zentral von der TTP durchgesetzt werden, ermöglichen jedoch keine Auswahl unterschiedlicher Detailstufen der freigegebenen Informationen. Da zudem keine Mechanismen zur Durchsetzung einer Ausgewogenheit des Informationsaustausches existieren, kann ein Nutzer beliebige Anforderungen an den Kontext des Anfragenden stellen. Durch geschickte Regelformulierung kann dies die Privatsphäre des anfragenden Nutzers gefährden, der dadurch – ohne es zu merken – u.U. viele Informationen von sich preisgibt, wenn er den Kontext eines anderen Nutzers abfragt. Um diese Gefahr einzuschränken, lässt sich für einen Nutzer hierbei nicht in Erfahrung bringen, wie eine ihn betreffende Kontextanfrage von *CPE* beantwortet wurde. Dieses Verfahren steht der Ausgewogenheit des Informationsaustausches damit aktiv entgegen und verhindert, dass ein Nutzer von der Herausgabe persönlicher Informationen an andere erfährt. Regelkonflikte werden über vom Nutzer festlegbare Prioritäten entschieden sowie im Fall uneindeutiger Angaben wie bei CoPS über die Spezifität der Regeln gelöst.

Sheikh et al. beschreiben die Berücksichtigung der Qualitätseigenschaften von Kontextinformationen (engl. *Quality-of-Context*, QoC) zum Schutz der Privatsphäre in kontextabhängigen Anwendungen [218]. Die Kontextverwaltung findet auch hier in einem zentralen Context Broker statt. Die Nutzer können kontextabhängige Regeln festlegen, mit deren Hilfe die Qualität der freigegebenen Kontextinformationen entlang verschiedener Dimensionen ggf. künstlich reduziert werden kann. Hierzu zählen Genauigkeit, räumliche und zeitliche Auflösung, Alter der Informationen und die Wahrscheinlichkeit, dass eine Beobachtung korrekt ist (engl. *Probability of Correctness*, PoC). Der letzte Aspekt berücksichtigt, dass verschiedene Sensoren und Klassifizierungsalgorithmen mit unterschiedlich hohen Erkennungs- und Fehlerraten arbeiten. Durch eine künstliche Verschlechterung der PoC kann somit das Konzept der plausiblen Abstreitbarkeit (engl. *plausible deniability*) umgesetzt werden. Darüber hinaus argumentieren die Autoren, dass in bestimmten Situationen der Nutzer auch einfach falsche Angaben machen möchte, d.h., lügen. Mögliche Regelkonflikte werden hierbei nicht berücksichtigt, was insbesondere bei der Ermöglichung von falschen Aussagen aber wichtig ist. Ebenso wenig wird auf eine Ausgewogenheit im Informationsfluss zwischen Nutzern geachtet.

Abgeleitet von Erkenntnissen aus der Sozialwissenschaft schlagen Jiang et al. in [132] das „*Principle of Minimum Asymmetry*" zum Schutz der Privatsphäre in kontextabhängigen Anwendungen vor. Deren Modell sieht zwei Möglichkeiten vor, wie sich stark einseitige Informationsflüsse vermeiden lassen: Entweder, indem der Fluss von Informationen vom Nutzer an den Datensammler reduziert wird oder indem der Fluss von Informationen in die Gegenrichtung erhöht wird. Im Falle eines ortsbezogenen Dienstes würde dies z.B. bedeuten,

dass der Dienst detaillierte Informationen darüber gibt, welche Daten er von Alice kennt, welche er davon weitergibt und wie sie gespeichert werden.

Damit allein lässt sich zwar noch kein echter Schutz der Privatsphäre erreichen, aber insbesondere in kontextabhängigen Anwendungen, in denen Nutzer untereinander Daten austauschen, stellt dies ein vielversprechendes Instrument dar. Eben diesen Grundsatz verfolgen Kofod-Petersen et al. in [145]. In deren Beispielszenario verändern sie das Serverprotokoll eines Instant Messenger Dienstes derart, dass Status und Standortangaben anderer Nutzer für einen Teilnehmer nur dann sichtbar sind, wenn er dieselben Informationen auch zur Verfügung stellt. Die Berücksichtigung von Kontextinformationen oder die Erstellung von kontextabhängigen Freigaberegeln wird hierbei nicht thematisiert. Stattdessen kann der Nutzer die Dienstnutzung manuell an- oder abschalten.

Die verschiedenen Ansätze treffen jeweils unterschiedliche Vorkehrungen, die den Schutz der Privatsphäre in kontextabhängigen Diensten gewährleisten sollen. Keiner der genannten Ansätze erfüllt alle der in 3.2.2 aufgelisteten Anforderungen. Die Ausgewogenheit beim Informationsaustausch zwischen den verschiedenen Aktoren kontextabhängiger Anwendungen findet z.B. nur in [145] Berücksichtigung.

Hinzu kommt bei all diesen Verfahren das grundsätzliche Problem der allwissenden, als vertrauenswürdig deklarierten zentralen TTP-Komponente. Da diese jeweils alle Informationen über die Nutzer des Systems besitzt, stellt sie ein lukratives Angriffsziel dar oder kann selbst als potentieller Angreifer angesehen werden, der die Daten seiner Nutzer entweder selbst mißbraucht oder mit einzelnen Teilnehmern des Systems kollaboriert, um andere auszuspionieren.

3.3.2.2 Clientseitige Kontextverwaltung

Im Gegensatz zu den bisher genannten Systemen überlassen einige Arbeiten die Kontextverwaltung nicht einer zentralen Serverkomponente, sondern führen diese direkt auf dem Endgerät des Nutzers durch.

Apolinarski et al. beschreiben einen Peer-to-Peer-basierten Ansatz für die privatsphäreschonende Verwaltung und Weitergabe von Kontextinformationen [8]. Ähnlich wie die vorliegende Arbeit argumentieren die Autoren, dass eine clientseitige Verwaltung dieser persönlichen Informationen vorzuziehen ist. Um dies umzusetzen, verwaltet jedes Endgerät seinen eigenen Kontext in einem lokalen Kontextmodell. Zur Identifikation und Authentifizierung anderer Geräte, Nutzer und Institutionen werden asymmetrische Schlüsselpaare, Zertifikate und Signaturen realisiert. Um dabei verschiedenen Nutzern unterschiedliche Detailstufen von Information zukommen lassen zu können, wird der Einsatz transitiver Vertrauenslevel eingeführt. Ein höherer Vertrauenslevel impliziert dabei den Zugriff auf alle unteren Level. Der Grad an Vertrauen, der einer anderen Entität entgegengebracht wird, ist dabei an deren Zertifikat gebunden, sodass ein Nutzer stets denselben Detailgrad über alle Kontexttypen hinweg abfragen kann. Um bei einer eingehenden Kontextanfrage entscheiden zu können, welche Information zurückgegeben werden darf, wird jedes Triplett

des OWL-basierten Kontextmodells mit dem entsprechenden Level markiert. Während der Ansatz damit die Freigabe unterschiedlicher Detailstufen von Kontextinformationen vorsieht, erlaubt er nicht die Angabe kontextabhängiger Freigaberegeln. Das Auftreten widersprüchlicher Freigaberegeln wird durch die fixe Zuteilung von Kontextinformationen und anfragenden Entitäten zu einem bestimmten Vertrauenslevel implizit umgangen. Das Vertrauenslevel ist dabei unidirektional, d.h., dass zwei Parteien sich gegenseitig unterschiedlich viel Vertrauen entgegenbringen können. Die Forderung nach der Ausgewogenheit im Informationsaustausch kann somit nicht erfüllt werden.

Eine weitere Möglichkeit zur clientseitigen Verwaltung von Privatsphärepräferenzen stellt die *Context-Aware Privacy Policy Language* (CPPL) von Behrooz et al. [22] dar. Die Regeln basieren auf Situationsbeschreibungen, die mit den Begriffen und Klassen der MUSIC Ontologie [199] erzeugt werden können. Die Regelauswertung ist getrennt vom Modell gehalten, welches dementsprechend auch keine expliziten Maßnahmen für die Unterstützung kontextabhängiger Freigaberegeln bietet. Eine Besonderheit ist, dass die Freigaberegeln, die ein Nutzer zur Verwaltung seiner Kontextinformationen angeben kann, nicht nur auf seine aktuelle Situation Bezug nehmen können, sondern auch die Regelauswertung selbst kontextabhängig umgesetzt wird. Durch Vorfilterung der Freigaberegeln müssen so nicht alle hinterlegten Regeln getestet werden, sondern nur jene, die für den aktuellen Nutzungskontext relevant sind. Hierdurch lassen sich bei der Beantwortung von Kontextanfragen unnötige Regelauswertungen vermeiden, was die Performanz der Regelauswertung erhöht. Die Regeln selbst lassen sich auf Basis des Nutzerkontexts und der sozialen Beziehung zum anfragenden Nutzer definieren. Der Kontext der anfragenden Entität kann nicht berücksichtigt werden und auch auf eine Ausgewogenheit des Informationsaustausches wird nicht geachtet. Zur Auflösung widersprüchlicher Freigaberegeln schlagen die Autoren vor, eine pauschale „*denyOverrides*"- oder „*permitOverrides*"-Strategie anzuwenden. Wie solche Widersprüche erkannt werden können, wird jedoch nicht erläutert.

Der Ansatz von Gosh et al. [93] ähnelt bzgl. der Umsetzung des Reasonings über die Privatsphärepräferenzen des Nutzers dem hier vorgestellten System am meisten. Die Autoren gehören derselben Forschergruppe an, die auch die zentrale Kontextverwaltungs-Plattform *CoBrA* konzipiert hat. Basierend auf einer Kontextmodellierung mit OWL und dem Einsatz des Semantic Web-Frameworks Jena[1], wird eine prototypische Implementierung einer erweiterten, kontextabhängigen Verwaltung von Standortinformationen unter Android präsentiert. Abhängig von dem Vertrauen, dass ein Nutzer einer bestimmten Anwendung gegenüber hat, wird dieser entweder der Zugriff auf die echten Standortkoordinaten gewährt oder eine zufällige Fake-Position zurückgeliefert. Während dieser Ansatz relativ simpel ist, beweist er die technische Umsetzbarkeit des ontologiebasierten Reasonings auf Basis des aktuellen Nutzungskontexts auf einem mobilen Endgerät, was einen enormen Fortschritt gegen-

[1] `https://jena.apache.org/`

über der heute auf Smartphones verfügbaren, binären Entscheidung über die Standortfreigabe darstellt. Regelkonflikte, die laut den Autoren gerade bei der kontextabhängigen Regeldefinition überaus wahrscheinlich sind, werden konservativ entschieden. Verbietet eine aktuell zutreffende Regel die Freigabe von Daten, werden keine Daten versendet.

In einer direkten Vorarbeit zu diesem System beschreiben Jagtap et al. auch die Möglichkeit, den Kontext eines anfragenden Nutzers bei der Definition von Freigaberegeln zu berücksichtigen [130, 131]. Sie gehen jedoch nicht darauf ein, wie dieser zur lokalen Regelauswertung zur Verfügung stehen kann, wenn jedes Endgerät doch seinen eigenen Kontext verwaltet und vor unzurechtmäßigem Zugriff schützt. Auch gibt es keine Vorkehrungen, die versuchen eine Ausgewogenheit der zwischen zwei Nutzern ausgetauschten Menge und Detailstufe an Informationen sicherzustellen.

Insgesamt ist damit auch keiner dieser clientbasierten Ansätze zur Verwaltung von Kontextinformationen dazu in der Lage, alle der zuvor aufgestellten Anforderungen erfüllen. Sie verlagern die Datenhoheit jedoch weg von einer allwissenden, zentralen Verwaltungskomponente auf das Endgerät des Nutzers. Insbesondere das letzte System bietet daher einen vielversprechenden Ausgangspunkt für die Umsetzung einer flexiblen, situationsabhängigen Kontextverwaltung, die direkt auf dem Endgerät des Nutzers stattfindet.

3.3.2.3 Praktische Lösungsansätze auf Android

Neben diesen oft theoretisch geprägten Ansätzen wurden insbesondere in der jüngsten Vergangenheit auch eine Reihe praxisnaher Lösungen vorgestellt. Diese beschäftigen sich konkret mit Möglichkeiten, den Datenschutz auf einem aktuellen mobilen Betriebssystem zu verbessern, indem sie dem Nutzer mehr Entscheidungs- und Kontrollmöglichkeiten bieten.

Beresford et al. gehen mit *Mockdroid* [25] einen ersten Schritt in diese Richtung. Mit dieser modifizierten Version des Android-Betriebssystems ist es möglich, Anwendungen vorzugaukeln, sie hätten Zugriff auf einen bestimmte Resource ohne diesen tatsächlich zu gewähren.

Android stellt den Anwendungsentwicklern das sog. *Application Framework* zur Verfügung – eine API, mit deren Schnittstellen auf die verschiedenen Systemresourcen zugegriffen werden kann. Für die verschiedenen `Permission`-Typen, denen ein Nutzer bislang bereits bei der Installation einer App zwangsweise zustimmen musste, setzen die Autoren jeweils unterschiedliche Schutzmechanismen ein: So wird die Standortabfrage schlichtweg nicht beantwortet, so als ob der GPS-Modul nicht in der Lage wäre, einen Position-Fix zu berechnen. Der Netzwerkzugang wird verwehrt, indem man die Sockets, die von einer App geöffnet werden, grundsätzlich in ein Timeout laufen lässt anstatt den Verbindungsaufbau zu gestatten.

Für die Umsetzung wird der Android Package Manager, der für die Überprüfung der App-Berechtigungen zuständig ist, modifiziert und für jede original Permission eine *Mock-Permission* agelegt. Mit Hilfer dieses Mock-Permissions

kann der Nutzer über ein eigenes Userinterface einer App nachträglich Rechte entziehen. Bei einem kritischen API-Aufruf werden nun zunächst wie gewöhnlich die Standardberechtigungen überprüft. Hat eine App eine Berechtigung bei der Installation angefordert und erhalten, wird im nächsten Schritt nach möglichen Mock-Permissions gesucht, mit denen der Nutzer diese Entscheidung ggf. rückgängig gemacht hat. Wird eine solche gefunden, wird wie oben beschrieben der Zugang zu den echten Resourcen verhindert.

Seit Ende 2015 sieht Android in der Version 6.0 vor, dass Nutzer einer App auch nach der Installation die Freigabe auf bestimmte Kontextquellen erteilen oder entziehen können. Eine nachträgliche Modifikation des Betriebssystems, wie *Mockdroid* sie ermöglicht hat, wurde damit weitgehend überflüssig. Doch auch das aktuelle Berechtigungssystem von Android bietet bei weitem nicht die umfangreichen Einstellungsmöglichkeiten, wie man sie dem Nutzer gemäß den in Kapitel 3.2.2 beschriebenen Anforderungen zur Verfügung stellen sollte. Es wird dadurch weiterhin nur eine binäre Entscheidung ermöglicht, ob man einer App alle Informationen eines bestimmten Kontexttyps zugestehen möchte – oder keine.

Während *Mockdroid* keinen Kontextbezug bei der Freigabe von Resourcen berücksichtigt, präsentieren Shebaro et al. [217] eine Lösung, die auf Basis der aktuell sichtbaren WLAN-Access Points und deren empfangbarer Signalstärke verschiedene Funktionalitäten des Betriebssystems deaktiviert. Die Autoren bedienen sich damit des Prinzips des WLAN-Fingerprintings (vgl. 2.1.3.2.2), um ortsbasierte Freigabeentscheidungen zu ermöglichen. Der Nutzer kann für die jeweils aktuelle „Situation" (d.h., den derzeit empfangbaren WLAN-Fingerprint) u.a. festlegen, dass innerhalb der aktuellen Umgebung der Android-interne Intent-Mechanismus zur Interprozesskommunikation geblockt wird und Permissions vorübergehend entzogen werden.

Für die technische Realisierung wird erneut das Betriebssystem an den entsprechenden Stellen modifiziert. So werden auch hier die Schnittstellen des *Application Frameworks* so angepasst, dass sie eine kontextabhängige Überprüfung der Systemberechtigungen erlauben. Insbesondere wird die Methode `checkComponentPermission(...)` innerhalb des Package Managers überschrieben, um neben den eigentlichen Permissions auch die aktuelle Situation in die Entscheidung einbeziehen zu können. Die Lösung ist im Hinblick auf die in Kapitel 3.2.2 aufgestellten Anforderungen insofern unvollständig für den Schutz der Privatsphäre, als dass sich mit dieser Methode z.B. keine feingranularen oder rezipientenabhängigen Freigaberegeln erstellen lassen. Diese Arbeit zeigt jedoch eindrucksvoll die technische Umsetzbarkeit kontextabhängiger Privatsphärepräferenzen auf einem mobilen Betriebssystem.

Fawaz et al. stellen gleich zwei praktische Umsetzungen der Verschleierung von sporadisch abgefragten Standortinformationen unter Android vor. Während *LP-Guardian* [79] wie die zuvor beschriebenen Ansätze eine grundlegende Modifikation des Betriebssystems erfordert, wurde dessen Nachfolger-Implementierung *LP-Doctor* [78] in Form eines sog. *Launchers* implementiert.

Diese lassen sich unter Android als Alternative zum mitgelieferten Homescreen auch nachträglich installieren. In seiner Funktion als Launcher ist *LP-Doctor* dazu in der Lage, den Start einer App zu unterbrechen und zunächst den aktuellen Ort zu prüfen. Befindet sich der Nutzer an einem Ort, für den er bereits eine Präferenz hinsichtlich der Verschleierung dieses Ortes angegeben hat, wird diese Entscheidung wiederholt. Handelt es sich um einen neuen Ort, wird der Nutzer gefragt, ob er diesen Ort vor der App verschleiern möchte oder nicht.

Soll der Ort verschleiert werden, benutzt *LP-Doctor* den in Android zu Debugging-Zwecken integrierten *MockLocationProvider*, um die Anwendung mit einem gefälschten Standort zu versorgen. Als Verschleierungsmechanismen setzen beide Implementierungen auf das Prinzip der *Geo-Indistinguishability* [6], bei dem auf die tatsächlichen Koordinaten des Standorts ein aus einer Laplace-Verteilung entnommenes Rauschen addiert wird. Dieses Verfahren ist grundsätzlich nicht für die Verschleierung kontinuierlicher Standortupdates geeignet. Auch lassen sich Anwendungen wie z.B. die Online-Routenplanung, die für ihre korrekte Diensterbringung auf exakte Standortdaten angewiesen sind, nicht benutzen. Da die überwiegende Mehrheit der übrigen von Fawaz et al. untersuchten Anwendungen jedoch ohnehin nur sporadisch von den Nutzern verwendet wird, kann dieses Verfahren zuverlässig für die Verschleierung exakter Standorte eingesetzt werden [78].

Einen Ansatz, der sich für die Überwachung der Einhaltung von Privatsphärepräferenzen verwenden lässt, bieten Enck et al. mit *TaintDroid* [76]. Hierbei handelt es sich um eine Erweiterung des Android-Betriebssystems, die ein Monitoring des Informationsflusses innerhalb von Drittanbieter-Apps ermöglicht. Grundsätzliche Annahme ist, dass aus dem Internet heruntergeladene Anwendungen prinzipiell nicht vertrauenswürdig sind. Aus diesem Grund wird in Echtzeit kontrolliert, wie eine Anwendung die persönlichen Daten des Nutzers verarbeitet und ob diese das Gerät über die Netzwerkschnittstelle verlassen.

Die technische Grundlage für dieses Monitoring stellt die sog. *dynamic taint analysis* dar [261]. Durch Modifikation der virtuellen Maschine, die unter Android für die Ausführung der in Java programmierten Apps zuständig ist, markiert *TaintDroid* zu diesem Zweck auf Variablenlevel automatisch alle Daten, die eine App aus einer privatsphärerelevanten Quelle (*TaintSource*) erhält mit einem 32 Bit langem Tag. Die notwendigen *Privacy Hooks*, an denen die Daten markiert werden, befinden sich z.B. im `SensorManager` und `LocationManager` des Android Application Frameworks. Im weiteren Verlauf wird gemäß fester Regeln beobachtet, welchen Weg die Daten durch das Betriebssystem nehmen. Werden sie über eine sog. *TaintSink* wie die Netzwerkschnittstelle verschickt, wird ein Eintrag in einer Log-Datei erzeugt. Der CPU-Overhead, den *TaintDroid* erzeugt, wird mit 14% angegeben, was verglichen mit vergleichbaren Verfahren überaus erträglich ist [76]. Ein solches Verfahren kann somit einen wichtigen Baustein bei der Durchsetzung und Überwachung der Privatsphärepräferenzen eines Nutzers darstellen.

In diesem Kapitel wurden verwandte Arbeiten zu verschiedenen Aspekten der Verwaltung von Kontextinformationen vorgestellt und kategorisiert. Einige davon sind eher visionär im Themenfeld des Ubiquitous Computing angesiedelt, andere stellen pragmatische und direkt anwendbare Lösungsansätze für den Schutz der Privatsphäre in kontextbezogenen Diensten dar, die in ihrem Funktionsumfang aber jeweils deutlich begrenzt sind.

Keine der bestehenden Lösungen ist dazu in der Lage, alle Anforderungen zu erfüllen, die sich in Kapitel 3.2.2 aus der Vereinigung der in den bekannten Vorarbeiten behandelten Aspekte ergeben. Vor diesem Hintergrund wird in den folgenden Abschnitten ein neues Konzept entwickelt, das den vollständigen Anforderungskatalog abdeckt, um eine clientseitige, flexible, privatsphäreorientierte, kontext- und rezipientenabhängige Verwaltung der Kontextinformationen eines Nutzers zu ermöglichen.

3.4 Feingranulare, situationsabhängige Kontextverwaltung mit ALPACA

Im Folgenden wird ALPACA (von engl. *Allowing for fine-grained Privacy Adjustments in Context-aware Applications*[2]) vorgestellt, ein generischer, modellbasierter Ansatz für die Verwaltung von Kontextinformationen auf einem mobilen Endgerät. Der Benutzer wird dadurch in die Lage versetzt, situationsabhängige Regeln für die Weitergabe persönlicher Daten an verschiedene Anwendungen und andere Nutzer festlegen zu können.

Um die mit einer solchen Entscheidungsfreiheit verbundene Komplexität zu reduzieren, wird zunächst eine intuitive Abstraktion von Privatsphärebedürfnissen eingeführt, an der sich im weiteren Verlauf auch der grundsätzliche Aufbau von ALPACA orientiert. Dabei wird das Datenschutzempfinden eines Nutzers in Form unterschiedlicher Ebenen modelliert, was einen gewissen Automationsgrad bei der Verwaltung des Benutzerkontexts zulässt. Der Übergang von Kontextinformationen von einer Ebene auf eine andere, weniger geschützte Ebene wird dabei strikt überwacht. Dieser sehr restriktive Umgang mit persönlichen Informationen findet sich im weiteren Verlauf in allen Teilaspekten und Mechanismen des vorgestellten Systems wieder.

Im Anschluss wird ein privatsphärezentrisches Kontextmodell vorgestellt, auf dessen Basis sich ausdrucksstarke, kontextabhängige Freigabeentscheidungen definieren lassen. Zur effektiven Durchsetzung und Überwachung dieser Zugriffsregeln wird schließlich die vollständige Systemarchitektur präsentiert sowie die zentrale Komponente, der *Privacy Manager*, beschrieben.

[2]Die ausgeschriebene Variante des Namens wurde gegenüber der Originalveröffentlichung zugunsten einer treffenderen Umschreibung leicht modifiziert.

3.4.1 Ebenenmodell für die privatsphärebezogene Einordnung von Kontextinformationen

Die Ermöglichung flexibler, situations- und rezipientenabhängiger Freigaberegeln stellt zugleich ein mächtiges, aber auch überaus komplexes Werkzeug dar. Um die Entscheidungen hinsichtlich der Preisgabe bestimmter Typen und Granularitäten von Kontextinformationen für den Nutzer handhabbar zu gestalten, wird daher zunächst ein intuitives Modell zur Beschreibung unterschiedlicher Privatsphärebedürfnisse entwickelt. Eine solche Abstraktion dient der Kategorisierung von Kontextinformationen und kontextabhängigen Anwendungen, die in gewissem Umfang automatisiert erfolgen kann. Dieses Modell repräsentiert zudem die dieser Arbeit zugrundeliegende, restriktive Herangehensweise bei der Preisgabe persönlicher Informationen und liefert somit zugleich eine Architekturskizze für die technische Umsetzung.

Die Einteilung in Ebenen soll einen leicht verständlichen Kompromiss zwischen den individuellen Privatsphärebedürfnissen eines Nutzers bezüglich der Preisgabe unterschiedlicher Informationen und einer Reduzierung der dadurch entstehenden Komplexität darstellen. Die verschiedenen Ebenen sind so gewählt, dass sie, von der Akquise von Kontextinformationen bis hin zu deren Verschleierung sowie letztendlichen Freigabe und Verteilung an bestimmte Empfänger, alle relevanten Schritte abdecken. Unterschiedliche Anwendungen, Personen und Typen von Kontextinformationen lassen sich frei innerhalb dieser Ebenen gruppieren, um auf Basis dieser Einteilung einen gewissen Teil an Freigabeentscheidungen automatisiert zu treffen.

Zwischen den einzelnen Ebenen werden unterschiedlich restriktive Kontrollmechanismen etabliert, die den möglichen Übergang von privatsphärekritischen Informationen auf eine der weniger geschützten Modellebenen überwachen. Auf welcher Ebene eine bestimmte Anwendung einsortiert wurde, ist für diese nicht ersichtlich.

Es werden wie in [22] zwei verschiedene Rollen definiert, die im weiteren Verlauf dazu verwendet werden, zwischen Subjekt und Objekt von Kontextanfragen eindeutig zu differenzieren:

- Der *Kontextinhaber* (engl. Context Owner) ist der Besitzer eines mobilen Endgeräts sowie möglicher Peripheriegeräte, die den Nutzungskontext über Sensoren ermitteln. Die gesammelten Kontextinformationen des Kontextinhabers gilt es zu beschützen. Dieser legt Regeln fest, unter welchen Bedingungen und von wem auf eine Teilmenge seiner Informationen zugegriffen werden darf.

- Der *Kontextempfänger* (engl. Context Requester) möchte eine bestimmte Teilmenge der Kontextinformationen des Kontextinhabers in Erfahrung bringen. Dies kann eine beliebige Anwendung, ein online Dienst oder ein anderer Nutzer sein. Insbesondere im letzten Fall ist es möglich, dass die vom Kontextinhaber festgelegten Zugriffsregeln sich zum Teil auch auf den aktuellen Kontext des Kontextempfängers beziehen.

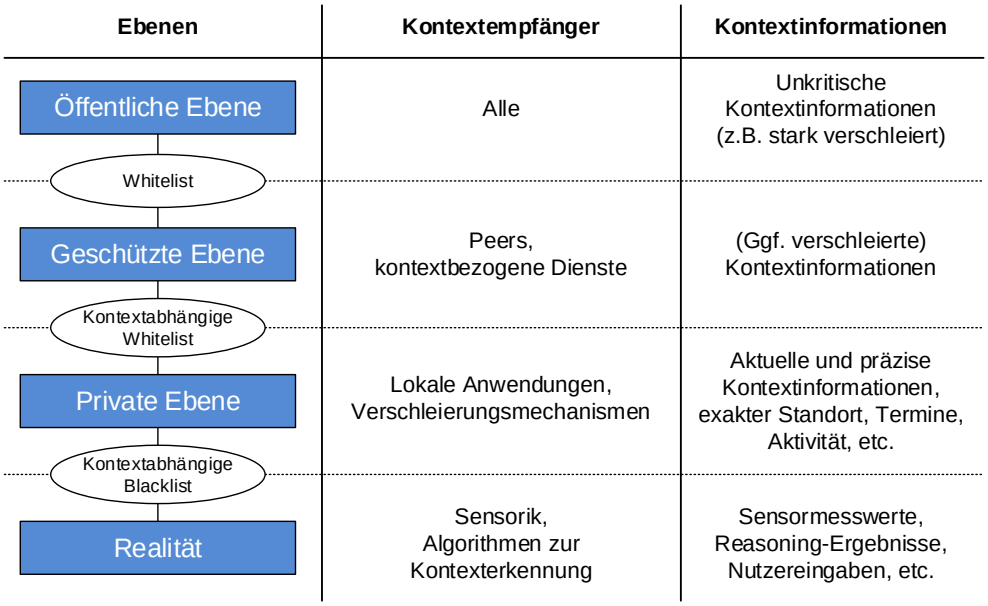


Abbildung 3.1: Die Privatsphärebedürfnisse eines Nutzers gegenüber unterschiedlichen Empfängern von Kontextinformationen lassen sich durch die Verwendung des Ebenenmodells intuitiv gruppieren.

Wie in Abb. 3.1 skizziert, werden vier intuitiv verständliche Ebenen vorgeschlagen, um eine einfache Kategorisierung von Informationen, Anwendungen und sonstiger Kontextempfänger zu ermöglichen: Die *Realität*, die *private Ebene*, die *geschützte Ebene* und die *öffentliche Ebene*.

Basierend auf dieser Vorgruppierung von Kontext und Kontextempfängern, sind innerhalb einzelner Ebenen automatisierte Entscheidungen hinsichtlich der Freigabe von Kontextinformationen möglich. Darüber hinaus können durch den in Kapitel 3.4.3 vorgestellten Mechanismus feingranulare Zugriffsregeln definiert werden, die – ähnlich einer Firewall an einer kritischen Stelle eines Netzwerks – die Durchlässigkeit der einzelnen Ebenen festlegen.

3.4.1.1 Erkennung der aktuellen Situation des Kontextinhabers

Die unterste der modellierten Ebenen steht stellvertretend für die *Realität*, also die tatsächliche Situation des Nutzers. Ihrer Natur entsprechend wird im Rahmen kontextabhängiger Anwendungen versucht, diese „Obermenge" aller Kontextinformationen durch verschiedenste Sensoren und Methoden der Kontext- und Situationserkennung so exakt wie möglich abzubilden und in eine semantisch angereicherte, maschinell verarbeitbare Form zu bringen. Die Rolle der Kontextempfänger nehmen auf dieser Ebene die Sensoren sowie die Verfahren zur Ermittlung von Kontextinformationen auf dem Endgerät des Nutzers ein.

Die reale Umgebung und Situation eines Nutzers umfassen dabei viel mehr Informationen, als durch die zur Verfügung stehenden Sensoren und Algorithmen in Erfahrung gebracht werden können. Welche dieser potentiellen Kontext-

quellen tatsächlich verwendet werden sollen, gilt es an dieser Stelle festzulegen. Denn während die Akquise und Auswertung von Sensordaten einen unerlässlichen Schritt bei der Nutzung kontextbezogener Dienste darstellt, muss ein umfassender Ansatz zur Verwaltung von Kontextinformationen auch die Deaktivierung dieser Komponenten vorsehen. Damit kann sich der Nutzer entweder situationsabhängig oder permanent dazu entscheiden, bestimmte Sensoren oder Erkennungsverfahren nicht zu aktivieren, wodurch entsprechende Kontextinformationen gar nicht erst erfasst werden.

Hierfür sind unterschiedliche Beweggründe vorstellbar. In bestimmten Situationen kann es dem Nutzer widerstreben, dass verschiedene Sensoren wie beispielsweise das Mikrofon überhaupt Daten aufnehmen. Vielleicht ist er sich auch im Vorhinein bereits sicher, dass er per se kein Interesse an der Nutzung bestimmter Typen von Kontextinformationen und entsprechender Anwendungen hat. Darüber hinaus kann die Deaktivierung von Sensoren und aufwändigen Analyseverfahren auch in Verbindung mit dem intelligenten Energiemanagement eines Endgeräts verbunden sein. Entsprechende Verfahren entscheiden – in der Regel selbst wiederum kontextabhängig – automatisch über die Abschaltung bestimmter Systemkomponenten [73, 175]. Hierbei würde beispielsweise die Nutzung der energieintensiven GPS-Ortung automatisch deaktiviert, um die verbleibende Laufzeit des Geräteakkus zu erhöhen.

3.4.1.2 Private Ebene der Kontextinformationen

Die nächst höhere Ebene dieser Privatsphäre-Abstraktion ist die *private Ebene*, welche alle von den Sensoren und Erkennungsverfahren ermittelten Informationen über den Nutzungskontext umfasst. Dies beinhaltet sowohl alle möglichen Subtypen von Kontextinformationen als auch alle verfügbaren Darstellungsformen und Granularitäten, die vom Endgerät ermittelt werden und dieses nicht unkontrolliert verlassen sollen. Im Rahmen des hier verfolgten, privatsphärezentrischen Ansatzes wird davon ausgegangen, dass sich ohne eine explizite Freigabe durch den Nutzer alle Kontextinformationen ausschließlich auf dieser Ebene wiederfinden.

Kontextempfänger, die ohne Gefahr für die Privatsphäre des Kontextinhabers auf dieser Ebene eingeordnet werden können, sind Anwendungen, die nicht mit fremden Systemen oder anderen Nutzern kommunizieren. Dies können beispielsweise Dienste sein, die ihr Verhalten oder das des gesamten Endgeräts dem aktuellen Nutzerkontext anpassen, wie das automatische Wiederanschalten des WLAN-Moduls bei Erreichen der eigenen Wohnung oder das Stummschalten bei Betreten eines Besprechungsraums. Insbesondere stellt auch der in Kapitel 3.4.4 vorgestellte Privacy Manager eine solche Anwendung dar, die auf die Gesamtheit der Informationen über die Situation des Nutzers zugreift, um die situationsabhängige Zugriffskontrolle effektiv durchsetzen zu können.

Ein Beispiel für eine ortsbezogene Anwendung aus dieser Kategorie ist eine Offline-Navigationsanwendung. Zwar benötigt sie mit dem jeweils aktuellen Standort sowie dem gewünschten Zielort zwingend persönliche Informationen

des Nutzers als Eingabeparameter, gibt diese aber nicht nach außen weiter.

Aus Sicht der Privatsphäre stellen solche Anwendungen den Optimalfall dar – mit dem Nachteil, dass sie in ihrem Funktionsumfang meist begrenzt sind und das eigentliche Potential kontextbezogener Anwendungen nicht ausschöpfen.

Darüber hinaus steht es dem Kontextinhaber natürlich frei, beliebige Kontextempfänger, denen er vertraut, ebenfalls auf dieser Ebene zu platzieren. Sofern es jedoch nicht durch explizite Freigaberegeln des Nutzers erlaubt ist, muss das System dafür sorgen, dass keine Informationen diese private Ebene verlassen. Kontextempfänger, die innerhalb dieser Ebene eingestuft werden, haben hingegen uneingeschränkten Zugriff auf die gesammelten Informationen des Kontextinhabers in ihrer genauesten und aktuellsten Form.

3.4.1.3 Geschützte Kontextinformationen

Kontextabhängige Anwendungen, die nicht zu der soeben beschriebenen privaten Ebene gehören, dürfen nicht mit privaten Kontextinformationen versorgt werden. Um jedoch auch solche Dienste nutzen zu können oder um bestimmte Informationen mit anderen Nutzern zu teilen, muss der Kontextinhaber die Möglichkeit haben, geeignete Freigaberegeln aufzustellen. Um zu ermöglichen, dass die Freigabe einzelner Informationen den Privatsphärebedürfnissen des Nutzers entsprechend nicht nur je nach Empfänger, sondern darüber hinaus auch je nach Nutzungssituation unterschiedlich ausfallen kann, müssen diese Regeln selbst kontextabhängig definiert werden können.

Zum Schutz der Privatsphäre gegenüber unterschiedlichen Kontextempfänger lautet die Frage, wie sehr der Kontextinhaber einem bestimmten Empfänger vertraut und mit welchen Informationen dieser in der aktuellen Situation ausgestattet werden soll. Da es unwahrscheinlich ist, dass der Kontextinhaber jeder Anwendung bzw. jedem Dienstanbieter oder Nutzer stets dasselbe Vertrauen entgegenbringt, lassen sich bei der Zuordnung von Informationen zu Empfängern keine pauschalen Entscheidungen treffen.

Artverwandte Problemstellungen sind das Risiko- und Trustmanagement in IT-Systemen. In [134] und [167] wird das Vertrauen (engl. *Trust*) einer Entität in eine andere formal auf einen Wert zwischen 0 (vollständiges Misstrauen) und 1 (vollständiges Vertrauen) abgebildet. Durch Beobachtung, ob das Ergebnis der von der zweiten Entität ausgeführten Aktion dem Initiator einen Vorteil oder einen Nachteil beschert, kann der Wert sukzessive angepasst werden. Münzt man diese Definition von Trust auf das hier vorliegende Szenario um, lässt sich das Vertrauen in einen bestimmten Kontextempfänger als der Glaube des Kontextinhabers definieren, dass die Weiterverarbeitung einer Informationen durch diesen Empfänger positive oder negative Auswirkungen für den Inhaber haben wird.

Eine solche Herangehensweise kann in vielen Szenarien sinnvoll zur probabilistischen Risikoabschätzung eingesetzt werden. Es ist aber davon auszugehen, dass ein privater Anwender einen derartig hohen Formalisierungsgrad vermeiden möchte. Stattdessen wird er, wie in [257] beschrieben, wohl eher gefühls-

basiert festlegen, welchem Kontextempfänger er welchen Grad an persönlichen Informationen zukommen lassen möchte. Zudem ist z.B. der negative Effekt, der durch die dauerhafte Aufzeichnung und Auswertung von übermittelten Informationen durch einen Kontextempfänger entstehen kann, für den Nutzer ohnehin nicht direkt ersichtlich oder messbar. Im Folgenden wird daher auf eine solche formale Interpretation von Vertrauen verzichtet.

Während es einem Privatsphäre-bewussten Nutzer widerstrebt, seinen aktuellen Kontext immer und mit jedem zu teilen, verspricht er sich davon in einigen Fällen dennoch einen Mehrwert. So ist er u.U. bereit, während der Arbeitszeit seinen aktuellen Aufenthaltsort mit seinen Kollegen oder in seiner Freizeit mit engen Freunden und Familienmitgliedern zu teilen. Letzteres aber auch nur, wenn er sich nicht gerade in einer z.B. als peinlich oder privat empfundenen Situation befindet. Bei einem Unfall hingegen möchte er, dass mit dem Rettungsdienst auch für ihn völlig Fremde seine exakte Position kennen.

Es handelt sich hierbei also um schützenswerte Kontextinformationen, die nicht immer, nicht stets im selben Detailgrad und nicht jedem Kontextempfänger im selben Maße zugänglich sein sollen. Hierfür müssen individuelle Freigabeentscheidungen definiert werden können.

3.4.1.4 Öffentliche Kontextinformationen

Kontextempfänger, für die in der aktuellen Nutzungssituation keine expliziten Freigaberegeln existieren und die auch nicht auf der privaten Ebene eingruppiert sind, werden analog zu [159] automatisch auf der öffentliche Ebene platziert. Dadurch wird die Anforderung nach der Vollständigkeit des Regelwerks umgesetzt, denn Kontextanfragen von Nutzern und Diensten, die nicht explizit kategorisiert sind, können somit automatisch mit den für diese Ebene vorgesehenen Kontextinformationen versorgt werden. Aus Sicht der Privatsphäre sollten hier also ausschließlich Werte zurückgegeben werden, die entweder als unbedenklich eingeschätzt werden, stark verzerrt oder sogar unwahr sind.

Durch die Einführung der öffentlichen Ebene ergibt sich aus Nutzersicht der Vorteil, dass nicht explizit für jeden Kontextempfänger eigene Freigaberegeln erstellt werden müssen und dennoch alle Anfragen verarbeitet werden können. So können Kontextanfragen von Anwendungen, die z.B. für die Ausspielung ortsbezogener Werbung seinen aktuellen Standort abfragen, automatisch mit einer vom Nutzer festgelegten Standardantwort für diesen Typ von Kontextinformation beantwortet werden, ohne dass dies einer Interaktion seitens des Nutzers bedarf. Wie die auf der öffentlichen Ebene verfügbaren Kontextinformationen im Detail auszusehen haben, sollte individuell einstellbar sein.

Die Zuordnung von Kontextempfängern und -informationen auf die öffentliche oder die geschützte Ebene unterliegt einer Dynamik, die abhängig von den aufgestellten Regeln und der Änderungsrate des Nutzungskontexts ist. Hierfür muss ein Mechanismus gefunden werden, der die Erstellung ausdrucksstarker

und flexibler Freigaberegeln ermöglicht, um der Kontextabhängigkeit der Freigabeentscheidungen eines Nutzers gerecht zu werden.

3.4.1.5 Auswahl geeigneter Kontrollmechanismen pro Ebene

Für den Übergang von Informationen auf eine bestimmte Ebene des Modells eignen sich jeweils unterschiedlich restriktive Kontrollmechanismen. Hierbei sollen Benutzerfreundlichkeit und Verfügbarkeit kontextabhängiger Dienste maximiert werden, sofern dadurch die volle Einhaltung der Privatsphärebedürfnisse des Kontextinhabers nicht beeinträchtigt wird.

Der Kontrollmechanismus, der die Akquise von Kontextinformationen – also die Ermittlung des Nutzungskontexts aus der realen Situation des Kontextinhabers – überwacht, wird daher als *Blacklist-Verfahren* implementiert. Diese Wahl sorgt dafür, dass auf der privaten Ebene standardmäßig so viele qualitativ hochwertige Kontextinformationen wie möglich zur Verfügung stehen. Nachdem die Blacklist nur den Übergang von Informationen auf die private Ebene kontrolliert, ist hierbei kein Verlust an Privatsphäre möglich.

Der üblichen Funktionsweise eines blacklistbasierten Ansatzes entsprechend, werden dabei nur solche Sensormesswerte und Kontextinformationen ausgeschlossen, die vom Nutzer explizit auf diese Liste gesetzt wurden. Dabei muss auch diese Blacklist als kontextabhängiger Mechanismus implementiert werden, da wie in 3.4.1.2 beschrieben auch das Bedürfnis, bestimmte Sensoren oder andere Mittel der Kontextermittlung an- oder abzuschalten, abhängig von der aktuellen Situation des Nutzers sein kann.

Alle Anwendungen, die sich nicht auf der privaten Ebene befinden, werden von ALPACA automatisch auf die öffentliche Ebene gesetzt. Dort haben sie keinen Zugriff auf qualitiativ hochwertige Kontextinformationen, sondern werden mit unkritischen Informationen versorgt. Sie haben jedoch die Möglichkeit, auf die geschützte Modellebene zu gelangen, indem der Kontextinhaber entsprechende Freigaberegeln definiert, die den Übergang von Kontextinformationen von der privaten Ebene auf die geschützte Ebene überwachen.

Im Gegensatz zur oben beschriebenen Kontextakquise wird dieser Übergang jedoch restriktiv durch einen *Whitelist-Mechanismus* geschützt. Bei dem whitelistbasierten Ansatz muss jede Kontextinformationen explizit für bestimmte Kontextempfänger freigegeben werden, was den Schutz der Privatsphäre über eine unkomplizierte Out-of-the-Box-Funktionalität von Anwendungen stellt. Es gelten hierbei dieselben Anforderungen hinsichtlich der Kontextabhängigkeit dieser Regeln wie zuvor. Hinzu kommt, dass durch das unterschiedliche Vertrauen, das der Kontextinhaber in verschiedene Kontextempfänger haben kann, nun auch eine Rezipientenabhängigkeit bei der Definition und Auswertung dieser Regeln berücksichtigt werden muss.

Anhand dieser Konzeptualisierung von Privatsphärebedürfnissen erklärt sich auch die restrikte Herangehensweise bei der Kontextverwaltung mit ALPACA. Im Folgenden wird die technische Umsetzung der dafür notwendigen Teila-

spekte vorgestellt. Das nächste Kapitel beschreibt das privatsphärezentrische CoRe-Modell, auf dessen Basis im Anschluss ein Mechanismus für die Umsetzung der kontextabhängigen Freigaberegeln konzipiert wird.

3.4.2 Privatsphärezentrische Kontextmodellierung

ALPACA verfolgt bei der Bereitstellung von Kontextinformationen einen primär auf die Privatsphärebedürfnisse eines Nutzers ausgerichteten Ansatz. Aus diesem Grund wird der Schutz der Privatsphäre wie in [174] gefordert als übergreifendes Thema betrachtet. Entsprechende Konzepte und Vorkehrungen werden bei allen Teilschritten berücksichtigt, die für die Nutzung kontextabhängiger Dienste nötig sind.

Bereits bei der Ermittlung und Modellierung von Kontextinformationen muss daher sichergestellt sein, dass alle hierfür relevanten Informationen mit erhoben und angemessen integriert werden. Damit kann sichergestellt werden, dass das Kontextmodell alle Informationen beinhaltet, die später zur Definition und Auswertung von Freigabeentscheidungen bezüglich einzelner Ausschnitte dieser Daten benötigt werden. Darüber hinaus ermöglicht die enge Verzahnung von Kontextmodell und Zugangskontrolle, die einer formalen Modellierung inherenten Möglichkeiten zur Validierung eines konkreten Modellzustands auch zur Einhaltung der privatsphärebezogenen Systemanforderungen einzusetzen.

In Kapitel 2.2 wurden unterschiedliche Konzepte zum Schutz der Privatsphäre in kontextabhängigen Diensten vorgestellt. Bei einigen dieser Verfahren handelt es sich um Mechanismen, die sich unabhängig von Semantik und Typ allgemeingültig für die Verwaltung von Kontextinformationen anwenden lassen. Insbesondere stellt die Bereitstellung einer situations- und rezipientenabhängigen Zugangskontrolle einen solchen allgemeinen Schutzmechanismus dar. Allein damit kann bereits effektiv über die Preisgabe einer Information an einen bestimmten Empfänger entschieden werden. Um die Erstellung und Auswertung situationsabhängiger Freigaberegeln zu unterstützen, muss das Modell alle dafür notwendigen Informationen abbilden.

Darüber hinaus gibt es unterschiedliche Ansätze zur Kontextverschleierung, um die Privatsphäre z.B. durch die Verzerrung bestimmter Details zu schützen. Somit ist nicht nur die Entscheidung zu treffen, ob man Informationen an einen bestimmten Kontextempfänger preisgibt, sondern zudem auch wie diese auszusehen haben. Einige von diesen Verschleierungsmechanismen sind wiederum generisch einsetzbar, wie z.B. die Anpassung der Aktualität oder der Angaben zur Kontextqualität. Im Fall von hierarchisch modellierbaren Kontexttypen lassen sich durch Generalisierung selbst Auflösung und Detailgrad der preisgegebenen Informationen allgemeingültig anpassen, indem auf die Darstellung einer entsprechenden Hierarchieebene ausgewichen wird [255].

Oft reichen diese einfachen, generisch anwendbaren Verfahren jedoch nicht aus, um effektiv gegen verschiedene Arten eines potentiellen Angreifers zu schützen. Oder die einfache Form der Verschleierung macht den Dienst mehr

oder weniger unbrauchbar, da auf Basis der verschleierten Kontextinformationen keine qualitativ hochwertige Dienstnutzung mehr möglich ist. Wie in Kapitel 2.2.3 gezeigt, ist in der Literatur eine laufend wachsende Zahl an spezialisierten Verschleierungstechniken zu finden, die sich meist mit exakt einem Informationstyp beschäftigen und sich auf die Verhinderung spezifischer Angriffsszenarien fokussieren. Auch diese spezialisierten Formen der Kontextverschleierung müssen daher im Kontextmodell abbildbar sein und bei der Herausgabe von Informationen als gleichwertige Alternativen zur Verfügung stehen. Wie dies im Rahmen der Modellierung von Kontextinformationen bei ALPACA umgesetzt ist, wird im Folgenden erläutert.

3.4.2.1 Modellierung alternativer Kontextrepräsentationen

Um die vielfältigen Möglichkeiten zur Kontextverschleierung abzubilden, muss ein privatsphärezentrisches Kontextmodell neben der eigentlichen Kontextinformationen eine Reihe zusätzlicher Daten bereithalten. Diese werden entweder als Metadaten bei der Ermittlung der echten Kontextinformationen erhoben, dienen als Parameter für die Anwendung von Schutzmechanismen oder sind das Ergebnis der Verschleierung. Der grundlegende Ansatz, den das für ALPACA entwickelte *Context Representations*-Modell (CoRe) hierfür verfolgt, ist es, zu jeder Kategorie von Kontext eine dynamische Anzahl an alternativen Darstellungsformen samt den damit verbundenen Metainformationen zu modellieren. Je nach aktueller Situation und dem Maß an Vertrauen des Kontextinhabers in einen bestimmten Kontextempfänger, können letzterem dadurch jeweils maßgeschneiderte Informationen zurückgeliefert werden.

Wie auch in anderen Ansätzen [28, 254, 108, 244, 255, 199, 146] werden die Kontextinformationen eines Nutzers in CoRe wenn möglich hierarchisch modelliert. Solche Hierarchien lassen sich in einer baumartigen Struktur darstellen, deren Wurzel die Oberklasse aller Kontextinformationen darstellt. Eine exemplarische Darstellung einer solchen hierarchischen Modellierung ist in Abb. 3.2 zu sehen. Auf der zweiten Hierarchieebene spalten sich diese Informationen in die unterschiedlichen Subtypen von Kontext auf.

Für bestimmte Typen von Kontextinformationen existieren auf dieser Ebene semantisch verschiedene Darstellungsformen. So kann der aktuelle Aufenthaltsort eines Nutzers in Form von Koordinaten angegeben werden, als geographischer Bezeichner wie „München" oder aber als symbolischer Ort wie z.B. „Büro". Je weiter man sich auf entlang einer solchen Hierarchie nach unten bewegt, desto detaillierter werden die Kontextbeschreibungen.

Üblicherweise wird für jeden Typ von Kontextinformation dabei mindestens der aktuell gültige Wert im Modell abgelegt. Zusätzlich wird für bestimmte Typen kontextbezogener Dienste auch auf frühere Kontextinformationen zugegriffen [151], sodass diese ebenfalls im Modell vorgehalten werden sollten. Aus Sicht der Privatsphäre können anhand dieser historischen Daten generische Verschleierungsmechanismen wie die Anpassung der Aktualität von Kontextinformationen umgesetzt werden.

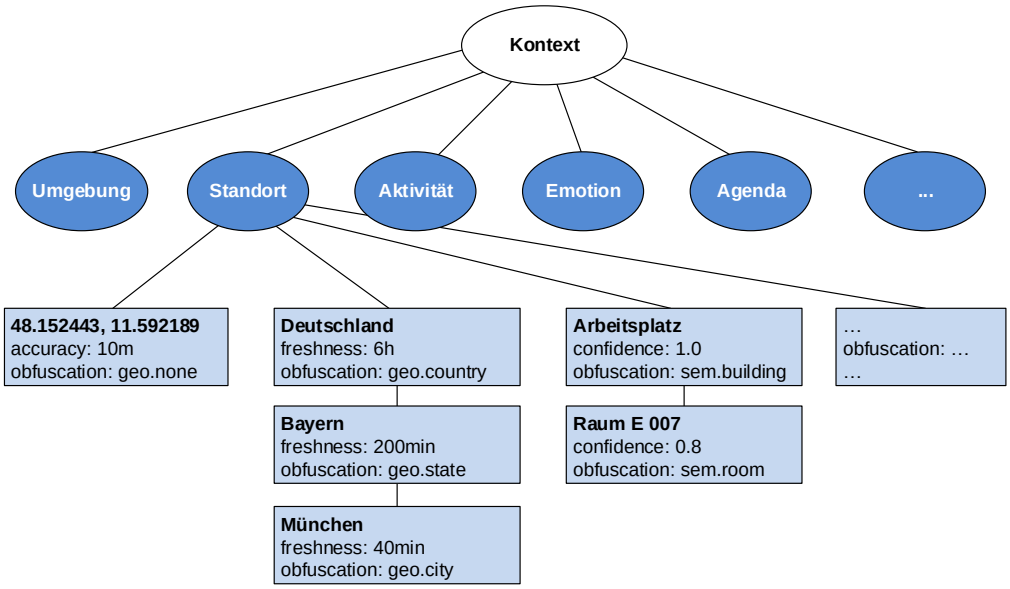


Abbildung 3.2: Knoten auf der zweiten Ebene eines hierarchischen Kontextmodells stellen verschiedene Subytpen von Kontext dar, die auf unteren Ebenen immer detaillierter beschrieben sind.

Das neu vorgestellte CoRe-Modell ist daher darauf ausgelegt, für jede Kontextkategorie parallel eine beliebige Anzahl an alternativen Darstellungen verwalten. Eine einzelne solche Information wird im Folgenden *Repräsentation* (engl. Representation) genannt. Im Gegensatz zu anderen Kontextmodellen wie z.B. MUSIC [199] wird der Begriff der Repräsentation hier also nicht als Beschreibung der Datenkodierung (z.B. XML oder JSON) betrachtet. Stattdessen beinhalten unterschiedliche Repräsentationen einer Information inhaltlich verschiedene Darstellungen des jeweiligen Kontexttyps.

Die verschiedenen Repräsentationen einer Kontextinformation können sich in ihrer Semantik, ihrem Alter und – insbesondere vor dem Hintergrund der Privatsphäre des Nutzers – in ihrer Genauigkeit oder ihrem Wahrheitsgehalt unterscheiden. Um dem Nutzer beispielsweise zu ermöglichen, verschiedenen Kontextempfängern unterschiedliche Informationen über seinen aktuellen Standort mitzuteilen, kann eine Repräsentation beispielsweise die exakten GPS-Koordinaten beinhalten, eine zweite seinen Aufenthaltsort vor 15 Minuten und eine dritte die grobe Angabe, in welcher Stadt er sich aufhält. Hierbei liegt es jeweils an der optional eingesetzten Verschleierungstechnik, ob eine solche Angabe der Wahrheit entspricht oder nicht.

Unterhalb eines einzigen Knotens auf der zweiten Hierarchieebene, wie z.B. der Ortskontext des Kontextinhabers, befinden sich in CoRe somit eine Vielzahl unterschiedlichster Repräsentationen dieses Kontexttyps für verschiedene Adressaten. Wie diese verschiedenen Repräsentationen erzeugt werden, wird an dieser Stelle bewusst offen gelassen. Zwar sind wie beschrieben einige allgemeingültig anwendbare Verschleierungsmechanismen in ALPACA integriert,

es wird jedoch nicht wie in [45] oder [255] versucht, alle möglichen Verschleierungsergebnisse durch die Angabe umfassender Generalisierungstaxonomien vorzudefinieren. Erstens lässt sich dies nicht einheitlich für alle Kategorien von Kontext zur Verschleierung einsetzen. Zweitens wäre dies in vielen Fällen mit großem manuellen Aufwand verbunden und drittens gibt es neben der Generalisierung, wie in Kapitel 2.2 gezeigt, noch eine Vielzahl an anderen Verschleierungstechniken, die ebenfalls berücksichtigt werden sollen.

Stattdessen wird auf eine einfache Erweiterbarkeit des CoRe-Modells sowie der ALPACA-Systemarchitektur gesetzt. Im weiteren Verlauf wird daher u.a. gezeigt, wie neue Typen von Kontextinformationen oder spezialisierte Verschleierungsmechanismen integriert werden können. Aus Sicht der Privatsphäre ist dabei erstrebenswert, dass solche Verschleierungsmechanismen gefunden werden, die sich direkt auf dem Endgerät des Kontextinhabers durchführen lassen ohne exakte Informationen an Dritte weiterzugeben. Wie dies konkret aussehen kann wird in Kapitel 4 anhand zweier konkreter Beispiele für den Schutz der Privatsphäre in verschiedenen ortsbezogenen Diensten gezeigt.

3.4.2.2 Umsetzung des CoRe-Modells mit OWL

Den Empfehlungen aus Übersichtsarbeiten wie [28] und [226] folgend, wird die Beschreibung des CoRe-Modells als Ontologie realisiert. Die Grundlagen von Ontologien und deren Einsatz für die Kontextmodellierung wurden bereits in Abschnitt 3.3.1.2 beschrieben. Von den Vorteilen, die der damit verbundene hohe Formalisierungsgrad bietet, wird im weiteren Verlauf insbesondere die Möglichkeit zur automatischen Validierung des Modellzustands eingesetzt. So lassen sich die formal modellierten Kontextinformationen mit Hilfe standardmäßiger Reasoner-Implementierungen direkt für die Umsetzung und Konsistenzprüfung der situationsabhängigen Freigaberegeln verwenden.

Konkret orientiert sich die im Folgenden vorgestellte Umsetzung des Modells an den Vorgaben und Möglichkeiten der *Web Ontology Language 2* (OWL 2) [240], die dem Themenumfeld des Semantic Web [27] entstammt. Die Erstellung und Validierung des Modells erfolgt mit Hilfe des Ontologie-Editors *Protégé* [184] sowie des integrierten OWL-Reasoners *HermiT* [180]. Auf Auszüge der XML-basierten Modellbeschreibung wird aus Gründen der Lesbarkeit des Textes weitestgehend verzichtet. Stattdessen werden relevante Klassen und Eigenschaften des Modells anhand grafischer Darstellungen erklärt.

3.4.2.2.1 Klassen und Struktur des CoRe-Modells Die wichtigsten Begriffe des CoRe-Modells sind in Abb. 3.3 dargestellt. Dabei handelt es sich um die Oberklassen *Context*, *Representation*, *Grant* und *Entity*. Für die ersten drei dieser Klassen existieren pro Kontexttyp eigene Unterklassen. So wird die Aktivität des Kontextinhabers z.B. in Form der Unterklasse *ActivityContext* kategorisiert, durch Instanzen der Klasse *ActivityRepresentation* mit ihrem aktuell gültigen Wert im Modell abgebildet sowie mit Hilfe von Individuen, die der entsprechenden Unterklasse *ActivityGrant* angehören, verwaltet.

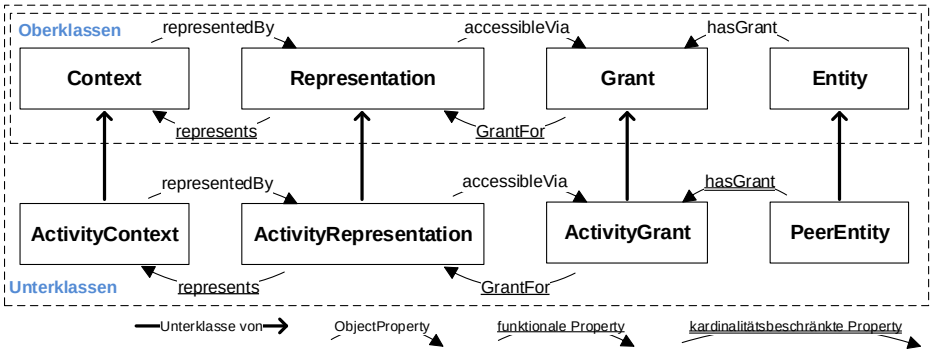


Abbildung 3.3: Die konzeptuellen Eckpfeiler des CoRe-Modells: *Context*, *Representation*, *Grant* und *Entity* sowie jeweils eine exemplarische Unterklasse für den Kontexttyp „Aktivität".

Der erlaubte Wertebereich der entsprechenden Relationen *represents* und *GrantFor* sind durch die Verwendung der *allValuesFrom*-Restriktion innerhalb der entsprechenden Unterklassen jeweils lokal durch die Beschreibung einer anonymen Oberklasse auf den entsprechenden Kontexttyp limitiert:

```
<owl:Class rdf:ID="ActivityRepresentation">
  <rdfs:subClassOf rdf:resource="#Representation"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#represents" />
      <owl:allValuesFrom rdf:resource="#ActivityContext" />
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```

Vor dem Hintergrund des in Kapitel 3.3.1.1 beschriebenen *Open World*-Prinzips einer Ontologie sind diese Klassen sowie ihre Unterklassen zudem jeweils explizit als paarweise disjunkt gekennzeichnet. Ansonsten würde der Reasoner davon ausgehen, dass eine Instanz der Klasse *ActivityGrant* z.B. auch ein *LocationGrant* sein kann, usw.

Entity stellt die Oberklasse für alle verschiedenen Typen von Kontextempfängern dar, also unterschiedliche kontextbezogene Anwendungen (Instanzen der Unterklasse *ServiceEntity*) oder andere Nutzer (*PeerEntity*). Diese können selbst wiederum Kontextinformationen besitzen, die im Modell gespeichert werden. Um die Einhaltung der zeitlichen Konsistenz bei einer eingehenden Kontextanfrage überprüfen zu können, werden im Modell zudem zu jeder Entität pro Kontexttyp deren zuletzt bekannten Informationen sowie die dieser Entität bereits zurückgegebenen Antworten auf Kontextanfragen mit der Relation *hasSeen* gespeichert. Zur Identifikation und Unterscheidung verschiedener Entitäten besitzen diese eine Relation *hasID*, die auf ein Stringliteral verweist. Darüber hinaus können Nutzer verschiedenen Gruppen hinzugefügt werden, was wie in [22] die Erstellung von Freigaberegeln z.B. anhand der sozialen Beziehung des Kontextinhabers zu einem Kontextempfänger ermöglicht.

Instanzen der Unterklasse *Representation* modellieren die zuvor beschrie-

benen unterschiedlichen Darstellungsalternativen der jeweiligen Kontextkategorie. Eine bestimmte Kontextklasse kann zur Ermöglichung feingranularer Freigabeentscheidungen dabei wie erwähnt mehrere solcher Repräsentationen parallel besitzen. Im Modell wird dies durch die invers-funktionale Eigenschaft *representedBy* ausgedrückt, die somit eine 1:n-Zuordnung von Kontextkategorie und entsprechenden Repräsentationen zulässt.

Diese Struktur ähnelt dem ASC-Modell von Strang et al. [227], das sich ebenfalls mit unterschiedlichen Darstellungsformen ein und derselben Information beschäftigt. Fokus und Umsetzung folgen dabei allerdings einer gänzlich anderen Zielsetzung. Dort soll die Interoperabilität von Diensten erreicht werden, was durch die automatische Überführung ausgetauschter Informationen von einer Darstellungsart in eine andere ermöglicht wird. Zu diesem Zweck definiert deren Modell Regeln, wie zwischen verschiedenen, aber inhaltlich äquivalenten Skalen gewechselt werden kann. So wird die Beziehung verschiedener Skalen zueinander, wie z.B. eine Kilometer- und eine Meilenskala, modelliert sowie entsprechende Transformationsregeln hinterlegt.

Das hier vorgestellte CoRe-Modell modelliert zwar auch verschiedene Darstellungsformen eines Kontexttyps, tut dies aber nicht zum Zwecke der gegenseitigen Überführbarkeit, sondern – ganz im Gegenteil – um ggf. möglichst verschiedenartige Repräsentation zur Verfügung zu haben, die bzgl. ihrer Semantik nicht zwangsläufig in einer Äquivalenzrelation stehen. Zum Schutz der Privatsphäre des Kontextinhabers existieren oft gerade keine solchen Transformationsregeln, mit denen sich z.B. eine verschleierte Information wieder eindeutig auf ihren Ursprungswert zurückführen lässt.

3.4.2.2.2 Freigabe von Repräsentationen über Grants Um die Erstellung individueller Privatsphärepräferenzen auf Basis des CoRe-Modells zu ermöglichen, kann eine Instanz einer bestimmten Unterklasse von *Representation* mit Hilfe einer Unterklasse von *Grant* für verschiedene Kontextempfänger zugänglich gemacht werden. Um die aufgestellte Forderung nach der Konsistenz und Eindeutigkeit der Freigaberegeln zu gewährleisten, muss sichergestellt werden, dass ein Kontextempfänger zu jedem Zeitpunkt maximal Zugang zu einer Repräsentation eines bestimmten Typs hat. Andernfalls kann nicht entschieden werden, welche der verschiedenen Darstellungsformen als Antwort auf die Kontextanfrage zurückgegeben werden soll. Dies würde zu inkonsistenten Antworten führen und könnte z.B. verraten, dass der Kontextinhaber nicht ehrlich antwortet. Die Vollständigkeit der Regeln ist auch gewährleistet, wenn kein *Grant* besteht. In diesem Fall wird die entsprechende Information der öffentlichen Ebene zurückgegeben.

Im Modell wird die Verfügbarkeit einer Kontextinformationen für einen Kontextempfänger durch die Existenz einer entsprechenden *hasGrant*-Eigenschaft abgebildet, siehe Abb. 3.3. Diese nicht-funktionale Relation ist durch die Angabe einer maximalen Kardinalität von 1 je *Grant*-Unterklasse eingeschränkt:

```
1 <owl:Class rdf:about="Entity">
```

```
2     <rdfs:subClassOf>
3       <owl:Restriction>
4         <owl:onProperty rdf:resource="#hasGrant"/>
5         <owl:maxQualifiedCardinality ...>1</owl:maxQualifiedCardinality>
6         <owl:onClass rdf:resource="#ActivityGrant"/>
7       </owl:Restriction>
8     </rdfs:subClassOf>
9     <rdfs:subClassOf>
10      <owl:Restriction>
11        <owl:onProperty rdf:resource="#hasGrant"/>
12        <owl:maxQualifiedCardinality ...>1</owl:maxQualifiedCardinality>
13        <owl:onClass rdf:resource="#LocationGrant"/>
14      </owl:Restriction>
15    </rdfs:subClassOf>
16    ...
17  </owl:Class>
```

Somit kann jede Entität belebig viele *Grant*-Instanzen gleichzeitig besitzen, jedoch immer nur eine pro Kontexttyp. Der Nutzer kann dadurch einem Kontextempfänger Zugang zu verschiedenen Kontextinformationen erlauben. Durch diese Restriktionen kann bei der Validierung des Modellzustands sichergestellt werden, dass jede *Entity* tatsächlich Zugriff auf maximal eine Repräsentation eines Kontexttyps hat. Wie in Kapitel 3.4.3.3 gezeigt wird, kann dies obendrein für die dynamische Erkennung widersprüchlicher Freigaberegeln verwendet werden.

Der Kontextempfänger kann eine beliebige Instanz von *Entity* sein, die auf eine bestimmte Information des Kontextinhabers zugreifen möchte. Die Identifizierung des Kontextempfängers, der sowohl der Server eines kontextabhängigen Dienstes oder ein anderer Nutzer sein kann, ist dabei entweder direkt über eine die Angabe eines Identifikators, der Zugehörigkeit zu einer gewissen Gruppe oder durch die Beschreibung dessen Kontexts möglich. Zusätzlich können diese Varianten frei miteinander kombiniert werden, um Kontextempfänger auf Basis der sozialen Beziehung und deren eigenen Kontexts z.B. als „*Kollege, der sich aktuell im Bürogebäude aufhält*" genauer zu charakterisieren.

Die *GrantFor*-Relation sagt aus, für welche Repräsentation eines Kontexttyps ein Grantobjekt gültig ist und ist daher ebenfalls als funktional gekennzeichnet. Die dazu inverse Eigenschaft *accessibleVia* unterliegt jedoch nicht dieser Einschränkung, d.h., dass eine Instanz von *Representation* durch verschiedene *Grant*-Objekte problemlos verschiedenen Kontextempfängern zugänglich gemacht werden kann.

Erneut ist bei der Erzeugung neuer Instanzen der Unterklassen von *Representation, Grant* und *Entity* aufgrund der *Open World Assumption* darauf zu achten, diese jeweils explizit als unterschiedliche Individuen zu kennzeichen. Ansonsten würde das Reasoning auf dem aktualisierten Modellzustands u.U. nicht zur Erkennung eines ungültigen Zustands führen, den z.B. der Besitz mehrerer Grants für denselben Kontexttyp durch eine Entität hervorrufen sollte. Stattdessen würde die logische Schlussfolgerung gezogen, dass es sich z.B. bei zwei verschiedenen Instanzen der Klasse *Grant* um dasselbe Individuum handeln muss, der in der Folge Zugriff auf unterschiedliche *Representation*-Instanzen ermöglichen würde. Die Tatsache, dass es sich bei allen Instanzen

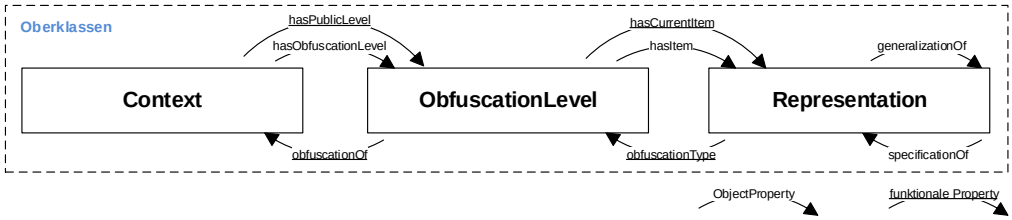


Abbildung 3.4: Jede Instanz der Klasse *Representation* stellt einen Kontexttyp gemäß eines bestimmten Schemas dar, das durch die Angabe der entsprechenden *ObfuscationLevel*-Klasse definiert ist.

einer bestimmten Klasse um verschiedene Individuen handelt, lässt sich in OWL z.B. effizient mit dem *owl:AllDifferent*-Axiom realisieren.

3.4.2.2.3 Klassifizierung von Repräsentationen anhand ihres Verschleierungstyps Um die privatsphärerelevanten Eigenschaften einer bestimmten *Representation*-Instanz einordnen zu können, verweisen diese mit der funktionalen Eigenschaft *obfuscationType* auf ihren jeweiligen Verschleierungstyp, d.h. auf einen bestimmten *ObfuscationLevel*. Dieser charakterisiert die jeweilige Repräsentation hinsichtlich ihrer Genauigkeit, Richtigkeit oder Aktualität und gibt an, durch welchen Verschleierungsmechanismus diese Repräsentation erzeugt wurde und welchem Schutzziel sie dient. Anhand dieser Beschreibung kann bei der Anfrage von Informationen durch einen bestimmten Kontextempfänger eindeutig bestimmt werden, welche Repräsentation den Anfordungen des Kontextinhabers entsprechend zurückgegeben werden soll.

Die unverfälschten Kontextrepräsentationen speichern die echten Kontextinformationen des Nutzers in der höchsten verfügbaren Aktualität, Qualität und Detailtiefe. Zur korrekten Einordnung der dieser Repräsentationen existiert standardmäßig eine entsprechende *ObfuscationNone*-Instanz. Diese wahren Informationen werden wie erläutert den Kontextempfängern auf der privaten Ebene zur Verfügung gestellt und zur Auswertung der Freigaberegeln verwendet. Zusätzlich wird auch jeder Hardware-Sensor über einen eigenen *ObfuscationLevel* modelliert. Diese Instanzen können sowohl für die Umsetzung des situationsabhängigen Blacklist-Mechanismus verwendet werden als auch zur direkten Freigabe eines Sensors an eine Anwendung. Dies wird in Kapitel 3.4.4.1 näher beschrieben.

Die wichtigsten Relationen zwischen Kontexttyp, Verschleierungstyp und Repräsentationen sind in Abb. 3.4 zu sehen. Jeder Kontexttyp kann eine beliebige Zahl an verschiedenen Verschleierungsmechanismen besitzen. Eine Instanz einer Unterklasse von *Representation* hingegen gehört exakt einem *ObfuscationLevel* an, genau wie ein Verschleierungstyp sich eindeutig zu einem Kontexttyp zuordnen lässt. Natürlich kann dabei z.B. ein Individuum, das der Klasse *LocationRepresentation* angehört auch ausschließlich eine Darstellung eines *ObfuscationLevel* sein, der der Klasse *LocationContext* zugeordnet ist.

[159] schlägt zur Kontextverschleierung eine homogene Einteilung aller Typen von Kontextinformationen in vier Generalisierungsebenen vor. In [254] wird dies für die Kontexttypen Ort und Aktivität auf eine beliebige Anzahl von Hierarchieebenen pro Kontexttyp erweitert. Im Gegensatz dazu wird hier angenommen, dass aufgrund der großen semantischen Unterschiede zwischen den einzelnen Kontexttypen sowie der Ausrichtung verschiedener Verschleierungsmechanismen auf angestrebten Schutzziele nicht per se von der Existenz oder der Anwendbarkeit solcher Hierarchien ausgegangen werden kann. Stattdessen sieht das Modell vor, dass jede Unterklasse von *Context* eigene Verschleierungsmechanismen besitzt, die sich auf diesen bestimmten Typ von Kontextinformationen spezialisiert hat und dabei eigene Parameter zur Verschleierung entlang unterschiedlicher Dimensionen einsetzt. Im Modell kann daher jeder Verschleierungsmechanismus durch einen eigenen *ObfuscationLevel* dargestellt werden, der losgelöst von anderen desselben Kontexttyps existiert.

Innerhalb einzelner Kontexttypen kann es jedoch wie z.B. der Standortinformationen – auch nach der Anwendung bestimmter Veschleierungsmechanismen – solche natürlichen Hierarchien geben. Dies wird in CoRe durch die transitive Eigenschaft *generalizationOf* abgebildet, die zwischen Instanzen derselben Unterklasse von *Representation* definiert werden kann und somit die Modellierung der hierarchischen Struktur bestimmter Kontextinformationen ermöglicht. Stellt man sich die Darstellung eines solchen Kontexttyps wieder als Baumstruktur vor, kann auf Basis dieser Eigenschaft die automatische Vergröberung von Information umgesetzt werden, indem die Repräsentation einer entsprechend hohen Hierarchieebene ausgewählt werden.

Zudem besitzt jede Instanz der Klasse *ObfuscationLevel* die funktionale Eigenschaft *hasCurrentItem*, anhand derer sich jeweils direkt nachvollziehen lässt, was die aktuellste *Representation* dieses Verschleierungsmechanismus ist. Diese zusätzliche Information vereinfacht die Auswahl der modellierten Repräsentationen für die Regelauswertung, da neben den aktuellen Kontextinformationen wie beschrieben auch historische Darstellungen im Modell verbleiben. In diesem Zusammenhang wird die asymmetrische, nicht-transitive Eigenschaft *directPredecessorOf* zwischen entsprechenden Repräsentationen definiert, mit deren Hilfe sich z.B. die Aktualität der Kontextinformationen festlegen lässt, um die Privatsphäre des Nutzers durch Preisgabe veralteter Werte zu schützen.

Die unterschiedlichen im Modell hinterlegten Repräsentationen eines Kontexttyps können mittels der *accessibleVia* Eigenschaft und der Existenz entsprechender *Grant*-Objekte für verschiedene Kontextempfänger freigegeben werden. Repräsentation, die nicht Bestandteil einer *accessibleVia*-Instanz sind, stehen nur auf der privaten Ebene zur Verfügung. Die Kennzeichnung jenes Verschleierungstyps, dessen aktuelle Repräsentation für den allgemeinen Zugriff auf den entsprechenden Kontexttyp auf der öffentlichen Ebene gedacht ist, erfolgt über die funktionale Relation *hasPublicLevel*.

Welche der zur Verfügung stehenden Darstellungsformen einem bestimmten Empfänger zugänglich gemacht wird, hängt davon ab, welche Privatsphä-

rebedürfnisse der Kontextinhaber in der aktuellen Situation gegenüber der anfragenden Entität hat. Vor dem Hintergrund des von ALPACA verfolgten konservativen Umgangs mit persönlichen Informationen muss dies mit Hilfe kontextabhängiger Freigaberegeln explizit durch den Nutzer festgelegt werden.

3.4.2.3 Zusammenfassung

In diesem Kapitel wurde die ontologiebasierte Umsetzung des CoRe-Modells vorgestellt, das den aktuellen Kontext eines Nutzers privatsphärezentrisch modelliert, indem es für unterschiedliche Empfänger mehrere, semantisch verschiedene Repräsentationen derselben Kontextinformationen vorhält.

Durch geeignete Kombination verschiedener Ontologiekonzepte wurde bereits auf Modellebene sichergestellt, dass jeder potentielle Kontextempfänger in jeder Situation maximal Zugriff auf eine einzige Instanz einer Unterklasse von *Grant* erhält, welche wiederum nur für eine *Representation* gilt. Wie sich auf Basis des CoRe-Modells solche situations- und rezipientenabhängigen Freigabeentscheidungen umsetzen lassen, wird im Folgenden beschrieben.

3.4.3 Feingranulare und situationsabhängige Freigabe von Kontextinformationen

Wie in Kapitel 3.4.1 beschrieben, nimmt ALPACA bei der Verwaltung von Kontextinformationen eine konservative Haltung ein: Die Kommunikation von Informationen an Dritte wird durch einen Whitelist-Ansatz verwaltet, sodass der Kontextinhaber jeder Freigabe von Daten explizit zugestimmt haben muss.

In diesem Abschnitt wird ein Mechanismus entwickelt, der auf Basis des soeben vorgestellten CoRe-Modells die Erstellung situations- und rezipientenabhängiger Freigaberegeln erlaubt. Funktionsweise und Ablauf der Regelauswertung werden anhand einiger Beispiele gezeigt und die Möglichkeit zur Erkennung von inkonsistenten Freigaberegeln zur Laufzeit erläutert.

3.4.3.1 Kontextabhängige Freigabe von Informationen

Ziel der Kontextverwaltung ist es, dem Kontextinhaber feingranulare und kontextbezogene Freigabeentscheidungen zu ermöglichen. Hierzu wird das Konzept der *Release Trigger* eingeführt. Ein Trigger stellt die technische Umsetzung einer kontextabhängigen Privatsphärepräferenz des Nutzers dar und kann auf Basis aller im Modell hinterlegten Informationen definiert werden. Auf Grundlage dieser Trigger lassen sich somit situationsabhängige Freigaberegeln erstellen, welche dynamisch der eingesetzten Whitelist hinzugefügt werden.

Der schematische Aufbau eines Triggers und seiner Bestandteile ist in Abb. 3.5 zu sehen. Ein Trigger wird über einen eindeutigen Bezeichner identifiziert. Darüber hinaus besteht er aus einer Menge an sogenannten *Conditions*, einem *Effects*-Abschnitt und einem *Metadata*-Bereich, der zusätzliche Hinweise zu

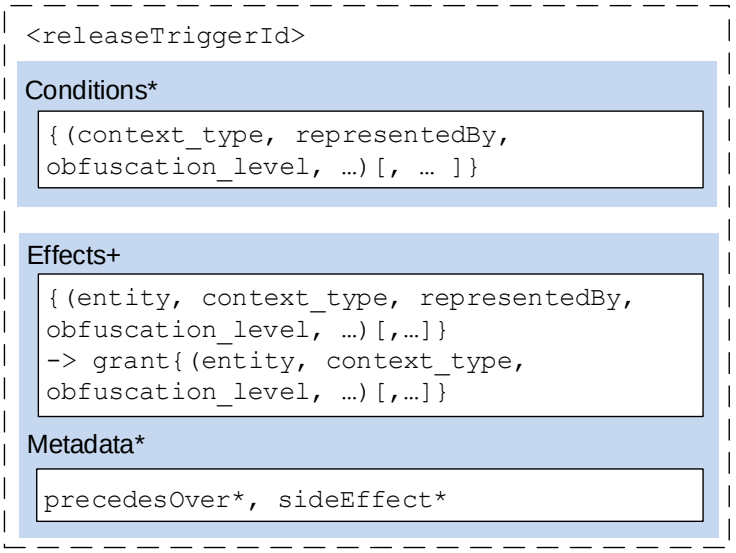


Abbildung 3.5: Schematischer Aufbau eines Release Triggers: *Conditions* und *Effects* erlauben die situationsabhängige Freigabe von Kontextrepräsentationen.

diesem Trigger speichert. Während die Bedingungs- und Metadatenabschnitte optional sind, muss ein Trigger verpflichtend einen gewissen Effekt haben. Andernfalls bewirkt er keine Änderungen hinsichtlich der Freigabe von Informationen und wäre überflüssig.

Der *Conditions*-Teil enthält alle Bedingungen, die an die aktuelle Situation des Nutzers, d.h. des Kontextinhabers, gestellt werden, um den jeweiligen Trigger zu aktivieren. Pro Trigger können beliebig viele Bedingungen eingetragen werden, die bei der Auswertung mittels logischer `UND`-Verknüpfung interpretiert werden. Sind keine Bedingungen angegeben, trifft die jeweilige Regel immer zu. Dadurch lassen sich statische Freigaberegeln definieren, die unabhängig von der aktuellen Sitation des Nutzers eine bestimmte Repräsentation an einen Kontextempfänger freigeben.

Die einzelnen Bedingungen können auf die Repräsentationen eines beliebigen Kontext- und Verschleierungstyps des CoRe-Modells Bezug nehmen. Dadurch lässt sich sowohl eine freie Kombinierbarkeit unterschiedlichster Kontextinformationen erreichen als auch die Verwendung verschiedener Granularitätsstufen des modellierten Kontexts.

Im Normalfall beziehen sich die Bedingungen eines Triggers auf den echten Kontext des Nutzers, dessen Repräsentationen sich über die entsprechenden Unterklassen von *ObfuscationNone* identifizieren lassen. Der Nutzer kann dies jedoch frei entscheiden und hat so die Möglichkeit, Regeln anhand der ihm verständlichsten Darstellungsform zu definieren. So ist z.B. die Angabe eines Städtenamens meist sicherlich intuitiver als die Definition eines Mittelpunktes anhand geographischer Koordinaten und eines Umkreises.

Im *Effects*-Abschnitt wird festgelegt, welche Repräsentationen von Kontex-

tinformationen in der durch die *Conditions* beschriebenen Situation an einen bestimmten Kontextempfänger ausgegeben werden dürfen. Um dabei zwischen verschiedenen Kontextempfängern zu differenzieren, können diese entweder explizit identifiziert oder implizit charakterisiert werden.

Die Identifizierung erfolgt dabei anhand des eindeutigen Bezeichners, der einem Individuum einer Unterklasse von *Entity* im Kontextmodell über die *hasID*-Eigenschaft zugeordnet wurde. Handelt es sich dabei um andere Nutzer, kann dies über ihre Emailadresse oder Mobilfunknummer erfolgen. Im Fall von Anwendungen, die auf dem Gerät installiert sind, bietet sich hierfür unter Android z.B. die entsprechende Linux User-ID oder der vollständige Package-Name der Applikation an.

Um Kontextempfänger darüber hinaus auch über die Angabe verschiedener Eigenschaften zu beschreiben, können diese auch über ihre Zugehörigkeit zu bestimmten Gruppen oder ihre eigenen Kontexteigenschaften definiert werden. So lassen sich Dienste und Libraries beispielsweise danach filtern, zu welcher Kategorie (Werbung, ortsbezogene Dienste, soziale Anwendungen, etc.) sie gehören oder ob sie gerade im Vordergrund oder im Hintergrund laufen.

Die Einordnung anderer Nutzer wird analog zur Kontaktverwaltung in online sozialen Netzwerken in Form von Gruppen organisiert, die der Nutzer als Instanzen der Klasse *PeerEntityGroup* über ein geeignetes graphisches Interface selbst erstellen kann. Im Modell wird die Gruppenmitgliedschaft eines Individuums über die Relation *memberOf* abgebildet, wobei ein Individuum Mitglied mehrerer Gruppen sein kann.

Zudem können Regeln erstellt werden, die anfragende Entitäten anhand deren eigenen Kontext näher beschreiben. Beispielsweise kann der Nutzer festlegen, dass sein genauer Aufenthaltsort im Bürogebäude nur den Kollegen mitgeteilt wird, die ebenfalls anwesend sind. Die Bedingungen, die dabei ggf. an die Situation des Kontextempfängers gestellt werden, dürfen sich – wenn es sich um einen anderen Nutzer handelt – vor dem Hintergrund der Ausgewogenheit im Informationsfluss jedoch nur auf solche Kontexttypen mit maximal demselben *ObfuscationLevel* beziehen, die in den *Effects* selbst freigegeben werden. Dieser Aspekt wird auch durch das in Kapitel 3.4.4.2 vorgeschlagene Protokoll zum Austausch von Kontextinformationen berücksichtigt und eingesetzt.

Durch die Auswertung des *Effects*-Teils eines Triggers werden die darin beschriebenen Repräsentationen des Nutzerkontexts dem Kontextempfänger zur Verfügung gestellt. Im Modell geschieht dies durch Erzeugung neuer Instanzen der entsprechenden *Grant*-Unterklassen. Über die *grantFor*-Relation werden diese eindeutig den in den *Effects* charakterisierten Repräsentationen zugeordnet. Gleichzeitig wird auch das Individuum, das den Kontextempfänger darstellt, über die *hasGrant*-Relation mit dem neuen Grant verknüpft.

Im Abschnitt *Metadata* eines Triggers können zusätzliche Hinweise abgelegt werden, die zur vollständigen Beschreibung einer Freigabeentscheidung dienen. Hierzu zählen einerseits Nebeneffekte, die die Abfrage einer durch diesen Trigger freigegebenen Kontextinformation durch einen Kontextempfänger

ggf. auslösen soll. Unter Berücksichtigung der in Kapitel 2.2 beschriebenen grundsätzlichen Aspekte beim Umgang mit persönlichen Daten, kann hier z.B. festgelegt werden, dass der Nutzer unmittelbar über diesen Vorgang benachrichtigt wird, dass dieser in Form eines Zugriffslogs archiviert wird oder dass der Nutzer vor der Herausgabe von Informationen noch einmal explizit um Erlaubnis gefragt wird. Diese Bestimmungen werden vom *Privacy Manager* (Kapitel 3.4.4.1) bei der Beantwortung von Kontextanfragen entsprechend umgesetzt. Darüber hinaus kann innerhalb dieser Metadaten auch die relative Wichtigkeit von Triggern im Fall widersprüchlicher Regeldefinitionen angegeben werden, sodass stets klar ist, welche Regel in einer bestimmten Situation für einen bestimmten Kontextempfänger zum Tragen kommen soll.

Sowohl die vom Nutzer festgelegten Nebeneffekte als auch die vergebenen Prioritäten werden dem CoRe-Modell hinzugefügt, sodass sie dem später vorgestellten Privacy Manager bei der Kontextverwaltung zur Verfügung stehen.

Obwohl der hier vorgestellte Trigger-Mechnismus nur *Grant*-Instanzen verwaltet und keine explizite „*deny*"-Option kennt, lassen sich mit Hilfe dieses Ansatzes auch Negativ-Regeln umsetzen: Sollen einem Kontextempfänger beispielsweise immer hochwertige Informationen zur Verfügung gestellt werden, außer in bestimmten Situationen, kann ein Trigger definiert werden, der unter den entsprechenden Bedingungen auf verschleierte Repräsentationen verweist.

3.4.3.2 Kontextbezogene Auswertung von Freigaberegeln

Mit Hilfe des Trigger-Mechanismus lassen sich nicht nur kontextbezogene Freigaberegeln definieren, sondern auch die Regelauswertung selbst kann kontextabhängig ausgeführt werden. Hierfür wird unabhängig von eingehenden Kontextanfragen der aktuelle Nutzerkontext überwacht und die aktuell gültige Teilmenge an Triggern ggf. angepasst, wodurch die in [22] aufgestellte Anforderung einer kontextabhängigen Regelauswertung erfüllt wird. Dies hat den Vorteil, dass im Zuge einer eingehenden Kontextanfrage lediglich das in der aktuellen Situation gültige Subset an Freigaberegeln überprüft werden muss.

3.4.3.2.1 Zweistufiges Verfahren der Regelauswertung Um dies zu erreichen, findet diese Auswertung in zwei Stufen statt. In der ersten Stufe wird ausschließlich der *Conditions*-Teil eines Triggers mit dem aktuellen Nutzerkontext verglichen. Hierfür wird bei der Triggererstellung darauf geachtet, dass nur solche Bedingungen in den *Conditions*-Abschnitt übernommen werden, die sich auf die aktuelle Situation des Kontextinhabers beziehen und nicht auf die eines potentiellen Kontextempfängers. Durch diese Zweiteilung lassen sich die in den *Conditions* beschriebenen Situationen auch für verschiedene Freigaberegeln wiederverwenden, z.B. für den Fall, dass in derselben Situation unterschiedliche Trigger für verschiedene Empfänger aktiv sein sollen.

Ändern sich die Kontextinformationen des Kontextinhabers, werden zunächst alle *Grant*-Instanzen zusammen mit allen Relationen, an denen diese

beteiligten waren, aus dem Modell gelöscht. Zudem werden alle derzeit als aktiv gekennzeichneten Trigger entfernt, da diese in der neuen Situation ggf. nicht mehr gültig sind. Als nächstes werden die *Conditions* aller definierten Trigger auf Übereinstimmung mit der aktuell modellierten Situation des Nutzers hin untersucht. Trifft eine bestimmte Regel zu, d.h., alle angegebenen Bedingungen stimmen mit dem aktuellen Nutzerkontext überein, wird der entsprechende Trigger dem Modell wieder hinzugefügt und als aktiv gekennzeichnet.

Als Abschluss der ersten Stufe werden für jeden nun aktiven Trigger entsprechende Instanzen der durch diesen Trigger verwalteten Unterklassen von *Grant* im Modell instanziiert. Diese werden jedoch noch nicht mit einer Kontextrepräsentation oder Kontextempfängern verknüpft, da dies erst zum Anfragezeitpunkt möglich ist. Über das *owl:AllDifferent*-Axiom werden alle *Grant*-Instanzen wieder als unterschiedliche Individuen gekennzeichnet, um die korrekte Validierung des Modells zu ermöglichen. Dieser Prozess wird jedes Mal durchgeführt, wenn sich der relevante Nutzerkontext ändert.

Die zweite Stufe der Regelauswertung beginnt, wenn tatsächlich eine Kontextanfrage von einem Kontextempfänger eingeht. Während die erste Stufe proaktiv durchgeführt wird, um irrelevante Regeln zur Anfragezeit nicht untersuchen zu müssen, findet dieser zweite Schritt nur bei Bedarf statt. Die in der ersten Stufe als aktiv markierten Trigger werden nun dahingehend untersucht, ob der Kontextempfänger sowie ggf. sein aktueller Kontext den Bedingungen im Trigger entspricht. Für jeden Trigger und für jeden Kontextempfänger, auf den das zutrifft, werden die entsprechenden *Grant*-Instanzen aus dem ersten Schritt nun mit den im Trigger beschriebenen Kontextrepräsentationen und dem Empfänger verknüpft.

Im Anschluss an die Regelauswertung wird die Konsistenz des aktuellen Modellzustands validiert, um mögliche Konflikte in den aktiven Regeln zu erkennen. Wurden keine Inkonsistenzen erzeugt, kann durch Abfrage des Modells, mit welchen Grants der anfragende Kontextempfänger nach der Triggerüberprüfung ausgestattet ist, die Kontextanfrage mit der entsprechenden Repräsentation beantwortet werden.

3.4.3.2.2 Aufspaltung eines Triggers in Teilregeln Für die Umsetzung der kontextabhängigen Regelauswertung werden die Trigger in zwei Teile zerlegt, die getrennt voneinander verwaltet werden. Wie die sich die Aufteilung eines Triggers gestaltet, kann im Vergleich der Abb. 3.6 mit den beiden Abbildungen 3.7 and 3.8 gesehen werden. Die erste Abbildung zeigt dabei den schematischen Aufbau des gesammten Triggers, die anderen die resultierende Aufteilung in zwei Teilregeln.

In einem Regelset werden dabei die *Conditions* aller Trigger als einzelne Regeln hinterlegt. Diese betreffen wie beschrieben ausschließlich den Kontext des Kontextinhabers und müssen neu evaluiert werden, wenn sich dieser ändert. Einziger Vorgriff auf den *Effects*-Abschnitt des Triggers ist der Hinweis, welche Typen von Kontextinformationen durch diesen Trigger freigegeben wer-

```
1  Trigger1 {
2    Condition:
3      Context:        TimeContext
4      representedBy: "worktime"
5    Condition:
6      Context:        LocationContext
7      representedBy: "workplace"

9    Effect:
10     Context:        LocationContext
11     Obfuscation:    SemanticLocationObfuscationRoom
12     Entity:         coworkers
13       Context:        LocationContext
14       representedBy: "workplace"
15 }
```

Abbildung 3.6: Der schematischer Aufbau eines exemplarischen Triggers, der in Abhängigkeit von Tageszeit und Aufenthaltsort des Kontextinhabers die Freigabe des Standorts festlegt.

den, falls er später feuert. Dies ist notwendig, um die zu erzeugenden Grant-Individuen explizit der korrekten Unterklasse von *Grant* zuweisen zu können. In der Konsequenz fügt die erste Teilregel eines Triggers dem Modell die Aussage hinzu, dass dieser Trigger dem aktuellen Nutzungskontext gemäß nun aktiv ist und welche Kontexttypen davon betroffen sind (vgl. Abb. 3.7).

Im dazugehörigen zweiten Teil eines Triggers, in dessen Antezedenz der Kontextempfänger sowie ggf. Bedingungen an dessen Kontext beschrieben sind, findet sich in der Konsequenz die Übertragung des *Effects*-Abschnitt des Triggers sowie u.U. vorhandene Metadaten. Die zweiten Hälften der Trigger werden in einem getrennten Regelset verwaltet und können anhand des Wissens, welche Trigger gerade aktiv sind, kontextabhängig gefiltert werden, bevor sie an die für die Regelauswertung zuständige Komponente übergeben werden.

3.4.3.2.3 Beispielhafte Regelauswertung Das soeben beschriebene Zusammenspiel zwischen dem CoRe-Modell und dem Trigger-Mechanismus soll anhand eines Beispiels verdeutlicht werden. Die Regelsyntax entspricht dem Format, das die *General Purpose Rule Engine* des Opensource-Frameworks *Jena*[3] bei der Regeldefinition verwendet. Dieses verzichtet auf umständliche XML-Syntax und zeichnet sich daher durch eine gute Lesbarkeit aus. Anhand des Rechtspfeils ($\rightarrow$) ist ersichtlich, dass diese Regeln gemäß des datengetriebenen Prinzips der Vorwärtsverkettung (engl. *Forward Chaining*) auszuwerten sind. Der Teil vor dem Rechtspfeil drückt die Vorbedingungen aus, die erfüllt sein müssen, damit die dahinter stehenden Schlüsse gezogen werden.

Im Beispielszenario ist Alice die Kontextinhaberin. Sie hat unterschiedliche Release-Trigger definiert, die besagen, welche Informationen sie bereit ist, mit ihren Arbeitskollegen zu teilen.

Die abstrakte Beschreibung des ersten Triggers in Pseudocode ist in Abb.

[3] `https://jena.apache.org/`

```
1  [atWork:
2    (Alice     hasContext      ?time)
3    (?time     rdf:type        TimeContext)
4    (?timeobf  obfuscationOf   ?time)
5    (?timeobf  rdf:type        SemanticTimeObfuscationNone)
6    (?timeobf  hasCurrentItem  ?timerep)
7    (?timerep  description     "worktime")
8    (Alice     hasContext      ?loc)
9    (?loc      rdf:type        LocationContext)
10   (?locobf   obfuscationOf   ?loc)
11   (?locobf   rdf:type        SemanticLocationObfuscationPlace)
12   (?locobf   hasCurrentItem  ?locrep)
13   (?locrep   description     "workplace")
14   ->
15   (Trigger1  rdf:type        ReleaseTrigger)
16   (Trigger1  isActive        "true")
17   (Trigger1  controls        LocationContext)
18 ]
```

Abbildung 3.7: Der *Conditions*-Abschnitt von `Trigger1`.

3.6 zu sehen. In natürlicher Sprache sagt dieser Trigger aus, dass während der Arbeitszeit und wenn sie sich an ihrem Arbeitsplatz befindet, die Kollegen auf Raumgenauigkeit erfahren dürfen, wo Alice sich gerade aufhält. Sie setzt dafür allerdings voraus, dass die Kollegen ebenfalls an der Arbeitsstätte sind.

Bei der Definition der Trigger wird der *Conditions*-Abschnitt wie beschrieben in eine eigene Regel ausgelagert. Eine korrespondierende zweite Regel beschreibt die Auswirkungen des Triggers, Anforderungen an den Kontext der anfragenden Entität sowie eventuell vorhandene Metadaten. Die beiden Teilregeln, die dem vorigen Abschnitt entsprechend aus `Trigger1` erzeugt werden, sind in Abb. 3.7 und 3.8 dargestellt.

In ihren Vorbedingungen beschreibt diese erste Teilregel, in welcher Situation der Trigger aktiviert werden soll (Zeilen 2-11). Im Beispiel ist das der Fall, wenn der zeitliche Kontext (2) von Alice von der aktuellsten, semantischen Darstellungsform dieses Kontexttyps (4) durch das Literal `worktime` (6) beschrieben wird. Analog dazu wird festgelegt, dass ihr aktueller Ort (7) im Modell in einer ebenfalls semantischen Darstellungsart (9) als ihr Arbeitsplatz beschrieben wird (11).

In der Konsequenz der `atWork`-Regel wird der entsprechende Trigger dem Modell hinzugefügt und auf aktiv gesetzt. Damit dies geschieht, müssen alle Vorbedingungen der Regel erfüllt sein. Da diese Bedingungen nur auf Informationen über Alice Bezug nehmen, kann der erste Teil des Triggers proaktiv ausgewertet werden, wenn sich ihr Kontext ändert, obwohl der Trigger als Ganzes auch auf den Kontext der anfragenden Entität Bezug nimmt. Die `atWork`-Regel sorgt demnach dafür, dass `Trigger1` im Kontextmodell aktiviert wird, sobald sich Alice an einem Arbeitstag an ihrem Arbeitsplatz befindet.

Ihr Kollege Bob ist aktuell im Büro und möchte sich mit ihr spontan über ein Projekt unterhalten. Er tritt daher als Kontextempfänger auf und schickt eine entsprechende Kontextanfrage an Alice. Der entsprechende Trigger verlangt dabei jedoch, dass ihre Kollegen diese Information über Alice nur bekommen,

```
1  [Trigger1:
2    (Alice     hasContext      ?loc)
3    (?loc      rdf:type        LocationContext)
4    (?locobf   obfuscationOf   ?loc)
5    (?locobf   rdf:type        SemanticLocationObfuscationRoom)
6    (?locobf   hasCurrentItem  ?locrep)
7    (?creq     memberOf        coworkers)
8    (?creq     hasContext      ?crloc)
9    (?crloc    type            LocationContext)
10   (?crloc    representedBy   ?crlocrep)
11   (?crlocrep description     "workplace")
12   (?grant    rdf:type        LocationGrant)
13   (?grant    createdBy       "Trigger1")
14   ->
15   (?grant    GrantFor        ?locrep)
16   (?creq     hasGrant        ?grant)
17 ]
```

Abbildung 3.8: Die zweite Teilregel von `Trigger1` enthält den *Effects*-Abschnitt des Release-Triggers.

wenn sie sich selbst auch am Arbeitsplatz befinden. Aufgrund früherer Kontextabfragen weiß Bobs Endgerät das bereits und fügt diese Information der Anfrage bei. Wie sich der Kommunikationsablauf zwischen zwei Nutzern allgemein umsetzen lässt, wird in Kapitel 3.4.4.2 genauer beschrieben.

Anhand der Darstellung in Abb. 3.8 kann nachvollzogen werden, wie die zweite Hälfte eines Triggers aufgebaut ist. In den Zeilen 2-5 findet sich die Beschreibung der Kontextrepräsentation, die durch diesen Trigger freigegeben wird. In Zeile 4 wird ausgedrückt, dass der Standort als semantische Darstellung auf Raumebene zurückgeliefert werden soll. Zeile 5 stellt sicher, dass es sich dabei um die aktuellste Repräsentation dieser Darstellungsart handelt. An dieser Stelle könnte sich Alice in anderen Situationen z.B. auch dazu entscheiden, zum Schutz ihres aktuellen Aufenthaltsorts anstelle der aktuellen Repräsentation eine veraltete Version auszuwählen. In einem solchen Fall würde ein entsprechender Trigger um ein Triplett erweitert, das die *freshness* einer Repräsentation beschreibt. Anstelle der *hasCurrentItem*-Relation würde *hasItem* verwendet, um die Suche nicht auf die aktuelle Repräsentation einzuschränken.

Der Kontextempfänger und die Bedingungen an dessen Kontext werden in Zeile 6 sowie 7-10 beschrieben. So lässt sich dieser Trigger nur auf Nutzer anwenden, die der Gruppe `coworkers` angehören. Darüber hinaus muss der Ortskontext der anfragenden Person bekannt sein (7) und aussagen, dass sich die Person ebenfalls am Arbeitsplatz aufhält (10). In den Zeilen 11 und 12 wird die zuvor bereits erzeugte Instanz der Klasse *LocationGrant* referenziert, die von diesem Trigger verwaltet wird.

In der Konsequenz dieser Regel wird dieser Grant mit der beschriebenen Repräsentation des Ortskontexts von Alice verknüpft (14) und es werden alle Kontextempfänger, auf welche die obige Beschreibung zutrifft, mit dem Grant ausgestattet (15). Steht im Anschluss an die Regelauswertung fest, dass Bob einen *LocationGrant* auf den Ortskontext von Alice besitzt, wird ihm die

```
1  Trigger2 {
2    Condition:
3      Context:        TimeContext
4      representedBy: "worktime"

6    Effect:
7      Context:        LocationContext
8      Obfuscation:    SemanticLocationObfuscationCity
9      Entity:         coworker
10       Context:        LocationContext
11       representedBy: workplace
12 }
```

Abbildung 3.9: Schematischer Aufbau einer weiteren Freigaberegel, `Trigger2`, die nur auf den zeitlichen Kontext Bezug nimmt.

entsprechende Repräsentation zurückgeliefert.

Es soll nun angenommen werden, dass Alice noch einen zweiten Trigger definiert hat, der in Abb. 3.9 zu sehen ist. `Trigger2` nimmt nur Bezug auf den zeitlichen Kontext von Alice und stellt keine Bedingungen an ihren aktuellen Aufenthaltsort. Im Gegensatz zu `Trigger1` feuert diese Regel also immer, wenn ihre übliche Arbeitszeit beginnt und setzt den Trigger auf aktiv. Dies geschieht auch, wenn sich Alice nicht im Büro befindet, z.B. weil sie bei einem Kunden ist, einen Arzttermin hat oder schlicht spät dran ist. Zum Schutz ihrer Privatsphäre hat sich Alice dafür entscheiden, ihre Arbeitskollegen in diesem Fall nur wissen zu lassen, in welcher Stadt sie sich gerade befindet.

Möchte Bob nun wissen, wo sich Alice gerade aufhält, treten bei der Existenz beider Trigger abhängig von der Gesamtsituation unterschiedliche Fälle ein: Befindet sich Bob nicht an ihrem gemeinsamen Arbeitsplatz oder findet die Anfrage außerhalb von Alice' Arbeitszeiten statt, erlaubt ihm keiner der beiden Trigger Zugriff auf ihren Standort. Wenn er bei der Arbeit ist und Alice nicht, feuert nur `Trigger2` und gibt ihren Ort auf Städteebene für Bob frei.

Befinden sich jedoch beide im Bürogebäude, sind beide Trigger aktiv und feuern bei der eintreffenden Kontextanfrage. In diesem Fall würde Bob demnach mit zwei verschiedenen *Grant*-Instanzen ausgestattet und damit Zugriff auf unterschiedliche Repräsentationen von Alice' Aufenthaltsort erlangen. Es wäre somit nicht eindeutig, welche Information ihm zurückgegeben werden darf, was zu einer inkonsistenten Situation führt.

Es dürfen keinesfalls beide Repräsentationen zurückgegeben werden. Während dies im hier genanten Beispiel zwar das Problem lösen würde, sind viele Situationen denkbar, in denen ein solches Vorgehen die Integrität oder die Privatsphäre des Nutzers gefährden würde – z.B. wenn sich der Kontextinhaber zum Schutz seiner Privatsphäre verschiedener Verschleierungstechniken oder unwahrer Aussagen bedient und seinen echten Ort geheim halten möchte. Wie solche Konfliktsituationen mit Hilfe des CoRe-Modells erkannt und aufgelöst werden können, wird im nächsten Abschnitt beschrieben.

3.4.3.3 Vermeidung und Auflösung widersprüchlicher Freigaberegeln

Einen wichtigen Aspekt bei der Verwaltung von Kontextinformationen stellt aus Sicht der Privatsphäre die Erkennung von uneindeutigen bzw. widersprüchlichen Freigaberegeln dar. Hierfür gibt es unterschiedliche Lösungsansätze: In [22] plädieren die Autoren für eine pauschale Strategie, die sich im Zweifel je nach Grundeinstellung stets für „*deny*" oder „*permit*" entscheidet. [206, 34] hingegen setzen auf die Spezifität der angegebenen Regeln. Der zugrundeliegende Gedanke hierbei ist, dass je detaillierter eine Regel ist, desto wichtiger wird sie dem Kontextinhaber sein und ihre Entscheidung wird umgesetzt.

Im Rahmen der vorliegenden Arbeit wird jedoch argumentiert, dass – obwohl der Nutzer damit öfter belästigt wird – ihm die Entscheidungshoheit über die Freigabe von Kontextinformationen obliegen muss. Diese Annahme basiert auf folgenden Beobachtungen: Das Vertrauen eines Nutzers in ein System, das seine Privatsphäre gewährleisten soll, ist grundsätzlich höher, wenn er die Detailkontrolle über die Freigabe von Informationen behält. Daneben kann ein auftretender Regelkonflikt den Nutzer auf versehentliche Fehler bei der Regeldefinition hinweisen, die nur von ihm selbst seiner eigentlichen Intention entsprechend behoben werden können. Zudem kann davon ausgegangen werden, dass Regelkonflikte nicht häufig auftreten und der Kontextinhaber somit nicht zu oft gestört wird.

Um einen Regelkonflikt handelt es sich, wenn einem Kontextempfänger in einer Situation verschiedene Repräsentation desselben Kontexttyps zugestanden werden. Solche Situationen sind potentiell dazu in der Lage, die Integrität und Privatsphäre eines Nutzers zu verletzen. Dieses Szenario wird deutlich, wenn demselben Kontextempfänger z.B. zwei verschiedene Darstellungen des aktuellen Standorts – seinem tatsächlichen und einem gelogenen – zurückgegeben werden. Nicht nur, dass der Kontextempfänger nun wahrscheinlich gegen den Willen des Kontextinhabers den tatsächlichen Standort erfahren hat, er wurde darüber hinaus auch noch beim „Lügen" erwischt.

Eine triviale Möglichkeit für die Enstehung solcher Konflikte stellt die Angabe unterschiedlicher Trigger dar, in denen derselbe Kontextempfänger jeweils explizit identifiziert wird. Man könnte versuchen, diese Fälle mit Hilfe einer statischen Regelanalyse zu erkennen. Die jeweiligen *Conditions* der Trigger können sich dabei jedoch so unterscheiden, dass diese Trigger niemals zur selben Zeit aktiv werden und ein Konflikt – obwohl theoretisch möglich – nie auftritt. Eine statische Analyse könnte also zur Erzeugung falschpositiver Warnungen führen und soll daher vermieden werden.

Weiter an Komplexität gewinnen diese Fälle noch, wenn Kontextempfänger über ihre Zugehörigkeit zu verschiedenen Gruppen und unter Berücksichtigung ihres eigenen Kontexts beschrieben werden. Durch die Kontextabhängigkeit der Regeln können diese Fälle entweder nicht erkannt werden oder führen zu weiteren Falschpositiven. Es soll aber dennoch verhindert werden, dass ungewollte Informationen an einen Kontextempfänger versendet werden.

Um solche Konflikte zur Laufzeit erkennen zu können, wurde daher bereits

das CoRe-Modell mit entsprechenden Erkennungsmechanismen ausgestattet. Mit den Begriffen des Modells beschrieben liegt ein Konflikt genau dann vor, wenn ein Individuum der Klasse *Entity* zu einem Zeitpunkt mehr als eine Instanz derselben Unterklasse von *Grant* zugewiesen bekommt. Da jeder Grant eine unterschiedliche Repräsentation desselben Kontexttyps betrifft, würde dies dazu führen, dass der Kontextempfänger auf alle betroffenen Repräsentationen Zugriff erlangt, bzw. dass der Privacy Manager nicht entscheiden kann, welche Repräsentation zurückgegeben werden soll.

Zur dynamischen Erkennung solcher Situationen wird die durch die ontologiebasierte Modellierung gegebene Möglichkeit zur logischen Validierung des aktuellen Modellzustands eingesetzt. Durch die verschiedenen Restriktionen, die auf den Klassen und Relationen des CoRe-Modells definiert wurden, lassen sich widersprüchliche Regeln als inkonsistenter Modellzustand erkenen. Durch die in Kapitel 3.4.2.2 beschriebenen Einschränkungen bzgl. der Funktionalität von Relationen und der pro Klasse erlaubten Wertebereiche der betroffenen Relationen, werden insbesondere die folgenden Bedingungen sichergestellt:

1. Jede Instanz einer Unterklasse von *Entity* kann maximal je eine Instanz einer bestimmten Unterklasse von *Grant* besitzen.

2. Jede Instanz einer Unterklasse von *Grant* kann maximal auf eine Instanz einer dem Kontexttyp entsprechenden Unterklasse von *Representation* verweisen.

Entsteht durch das parallele Feuern unterschiedlicher Trigger eine Situation, in der mindestens eine dieser Bedingungen nicht eingehalten ist, wird der Modellzustand bei der anschließend standardmäßig durchgeführten Validierung automatisch als inkonsistent erkannt.

Um die Auflösung derartiger Konfliktsituationen zu ermöglichen, kann in den Metainformationen eines Triggers hinterlegt werden, vor welchem anderen Trigger dieser Vorrang genießt. Der Kontextinhaber kann beim ersten Auftreten eines Regelkonflikts darauf hingewiesen werden und eine Entscheidung treffen, welche der beide Regeln, die für diesen Konflikt verantwortlich sind, wichtiger ist. Diese Entscheidung wird als *precedesOver*-Hinweis in den Metadaten des vom Nutzer bestimmten, wichtigeren Triggers hinterlegt. Tritt in Zukunft erneut ein inkonsistenter Zustand auf, an dem diese beide Regeln beteiligt waren, können auf Basis dieser Information automatisch die von dem unterlegenen Trigger kreierten *hasGrant*-Instanzen aus dem Modell gelöscht werden. Die Zuteilung des überlegenen Grants an den Kontextempfänger bleibt hingegegen bestehen, sodass wieder eindeutig feststeht, welche Repräsentation der Kontextinhaber für diesen Empfänger vorgesehen hat.

3.4.3.4 Zusammenfassung

In diesem Kapitel wurde ein kontextabhängiger Mechanismus für die Umsetzung der whitelistbasierten Freigabe von Kontextinformationen präsentiert.

Die Privatsphärepräferenzen des Kontextinhabers können dabei in Form sogenannter *ReleaseTrigger* beschrieben werden, welche die Erstellung feigranularer, kontext- und rezipientenabhängiger Freigaberegeln ermöglichen. Abhängig von seiner eigenen Situation, der Charakterisierung des Kontextempfängers und dessen aktuellen Kontext, lässt sich damit die grundsätzliche Preisgabe und der Detailgrad von Kontextinformationen definieren.

Abschließend wurde gezeigt, wie auf Basis der im zuvor neu eingeführten CoRe-Modell integrierten Konzepte, Relationen und Restriktionen auch widersprüchliche Triggerdefinitionen dynamisch erkannt und durch einmalige Rückfrage an den Benutzer aufgelöst werden können.

3.4.4 Systemarchitektur und Kommunikationsablauf

In den vorangehenden Kapiteln wurden aufbauend auf einer abstrahierten Sichtweise auf die Privatsphärebedürfnisse eines Nutzers bereits zwei wichtige Komponenten für die Realisierung eines umfassenden Systems zur privatsphärezentrischen Verwaltung von Kontextinformationen vorgestellt.

Den ersten Baustein stellt das ontologiebasierte Kontextmodell CoRe dar. Durch die Integration privatsphärerelevanter Aspekte in das Modell wird einerseits die feingranulare Freigabe von Kontextinformationen durch den Kontextinhaber ermöglicht. Zudem wurde das Modell so entworfen, dass Situationen, die die Privatsphäre des Nutzers potentiell gefährden könnten bei der Validierung des Modellzustands als Inkonsistenzen erkannt werden. Darauf aufbauend wurde ein kontextabhängiger Trigger-Mechanismus präsentiert, der die Definition feingranularer Freigaberegeln erlaubt.

In diesem Kapitel werden diese Bestandteile durch den zentralen *Privacy Manager* integriert sowie die vollständige Systemarchitektur von ALPACA vorgestellt. Es wird beschrieben, wie die einzelnen Teilkomponenten zusammenspielen und wie der Privacy Manager eine Anfrage durch einen Kontextempfänger verarbeitet. Anschließend wird skizziert, wie sich ein System wie ALPACA in ein modernes mobiles Betriebssystem integrieren lässt.

3.4.4.1 Der Privacy Manager als zentrale Komponente zur Akquise und Freigabe von Kontextinformationen

Die zentrale Rolle bei der Verwaltung von Kontextinformationen mit ALPACA nimmt der *Privacy Manager* (PM) ein. Dieser ist für die gesamte Verwaltung der Sensoren, Kontextquellen und -informationen verantwortlich und setzt die Privatsphärepräferenzen des Kontextinhabers durch. Gemäß der Definition kontextabhängiger Anwendungen stellt diese Komponente selbst eine kontextabhängige Anwendung dar, da ihre Diensterbringung – die situationsabhängige Freigabe von Informationen – auf Kontext beruht.

Wie in Kapitel 3.4.1 beschrieben, hat der Nutzer unterschiedliche Gestaltungsmöglichkeiten bei der Kontextverwaltung. Er kann situationsabhängig

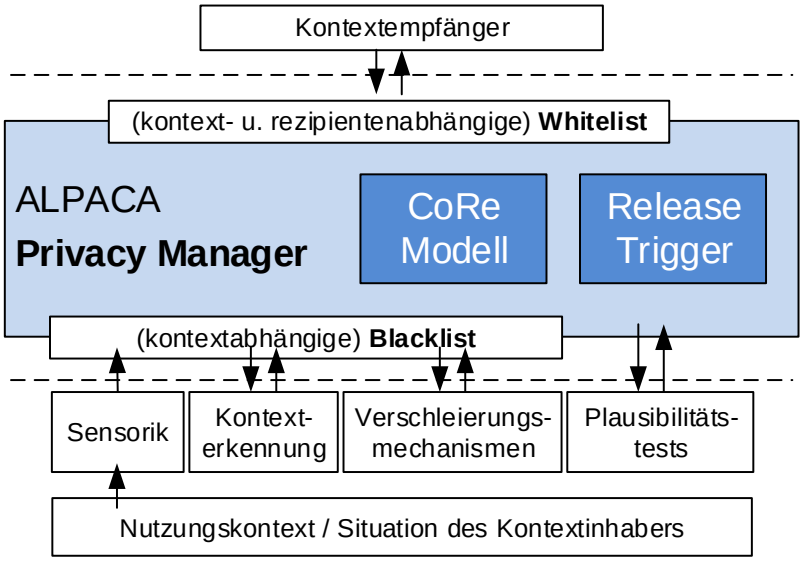


Abbildung 3.10: Der *Privacy Manager* stellt die Kernkomponente von ALPACA dar und agiert sowohl für die Kontextquellen als auch die Kontextempfänger als exklusive Schnittstelle zu den Kontextinformationen des Nutzers.

über einen Blacklist-Mechanismus festlegen, welche Sensorwerte und Kontextinformationen überhaupt in die aktuelle Beschreibung seines Kontexts im Modell gelangen dürfen. Zudem kann er durch die explizite Freigabe bestimmter Informationen kontextabhängig definieren, welche Informationen in welchem Detailgrad an einen bestimmten Empfänger weitergegeben werden dürfen.

Der schematische Aufbau von ALPACA sowie die Platzierung des Privacy Managers innerhalb dieser Architektur ist in Abb. 3.10 zu sehen. Als eine Art Torwächter ist der Privacy Manager die einzige Komponente, die direkt auf das Kontextmodell und dessen Daten zugreifen kann.

Im Rahmen der vorliegenden Arbeit wird davon ausgegangen, dass – wie auf einem aktuellen Smartphone üblich – eine variable Anzahl an unterschiedlichen Sensoren zur Ermittlung von Kontextinformationen zur Verfügung steht. Zusätzlich sind u.U. ebenfalls mit Sensorik ausgestattete Wearables wie Smartwatches, Fitnessarmbänder, usw., über Bluetooth mit dem Endgerät des Nutzers gekoppelt. Daneben stehen verschiedene Verfahren zur Ermittlung von semantisch angereicherten Kontextinformationen zur Verfügung, wie in Kapitel 2.1.3 einige beschrieben wurden. Zusätzlich ist eine ebenfalls beliebige Zahl an Verschleierungsverfahren implementiert, die für einen bestimmten Typ von Kontextinformation und einer gewissen Zielsetzung folgend Techniken zum Schutz der Privatsphäre auf die echten Kontextinformationen anwenden.

Verfahren zur Plausibilitätsprüfung von aufeinanderfolgenden Kontextanfragen ermöglichen die Einhaltung der zeitlichen Konsistenz für einen bestimmten Typ von Kontext. Neue Sensoren, Klassifizierungs- und Verschleierungsmecha-

nismen lassen sich einfach hinzufügen, indem sie nach ihrer Installation beim beim Privacy Manager registriert werden.

Die Verfahren zur Kontexterkennung und -verschleierung sind jeweils dafür verantwortlich, die für sie ermittelbaren Kontextinformationen eines Benutzers zu messen bzw. weiterzuverarbeiten. Die gemachten Beobachtungen melden sie an den Privacy Manager, zusammen mit allen zur Verfügung stehende Metainformationen. Dieser wiederum fügt diese Erkenntnisse als neue Individuen der jeweiligen Unterklasse von *Representation* in das Kontextmodell ein, löscht die bestehenden Instanzen der *hasCurrentItem*-Relation der entsprechenden *ObfuscationLevel* und trägt die neue Repräsentation dann als aktuelle Darstellung ein. Die zum weiteren Reasoning und ggf. zur Anpassung des Detailgrads nötigen Zusatzinformationen wie *freshness*, *accuracy*, *confidence* und den jeweiligen *ObfuscationLevel*, den ein bestimmter Erkennungs- oder Verschleierugsmechanismus erzeugt, werden dabei mit angegeben.

Da bei diesem Vorgang der Zustand des Kontextmodells verändert wird, entfernt der Privacy Manager daraufhin alle Individuen der Klassen *Grant* und *Trigger* aus dem Modell und veranlasst die erste Stufe der Regelauswertung. Um überflüssige Regelauswertungen zu verhindern, wird dies jedoch nur nach dem Hinzufügen solcher Kontextinformationen ausgeführt, die im Bedingungsteil einer Regel vorkommen. Zudem können mehrere Kontextänderungen innerhalb eines Zeitabschnitts vom Privacy Manager gesammelt und Bulk-artig in das Modell überführt werden. Für die optimale Einstellung eines solchen Intervalls bietet sich je nach Stabilität von Kontextinformationen und der Häufigkeit von Kontextanfragen selbst wiederum eine kontextabhängige Festlegung an.

Informationen, die kontinuierlich anfallen, wie z.B. der Datenstrom eines hochfrequenten Hardwaresensors, sind von diesem Vorgang ausgenommen. Stattdessen werden diese, wenn der Nutzer das in der Blacklist nicht ausgeschlossen hat, direkt den vom Privacy Manager kontrollierten Verfahren zur Kontexterkennung zur Verfügung gestellt, um die Performanz dieser Algorithmen zu gewährleisten.

Diesem Vorgehen liegt die Annahme zugrunde, dass ein Nutzer seine Privatsphärepräferenzen kaum anhand roher Sensordaten, z.B. des Gyroskops, erstellen wird. Stattdessen wird er diese Regeln auf Basis semantisch angereicherter Beschreibungen festlegen, die erst durch die Interpretation der Sensorwerte durch solche Klassifizierungsverfahren entstehen. Diese wiederum zeichnen sich durch eine deutlich größere Stabilität aus und werden wie oben beschrieben in das Kontextmodell überführt und zur Regelauswertung verwendet.

ALPACA ermöglicht zudem den Zugriff auf rohe Sensormesswerte durch eine bestimmte Anwendung sowie die Kontrolle darüber. Über die erkannten Situationen kann der Nutzer detailliert festlegen, wann ein bestimmter Sensor abgeschaltet werden soll oder welche Kontextempfänger direkt darauf zugreifen dürfen. Wird später eine Situation erkannt, die eine Änderung in den aktiven Freigaberegeln hervorruft, wird ein entsprechender Grant ggf. entfernt und die Weitergabe der Sensorwerte an eventuelle Kontextempfänger abgebrochen.

Der Privacy Manager ist somit für jegliche Freigabe von Kontextinformationen und Sensordaten an einen beliebigen Kontextempfänger zuständig. Wie in Kapitel 3.4.3 beschrieben, wird dazu ein kontextabhängiger, requestbasierter Freigabemechanismus implementiert.

Um die Integrität des Nutzers auch in solchen Situationen zu schützen, in denen er sich zur Herausgabe unwahrer Angaben an einen Kontextempfänger entschieden hat, führt der Privacy Manager neben der situativen Konsistenzprüfung von Freigaberegeln auch Plausibilitätsprüfungen zur Einhaltung der zeitlichen Konsistenz durch. Auf Basis der ebenfalls in CoRe gespeicherten, zuletzt an eine Entität zurückgegebene Kontextantwort und der aktuelle freigegebenen Repräsentation kann festgestellt werden, ob die Antworten auch über die Zeit hinweg konsistent sind.

Mit Hilfe von Plausibilitäts-Tests lässt sich z.B. feststellen, ob ein Nutzer innerhalb der Zeitspanne zwischen zwei Standortupdates die entsprechende Distanz tatsächlich zurückgelegt haben kann [33]. Wird eine inkonsistente Situation entdeckt, fragt der Privacy Manager beim Nutzer nach, wie die Kontextanfrage vor diesem Hintergrund beantwortet werden soll.

Bei der Anfrage von Kontextinformationen durch einen anderen Nutzer kann es sein, dass der Kontext der anfragenden Entität selbst ausschlaggebend für die jeweilige Antwort ist. In diesem Fall überprüft der Privacy Manager mittels derselben Verfahren auch die zeitliche Konsistenz aufeinanderfolgender Kontextanfragen. Damit wird verhindert, dass ein Kontextempfänger z.B. innerhalb kurzer Zeit gezielt gefälschte Angaben über seinen eigenen Kontext macht, um mehr über die aktuelle Situation des Nutzers herauszufinden.

Aber auch den Kontextzugriff durch Anwendungen kann Nutzer an Bedingungen knüpfen, die den Kontext der Anwendung betreffen, z.B. dass sie den Zugriff nur erhält, während sie im Vordergrund läuft.

Wie der Kommunikationsablauf bei verschiedenen Typen von Kontextempfängern und die Verarbeitung der Anfrage durch den Privacy Manager dabei jeweils abläuft, wird im Folgenden erklärt.

3.4.4.2 Kommunikationsablauf und Anfragebearbeitung

Die Kontextverwaltung mit ALPACA soll sich generisch für verschiedene Ausprägungen kontextbezogener Dienste einsetzen lassen. Es lassen sich drei unterschiedliche Kategorien von Kontextempfängern identifizieren, die für diese Anwendungen jeweils charakteristisch sind:

- *Lokale Anwendungen*, die ausschließlich auf dem Endgerät des Nutzers laufen und daher gefahrlos Zugang zu den privaten Kontextinformationen erhalten dürfen.

- *Onlinedienste von Drittanbietern*, die Kontextinformationen abfragen und zur Weiterverarbeitung an externe Komponenten und andere Nutzer kommunizieren.

- *Andere Nutzer*, sog. Peers, zu denen der Kontextinhaber eine soziale Beziehung hat und mit denen er kontextabhängige Anwendungen nutzt, ohne dass ein Dritter die Daten einsehen kann.

Um die Privatsphärepräferenzen des Nutzers durchzusetzen, stellt der Privacy Manager bei jeder Anfrage zunächst fest, auf welcher der in Kapitel 3.4.1 beschriebenen Ebenen der Kontextempfänger eingeordnet ist. Wie die Identifizierung bzw. die Authentifikation von Kontextempfängern in diesem Zusammenhang umgesetzt wird, wird in Kapitel 3.4.4.4 näher beschrieben.

3.4.4.2.1 Kontextabfrage durch lokale Anwendungen Gehört der Kontextempfänger zur Klasse der rein lokalen Anwendungen, ist sie auf der privaten Ebene platziert. Zu diesem Zweck verwaltet der Privacy Manager eine statische Tabelle, in der die Zugehörigkeit zu dieser Ebene verwaltet wird. Der Privacy Manager muss in diesem Fall keine weiteren Regeln überprüfen, sondern kann direkt mit der in der Anfrage spezifizierten Repräsentation aus dem Kontextmodell antworten.

Zur Kategorie dieser vollständig vertrauenswürdigen Empfänger zählen u.a. auch Verschleierungsmechanismen und lokale Inferenzalgorithmen, die aus den Kontextinformationen des Modells höherwertige Tatsachen schließen. Diese sind ebenfalls auf der privaten Ebene einsortiert und haben vollen Zugriff auf den Kontext des Benutzers.

3.4.4.2.2 Kontextabfrage durch Onlinedienste Handelt es sich bei dem Kontextempfänger um den Dienst eines Drittanbieters, der nicht der privaten Ebene zugeordnet wurde, überprüft der Privacy Manager, ob für diese Entität eine passende *Grant*-Instanz in CoRe vorliegt. Hat der Kontextinhaber Trigger definiert, die diesem Empfänger in der aktuellen Situation Zugriff auf eine Repräsentation des angefragten Kontexttyps gestatten, fragt der Privacy Manager diese vom Modell ab und gibt sie als Antwort zurück. In diesem Fall wurde dem Kontextempfänger durch die Existenz des Triggers kontextabhängig der Zugriff auf geschützte Kontextinformationen erlaubt.

Liegt hingegen kein entsprechender Grant vor, wurde der Dienst in der aktuellen Situation damit implizit auf die öffentliche Ebene verwiesen. Der Privacy Manager sucht in diesem Fall entsprechend die aktuelle, mittels *hasPublicLevel* als öffentlich gekennzeichnete Repräsentation des angefragten Kontexttyps heraus und liefert diese an den Kontextempfänger zurück. Diese öffentlichen Informationen sollten so gewählt sein, dass sie möglichst unspezifisch sind und die Privatsphäre des Nutzers nicht tangieren. Der Nutzer selbst kann festlegen, ob er z.B. eine stark vergröberte Version des echten Kontexts, eine falsche Angabe oder eine unwahre, aber plausible Antwort wie „unbekannt" zurückliefern lassen möchte.

Zu dieser Kategorie von Anwendungen zählen sowohl individuelle, kontextbezogene Dienste wie eine POI-Suche als auch komplexe kontextabhängige

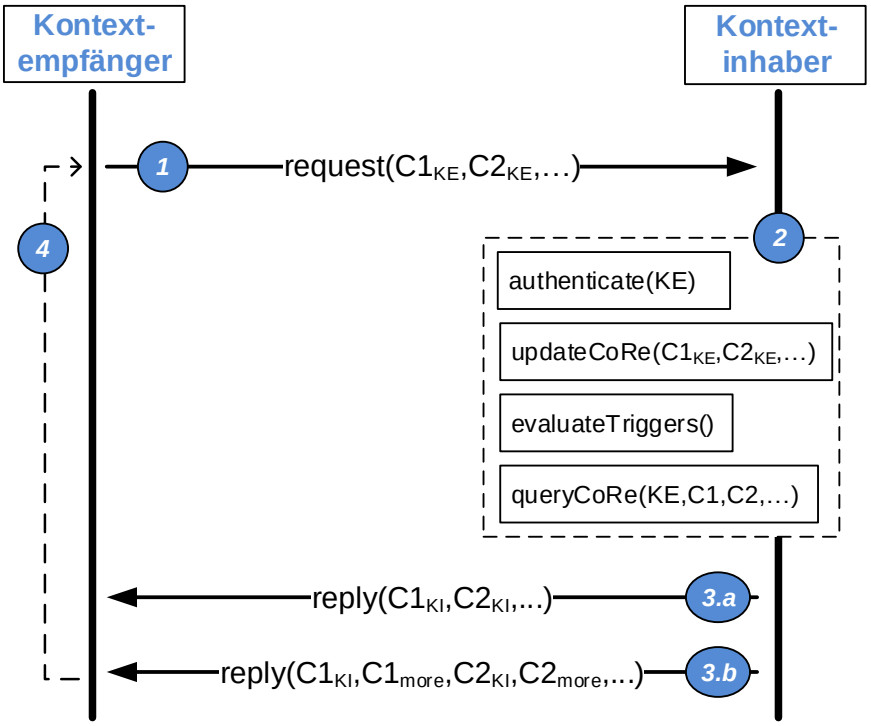


Abbildung 3.11: Für die Durchsetzung eines ausgewogenen Informationsflusses zwischen Kontextempfänger und -inhaber setzt ALPACA ein rundenbasiertes Kommunikationsprotokoll ein.

Anwendungen, die auf Basis der Kontextinformationen vieler Nutzer arbeiten (vgl. Kapitel 2.1.2).

3.4.4.2.3 Kontextabfrage durch Peers Interessante Möglichkeiten für die Umsetzung innovativer kontextbezogener Anwendungen bieten Ansätze, in denen auf die Existenz eines zentralen Dienstanbieters verzichtet wird. Insbesondere vor dem Gesichtspunkt der Privatsphäre stellt das eine großen Vorteil dar, da Kontextinformationen z.B. im Falle eines Dienstes, der mich über Freunde in der Nähe informiert, nicht mit einem Drittanbieter geteilt werden müssen, sondern im Rahmen einer Peer-to-Peer-Kommunikation ausschließlich an den eigentlich intendierten Kontextempfänger gehen. Beispiele hierfür sind Peer-to-Peer-basierte online soziale Netzwerke wie *Vegas* [72] oder die Klasse der kontextzentrischen sozialen Netze [250].

Bei der Verwendung solcher Dienste kann der Kontextinhaber wie bereits erwähnt auch Bedingungen an den Kontext des anfragenden Nutzers stellen. Darüber hinaus gilt es, eine gewisse Symmetrie beim paarweisen Austausch von Kontextinformationen einzuhalten, um ein Ungleichgewicht im Informationsfluss zwischen zwei gleichberechtigten Teilnehmern zu vermeiden [145]. Beide Aspekte sorgen dafür, dass die Anfrage der Kontextinformationen eines anderen Nutzers die Übermittlung des eigenen Kontexts impliziert. Das run-

denbasierte Protokoll, das ALPACA u.a. zum Zweck der Symmetrieerhaltung für diese Kommunikation vorsieht, ist in Abb. 3.11 zu sehen.

- *(1)* Im ersten Schritt schickt der Kontextempfänger (*KE*) eine Anfrage für bestimmte Kontextinformationen an den Kontextinhaber (*KI*). *KE* gibt implizit an, welchen Kontexttyp und welchen Detailgrad er erhalten möchte, indem er die entsprechenden Informationen über sich selbst dem Request hinzufügt. Um dabei nicht selbst ungewollt zu viele Informationen preiszugeben, beginnt *KE* mit einem niedrigen Detailgrad und orientiert sich an den eigenen derzeit aktiven Triggern, um auch hier den Privatsphäreanforderungen des anfragenden Nutzers zu entsprechen.

- *(2)* Der Privacy Manager von *KI* empfängt die Kontextanfrage, authentifiziert *KE* und fügt dessen mitgelieferten Kontextinformationen in CoRe ein. Es werden nun die aktiven Trigger ausgewertet, um zu ermitteln, welche Kontextinformationen für *KE* zur Verfügung stehen. Was im nächsten Schritt passiert, hängt davon ab, ob durch die Regelauswertung entsprechende Grants vorliegen oder nicht.

- *(3.a)* Existiert kein passender Grant, wird die öffentlich einsehbare Version des jeweiligen Kontexttyps an *KE* zurückgeliefert und das Protokoll endet hier.

- *(3.b)* Gibt es einen Grant, schlägt der Privacy Manager von *KI* die entsprechende Repräsentation in CoRe nach und passt ggf. die Detailstufe so an, dass sie der womöglich ungenaueren Information des Kontextempfängers entspricht. Dies lässt sich einfach anhand der *generalizationOf*-Relation des Modells durchführen. Tritt dieser Fall ein, schickt *KI* die ausgewählte Repräsentation an *KE* und weist mit einer Flag darauf hin, dass er auch Zugriff auf eine höhere Detailstufe erreichen kann.

- *(4)* Nach Erhalt der Kontextinformatioen von *KI* fügt *KE* diese Daten in seine lokale CoRe-Instanz ein. Wollte *KE* eigentlich detaillierte Informationen über den aktuellen Kontext von *KI* erfahren, kann er eine nächste Runde des Protokolls starten und dabei eine höhere Detailstufe seines eigenen Kontexts anfügen.

- Das Protokoll endet, sobald eine Partei beschließt, auszusteigen, z.B. weil der maximale freigegebene Detailgrad erreicht ist oder weil die bisherigen Antworten schon für die Diensterbringung ausreichen.

Durch die Verwendung dieses Protokolls wird erreicht, dass die Menge und der Detailgrad an Informationen, die zwischen den beteiligten Parteien ausgetauscht werden, sich die Waage halten. Der Kontextempfänger muss dieselben Informationen über sich preisgeben, wie er sie vom Kontextinhaber in Erfahrung bringen möchte. Es lässt sich damit jedoch keine perfekte Symmetrie

erreichen, da jeder Client aufgrund der in der letzten Nachricht enthaltenen Informationen an einer beliebigen Stellte entscheiden kann, aus der Kommunikation auszusteigen. Insbesondere in der ersten Runde muss der Kontextempfänger dabei in Vorleistung gehen und u.U. etwas preisgeben, ohne eine entsprechende Antwort zu erhalten. Die Verwendung mehrerer aufeinanderfolgenden Runden minimiert die dadurch entstehende Asymmetrie jedoch.

Um für den ersten Schritt zu ermitteln, ob und wenn ja, welche Repräsentation des angefragten Kontexttyps dabei in der aktuellen Situation von *KE* an *KI* freigegeben werden darf, simuliert der Privacy Manager von *KE* eine eingehende Kontextanfrage durch *KI*. Dies löst eine Trigger-Auswertung aus, die ggf. in der Erzeugung eines Grants für *KI* resultiert. Falls ein aktiver Trigger Bedingungen an den Kontext von *KI* stellt, wird dessen Auswertung in diesem Spezialfall ausnahmsweise ohne Berücksichtigung dieser Regelteile ausgeführt. Diese Informationen stehen i.d.R. nicht zur Verfügung, weshalb *KE* eine Art informationellen Vorschuss leisten muss, da er Informationen in Erfahrung bringen möchte. Anschließend wird die durch den Grant freigegebene Repräsentation über die Verwendung der *generalizationOf*-Eigenschaft in der Auflösung reduziert und ein detailärmerer *ObfuscationLevel* gewählt. Für die Erzeugung geeigneter Repräsentationen sind die zur Verfügung stehenden Mechanismen zur Kontexterkennung und Verschleierung zuständig. Z.B. lässt sich hierfür die in [255] vorgestellte Ontologie zum Reasoning über verschiedene Detailstufen von Standortdaten einsetzen.

Stellt der Privacy Manager bei einer eingehenden Kontextanfrage fest, dass es sich aufgrund eines nicht bestandenen Plausibilitätstests der nacheinander empfangenen Informationen um einen Angriff handeln könnte, wird der Nutzer darüber zusammen mit der Identität der anfragenden Entität informiert.

3.4.4.2.4 Beispielablauf des Peer-to-Peer-Protokolls Auf das in Kapitel 3.4.3.2.3 verwendete Beispielszenario mit Alice und ihrem Kollegen Bob lässt sich dieses Protokoll wie folgt anwenden: Alice ist bereit, ihren aktuellen Standort während der Arbeitszeit mit ihren Kollegen zu teilen. Wenn sie sich an ihrer Arbeitsstelle befindet, darf diese Information sogar mit Raumgenauigkeit weitergegeben werden, damit ihre Kollegen bei Bedarf genau wissen, wo sie gerade zu finden ist. Sie hat diese detaillierte Freigabe aber davon abhängig gemacht, ob sich der anfragende Kollege auch im Bürogebäude aufhält.

Um in solchen Fällen eine Alternative zum Erraten des initial mitzuliefernden *ObfuscationLevels* durch den Kontextempfänger zu bieten, kann der in den bestehenden Triggern ggf. an die Situation dieses Kontextempfängers jeweils verwendete Level im Vorfeld paarweise ausgetauscht werden.

Dies kann umgesetzt werden ohne die Privatsphäre einer beteiligten Entität zu gefährden, da weder Hinweise auf die aktuelle Situation des Kontextinhabers oder -empfängers noch Details der Regeldefinition verraten werden. Stattdessen wird nur kommuniziert, auf welchem Detailgrad die vom Nutzer angegegeben Bedingungen an den Kontextempfänger definiert wurden.

Angenommen, Bob weiß noch nicht, ob Alice heute überhaupt arbeitet. Aus diesem Grund beginnt er das Protokoll, indem er ihr zunächst seinen eigenen Ortskontext als `workplace` mitteilt und damit wie beschrieben in Vorleistung geht. Da sich Alice ebenfalls im Gebäude befindet, antwortet ihr Privacy Manager mit derselben Information, fügt aber noch ein Flag an, das aussagt, dass dieser Kontexttyp für Bob noch auf einer höheren Detailstufe verfügar ist. Aus diesem Grund tritt Bob in die zweite Runde des Protokolls ein, teilt Alice nun mit in welchem Raum er sich gerade befindet und bekommt in der Antwort ihren exakten Aufenthaltsort mitgeteilt.

Wäre Alice hingegen nicht im Büro, würde sie schon in der ersten Antwort ihre aktuelle Position gemäß den aufgestellten Regeln nur auf Städteebene zurückgeben und keinen Hinweis auf höherwertige Informationen zurücksenden. Bob könnte in diesem Fall an dieser Stelle aus dem Protokoll aussteigen und hätte außer der Information, dass er in der Arbeit ist, nichts verraten.

3.4.4.3 Integration von ALPACA in Android

Die bisherigen Architekturbeschreibungen sind schematischer Natur und zeigen unabhängig von einem darunterliegenden System, wie ALPACA aufgebaut ist. In diesem Abschnitt soll am Beispiel von Android beschrieben werden, wie sich ein solcher Ansatz zur Kontextverwaltung in ein aktuelles mobiles Betriebssystem integrieren lässt.

Die folgenden Betrachtungen basieren auf der theoretischen Auseinandersetzung mit der Architektur dieses Betriebssystems und zeigen, dass sich die vorgestellte Systemarchitektur nah an der Realität mobiler Betriebssysteme bewegt und dass ein solches System auch nachträglich integriert werden kann, ohne die Funktionalität bestehender Anwendungen zu beeinträchtigen. Verwandte Arbeiten aus der jüngsten Vergangenheit belegen zudem – wie in Kapitel 3.3.2.3 beschrieben – insbesondere für ein so offenes System wie Android die praktische Machbarkeit einer solchen Modifikation des Betriebssystems.

Wie in Abb. 3.12 dargestellt, basiert Android auf einem Linux-Kernel und verwendet seit Version 5.0 die Java-Laufzeitumgebung *Android Runtime* (ART), um in Java programmierte Apps auszuführen. Aus Sicherheitsgründen, z.B. um den Zugriff auf Sensoren oder fremde Dateien zu verhindern, bekommt jede Anwendung eine eigene Linux User-ID zugewiesen, läuft in einer dedizierten Sandbox und in einer eigenen Instanz der ART [100].

Abgesehen von den standardmäßigen Java-Bibliotheken liefert Android das sogenannte *Application Framework* mit aus, das Entwicklern viele Basisfunktionalitäten wie Fenster- und Lifecycle-Management ermöglicht. Das Application Framework ermöglicht den Zugriff auf die Kontextdaten, die das Smartphone zur Verfügung hat. Da der Privacy Manager im Rahmen von ALPACA wie beschrieben als Torwächter bzgl. des Zugriffs auf jegliche Kontextinformationen dient, muss er an exakt dieser Stelle platziert werden.

Um die Kontextinformationen des Nutzers effektiv schützen zu können, muss Anwendungen der direkte Zugriff auf die Originalfunktionalität des Applicati-

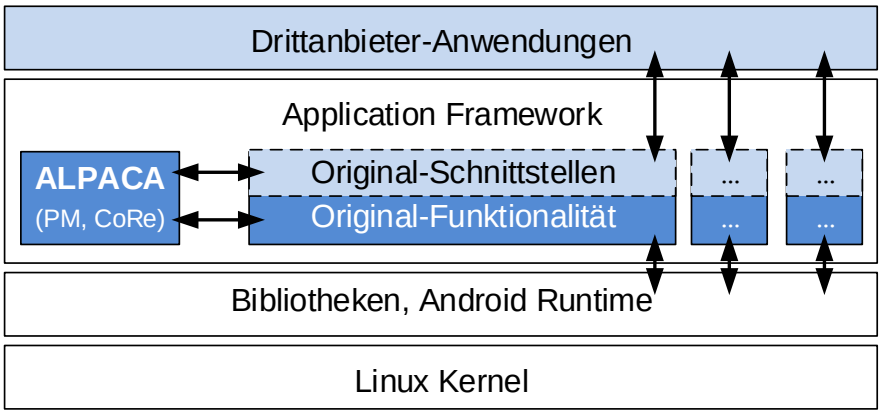


Abbildung 3.12: Transparente Integration des Privacy Managers innerhalb des Android Application Frameworks.

on Frameworks verwehrt werden. Da die Anwendungen jedoch weiterhin auf die standardisierten Programmierschnittstellen der Android API zugreifen können sollen, kann der Privacy Manager für Anwendungen transparent dem Application Framework hinzugefügt werden. Dies wird erreicht, indem die Standardschnittstelle der jeweiligen Androidkomponenten, wie in Abb. 3.12 in hellblau dargestellt, nach außen hin unangetastet gelassen werden, ihre interne Funktionalität allerdings auf die Rolle eines Proxys reduziert wird, der alle eingehenden Anfragen an den Privacy Manager umleitet.

Analog zu dem Berechtigungssystem von Android lässt sich die Identität des Kontextempfängers dabei anhand der entsprechenden User-ID der anfragenden Anwendung ermitteln [101]. Basierend auf den aktuell bekannten Kontextinformationen sowie den Privatsphäreeinstellungen des Nutzers, die im CoRe-Modell bzw. in den definierten Freigaberegeln hinterlegt sind, kann der Privacy Manager entscheiden, welche Repräsentation des angefragten Kontexttyps dieser Kontextempfänger erhalten soll.

Durch die Beibehaltung der Originalschnittstellen des Application Frameworks wird erreicht, dass die Benutzung von Standardanwendungen weiterhin möglich ist. Diese Apps können nicht trivial unterscheiden, ob sie die angefragten Informationen von der Originalimplementierung oder die von ALPACA zugunsten der Privatsphäre u.U. modifizierten Informationen erhalten haben.

Darüber hinaus kann der Privacy Manager auch explizit als neue Komponente des Application Frameworks prominent integriert werden, z.B. um die Entwicklung spezialisierter, kontextbezogener Anwendungen zu unterstützen. Der Privacy Manager stellt in diesem Fall eine umfangreiche Schnittstelle zu den gesamten in CoRe abgelegten Kontextinformationen des Nutzers dar, die weit über die Möglichkeiten des aktuellen Android Betriebssystems hinausgehen, da wie zuvor beschrieben auf unterschiedlichste Repräsentationen von Kontext Bezug genommen werden kann. Auch lassen sich erst damit solche kontextabhängigen Anwendungen umsetzen, bei denen verschiedene

Nutzer Kontextinformationen austauschen und bei denen die Freigabe von Informationen auch vom aktuellen Kontext des Peers abhängen kann.

Für die reine Akquise und Freigabeverwaltung von Kontextinformationen genügt es, den Privacy Manager wie in Abb. 3.12 auf Höhe des Application Frameworks einzubinden. Soll jedoch zudem die nachfolgend beschriebene, automatische Klassifizierung von Anwendungen umgesetzt werden, müssen weitere Vorkehrungen getroffen werden. Zum einen muss der Privacy Manager hierfür in der Lage sein, den Informationsfluss innerhalb einer Anwendung und ggf. zu den Kommunikationsschnittstellen zu überwachen. Zum anderen muss er das Versenden sensibler Daten bei Bedarf auch verhindern können.

Einen vielversprechenden Lösungsansatz hierfür präsentieren Enck et al. mit *TaintDroid* [76]. Für den umfassenden Schutz der Privatsphäre ist ein solches Verfahren unbedingt als komplementär zu den hier vorgestellten Mechanismen anzusehen und sollte daher parallel implementiert und mit den Fähigkeiten von ALPACA zur situationsabhängigen Kontextverwaltung kombiniert werden.

3.4.4.4 Authentifikation von Kontextempfängern

Die potentiellen Empfänger von Kontextinformation können sich in mehreren Aspekten voneinander unterscheiden, z.B. hinsichtlich ihrer Platzierung innerhalb und außerhalb des Betriebssystems, ihrer Vertrauenswürdigkeit oder der Frequenz, mit der sie mit dem Privacy Manager interagieren. Nach diesen Gesichtspunkten sind jeweils unterschiedliche Wege zur Authentifikation dieser Entitäten sinnvoll.

Wird der Privacy Manager wie oben beschrieben transparent in das Application Framework von Android eingebunden, kann für die Identifizierung lokal installierter Anwendungen jeweils die vom Betriebssystem vergebene Linux User-ID verwendet werden, die bei der Kontextanfrage zur Verfügung steht und von Android für die Rechteüberprüfung einer App verwendet wird.

Für Anwendungen, die bewusst mit dem Privacy Manager interagieren, werden darüber hinaus detailliertere Identifikationsmöglichkeiten benötigt. Insbesondere bei dem Einsatz von Peer-to-Peer-basierten kontextabhängigen Anwendungen muss es möglich sein, nicht nur die Anwendung selbst, sondern auch einzelne Nutzer eindeutig identifizieren zu können.

Als Grundlage für die sichere gegenseitige Authentifikation dient die Struktur des dezentralen online sozialen Netzwerks *Vegas* [72]. Die Authentifikation von Kontextempfängern findet dabei auf Basis paarweise ausgetauschter Schlüsselpaare statt, sodass sich der Ursprung einer Kontextanfrage eindeutig über die Verifizierung der angehängten Signatur überprüfen lässt. Da der Grundgedanke von ALPACA ähnlich wie bei Vegas der ist, dass der Kontextinhaber in einer sozialen Beziehung zu seinen Peers steht, wird auch hier das dort beschriebene Konzept eines Out-of-Band-Schlüsselaustauschs verwendet.

Sobald der Privacy Manager des Kontextinhabers die Identität des anfragenden Kontextempfängers festgestellt hat, kann die entsprechende *PeerEnti-*

ty-Instanz in CoRe ermittelt werden und der zuvor beschriebene Prozess der Kontextfreigabe durchgeführt werden. Die Vertraulichkeit der ausgetauschten Daten kann über den zusätzlichen Austausch symmetrischer Schlüssel und entsprechender Verschlüsselungsverfahren wie AES umgesetzt werden.

3.4.4.5 Automatische Eingruppierung von Kontextempfängern

Neben der klaren Ausrichtung auf den Schutz der Privatsphäre ist eine weitere Zielsetzung des hier beschriebenen Systems, den Nutzer nicht mit vermeidbaren Entscheidungsvorgängen zu belästigen. Es ist davon auszugehen, dass wenn sich der Nutzer zu übermäßig vielen manuellen Entscheidungen genötigt fühlt, der Datenschutz am Ende wieder einer unkomplizierten Nutzbarkeit zuliebe hintangestellt wird. Unter diesem Blickwinkel diskutiert dieser Abschitt daher unterschiedliche Möglichkeiten für die Einordnung von Kontextempfängern auf die verschiedenen Modellebenen.

Um die Privatsphäre des Nutzers nicht zu gefährden, lassen sich auf der privaten Ebene automatisch nur solche kontextbezogenen Dienste platzieren, die lokal auf dem Endgerät des Nutzers laufen ohne dabei Daten an dritte Parteien zu senden.

Die Zuordnung einzelner Anwendungen zu dieser Ebene, die den unbeschränkten Zugriff auf alle verfügbaren Informationen erlaubt, kann auf unterschiedliche Arten geschehen. Ein naheliegender Lösungansatz liegt darin, diese Entscheidung dem Nutzer zu überlassen. Dies hat zum einen jedoch den Nachteil, dass dieser viele solcher Entscheidungen treffen muss, von denen einige vielleicht überflüssig sind, z.B., weil eine Anwendung gar keine Kontextinformationen benötigt. Zum anderen weiß der Nutzer nicht, welche Daten eine Anwendung überhaupt verwendet.

Eine mögliche Verbesserung wäre es, die Systemberechtigungen, die eine App anfordert, zu analysieren. Durch diese Permissions wird bereits von Android selbst der Zugriff auf persönliche Daten und einige Sensoren geschützt [101]. Allein auf Basis der geforderten Berechtigungen lässt sich jedoch nicht erkennen, wie eine Anwendung mit persönlichen Informationen umgeht. Nur weil sie z.B. die Berechtigungen `INTERNET` und `ACCESS_FINE_LOCATION` anfordert, ist dadurch noch nicht klar, dass die Standortdaten das Gerät auch tatsächlich verlassen. Auch dieser Ansatz würde also potentiell zur Erkennung von Falschpositiven und unnötigen Nutzerinteraktionen führen.

Einen vielversprechenden Ansatz, der sich für die automatische Eingruppierung von Anwendungen einsetzen lässt, stellt die sog. *Informationsflussüberwachung* dar. In Kombination mit einem Ansatz wie [76] lässt sich die automatische Kategorisierung von Apps wie in Abb. 3.13 dargestellt mit möglichst wenig Nutzerinteraktion gestalten: Bei der Installation wird zunächst jede neue Anwendungen auf der privaten Ebene platziert. Sie erhält somit uneingeschränkt Zugang zu den Kontextinformationen des Smartphones, wird jedoch – da es sich um eine automatische Entscheidung handelt – unter Quarantäne gesetzt.

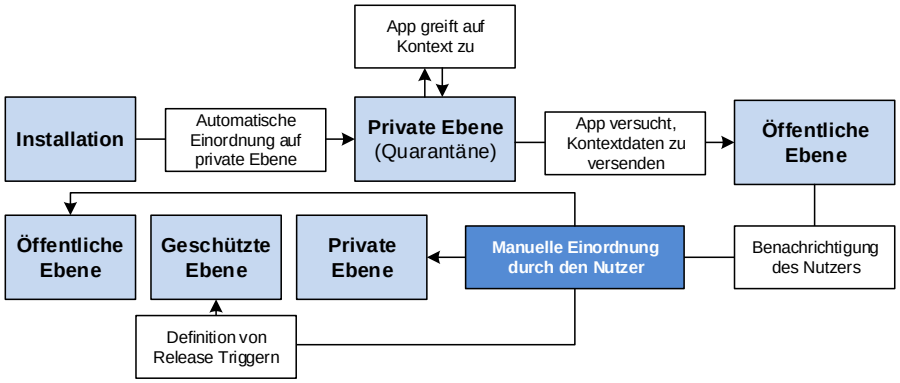


Abbildung 3.13: Ablauf der automatisierten Einordnung neuer Anwendungen unter Verwendung eines Quarantänestatus.

Innerhalb dieses Quarantänestatus unterliegt die Anwendung einer ständigen Überwachung durch TaintDroid. Erkennt der Privacy Manager mit Hilfe des Trackingsystems, dass die Anwendung Kontextinformationen über eine der Kommunikationsschnittstellen versenden will, kann dies unterbunden wurden, z.B. indem die App beendet wird. Da ALPACA grundsätzlich ein restriktives Verhalten bei der Freigabe von Kontextinformationen anstrebt, wird die Anwendung nun automatisch auf die öffentliche Ebene einsortiert.

An dieser Stelle wird erstmals der Nutzer involviert und über die Entscheidung in Kenntnis gesetzt. Dieser kann die Meldung ignorieren, individuelle Freigaberegeln festlegen oder die Anwendung explizit wieder in die private Ebene einordnen. Im Unterschied zu der vorigen Situation wird der Quarantänestatus dadurch jedoch aufgehoben und die Überwachung beendet.

3.4.4.6 Zusammenfassung

In diesem Abschnitt wurde die Systemarchitektur von ALPACA vorgestellt sowie die Aufgaben und Arbeitsweise des zentralen Privacy Managers bei der Akquise und Verwaltung von Kontextinformationen erläutert. Für den symmetrischen Austausch von Kontextinformationen unter gleichberechtigten Peers wurde ein rundenbasiertes Protokoll eingeführt, mit dessen Hilfe sich eine Ausgewogenheit im Informationsfluss zwischen Kontextinhaber und Kontextempfänger erreichen lässt.

Abschließend wurde skizziert, wie sich ALPACA in ein aktuelles mobiles Betriebssystem integrieren lässt und welche Maßnahmen ergriffen werden können, um den Nutzer möglichst selten mit Freigabe-Entscheidungen zu behelligen.

3.5 Qualitative Bewertung von ALPACA

Das in diesem Kapitel präsentierte System zur clientseitigen Verwaltung von Kontextinformationen soll nun im Hinblick auf die in Kapitel 3.2.2 zusam-

mengetragenen Anforderungen bewertet werden. Es werden dabei die Vorteile von ALPACA im Vergleich zu bestehenden Arbeiten herausgearbeitet, aber auch Schwachstellen identifiziert und mögliche Lösungsvorschläge gegeben. Eine kompakte Übersicht über die Erfüllung der privatsphärerelevanten Anforderungen ist in Tabelle 3.1 zu finden.

Anforderung	erfüllt?	Einschränkungen
Kontrolle über persönliche Informationen	•	soweit clientseitig möglich
Situationsabhängige Freigaberegeln	✓	
Vollständigkeit und Konsistenz	✓	
Ausgewogenheit im Informationsfluss	✓	
Allgemeingültigkeit	✓	
Erweiterbarkeit	✓	
Privatsphäre als Querschnittsthema	✓	
Benutzerfreundlichkeit	•	s. Kapitel 3.5.2

Tabelle 3.1: Erfüllung der Privatsphäre-Anforderungen durch ALPACA

3.5.1 Bewertung der technischen Anforderungen

Die Forderung nach der vollständigen Kontrolle über die Verarbeitung persönlicher Informationen durch den Nutzer wird von ALPACA in einem solchen Maße erfüllt, wie sich dies clientseitig und ohne die explizite Kooperation durch den Anbieter eines kontextabhängigen Dienstes umsetzen lässt. In den Metadaten einer Freigaberegel kann der Nutzer z.B. festlegen, dass Zugriffe auf Kontextinformationen durch einen bestimmten Empfänger eine Benachrichtigung verursachen oder geloggt werden. Dadurch kann der Nutzer jederzeit einsehen, welche Informationen ein bestimmter Dienst zu welchen Zeitpunkten von ihm abgefragt hat. Im Gegensatz zu allen TTP-basierten Systemen zur Kontextverwaltung mehrerer Nutzer liegt der Vorteil bei ALPACA darin, dass die Durchsetzung der Freigaberegeln direkt auf dem Endgerät des Nutzers stattfindet anstatt auf einer allwissenden zentralen Komponente.

Was sich im Rahmen der wie hier rein clientseitig konzipierten Kontextverwaltung nicht realisieren lässt, ist den Nutzer stets über Sinn und Zweck sowie die Archivierung und Weitergabe seiner persönlichen Daten in Kenntnis zu setzen oder die Löschung der gespeicherten Daten beim Dienstanbieter zu veranlassen. Lösungsansätze für diese Probleme sind in maschinell interpretierbaren Policy-Systemen wie P3P zu sehen oder in den *Privacy Tags* der *Confab*-Architektur [122], die ähnlich dem digitalen Rechtemanagement Kontextinformationen mit einem Verfallsdatum und Weitergaberestriktionen versehen. Auch Ansätze wie *pawS* [158] und die *Privacy Beacons* [212], bei denen der Nutzer von den Kontextquellen und Sensoren der Umgebung darüber in Kenntnis gesetzt wird, welche Daten über ihn erfasst werden und wie diese weiterverarbeitet werden, lassen sich hierfür adaptieren. Alle Ansätze zur Lösung dieser Anforderung setzen jedoch die Kooperation und Ehrlichkeit

des Kontextempfängers voraus und lassen sich nicht allein auf dem Endgerät des Nutzers umsetzen. Stattdessen sollten sie ergänzend zu der von ALPACA ermöglichten clientseitigen Kontextverwaltung eingesetzt werden.

Auf Basis der modellierten Informationen ermöglicht ALPACA die Definition von situationsabhängigen Freigaberegeln durch den Nutzer. Abhängig vom aktuellen Nutzungskontext, der Identität oder Gruppenzugehörigkeit des Kontextempfängers sowie ggf. dessen eigenen Kontexts lassen sich feingranulare Regeln erstellen, die nicht nur positiv oder negativ ausfallen können, sondern gemäß der Privatsphärebedürfnisse des Kontextinhabers auch zur Preisgabe von unwahren, veralteten oder geeignet verschleierten Informationen an einen bestimmten Empfänger führen können. Zu diesem Zweck wurde der kontextabhängige Triggermechanismus eingeführt, mit dessen Hilfe der Kontextinhaber auf Basis der in CoRe gespeicherten Kontextinformationen flexible und ausdrucksstarke Freigaberegeln festlegen kann. Bei geringerem Funktionsumfang und fehlender Integration der übrigen privatsphärerelevanten Aspekte haben auch [8] und [131] die Regelauswertung anhand eines OWL-basierten Kontextmodells auf mobilen Endgeräten demonstriert. Im Gegensatz zu diesen Arbeiten kann die Definition der Regeln unter ALPACA den gesamten Katalog an kontextabhängigen Einflussfaktoren berücksichtigen, die potentiell Einfluss auf die Privatsphärebedürfnisse des Nutzers haben. Die in Kapitel 3.3.2 beschriebenen verwandten Arbeiten berücksichtigen jeweils nur Teilaspekte davon, aus deren Vereinigung sich der hier verwendete Gesamtkatalog ergeben hat.

Die Forderung nach Vollständigkeit und Konsistenz dieser Freigaberegeln wird einerseits durch die Einteilung der möglichen Kontextempfänger auf die verschiedenen Ebenen gewährleistet, andererseits durch die Konsistenzprüfung des aktuellen Modellzustands. Zu diesem Zweck speichert CoRe die Zuordnung, welche Empfänger in der aktuellen Situation auf welche Kontextrepräsentationen zugreifen dürfen. Auf konzeptioneller Ebene des Modells wurden geeignete Restriktionen eingeführt, um dabei Konfliktsituationen gemäß formalisierter Regeln aufdecken zu können.

Jeder Kontextempfänger ist prinzipiell entweder auf der privaten oder auf der öffentlichen Ebene einsortiert. Gemäß dieser Einordnung kann der Privacy Manager Kontextanfragen auch dann beantworten, wenn der Nutzer keine kontextabhängigen Freigaberegeln für den entsprechenden Kontextempfänger angegeben hat. Insbesondere erhalten Kontextempfänger der öffentlichen Ebene Zugriff auf die vom Nutzer definierten, privatsphäretechnisch unbedenklichen Darstellungsformen des Nutzerkontexts. Existieren darüber hinaus kontextabhängige Regeln für einen bestimmten Empfänger, gelangt dieser in den durch die Trigger definierten Situationen auf die geschützte Ebene und erhält somit Zugang zu Kontextrepräsentationen, die nicht öffentlich einsehbar sind.

Um dabei mögliche Widersprüche in den Regeldefinitionen aufdecken zu können, wurde das Konzept der *Grants* in das CoRe-Modell integriert, mit deren Hilfe formal sichergestellt werden kann, dass stets eine eindeutige Zuordnung von Kontextempfänger zu einer Repräsentation eines Kontexttyps besteht.

Wird hierbei ein Konflikt erkannt, wird der Nutzer um eine manuelle Auflösung gebeten, die für die Zukunft in Form der *precedesOver*-Relation in den Metadaten des überlegenen Triggers vermerkt wird. Um dem Nutzer auch in solchen Konfliktsituationen die Entscheidungshoheit über Privatsphäre und Qualität der kontextabhängigen Diensterbringung zu gewähren, ist dieses Vorgehen als Vorteil gegenüber bestehenden Systemen zu sehen, die z.B. auf pauschale „*deny*"/„*allow*"-Strategien [22] oder die Spezifität von Regeln [206, 34] setzen. Diese Herangehensweisen entsprechen nicht zwangsläufig der eigentlichen Intention des Benutzers, der auf widersprüchliche Regeldefinitionen hingewiesen werden sollte, um diese manuell zu korrigieren oder zu priorisieren.

Die zeitliche Konsistenz der übermittelten Informationen wird durch die Speicherung der in der Vergangenheit bereits an einen Kontextempfänger übermittelten Daten in CoRe und die Integration geeigneter Algorithmen zur Plausibilitätsüberprüfung aufeinanderfolgender Informationen sichergestellt.

Für die Gewährleistung einer Ausgewogenheit im Informationsfluss zwischen zwei Nutzern bei der Verwendung Peer-to-Peer-basierter kontextabhängiger Anwendungen wurde für ALPACA ein rundenbasiertes Kommunikationsprotokoll vorgeschlagen, welches eine technische Umsetzung des in [132] formulierten Prinzip der minimalen Asymmetrie darstellt. Um zu verhindern, dass eine Partei übermäßige viele Informationen über eine andere abfragen kann und somit ein informationelles Ungleichgewicht entsteht, muss der Kontextempfänger dabei stets dieselben Informationen über sich preisgeben, wie er sie vom Kontextinhaber in Erfahrung bringen möchte.

Durch den Einsatz dieses Verfahrens kann ein hoher Grad an Symmetrie hinsichtlich der zwischen Peers ausgetauschten Menge an Informationen erreicht werden. Der Detailgrad der übermittelten Informationen wird dabei wie in Kapitel 3.4.4.2 beschrieben nur schrittweise erhöht. Jede Partei kann aus der Kommunikation aussteigen, z.B. wenn die zuletzt erhaltenen Informationen eine weitere Runde überflüssig machen.

Von den bekannten Systemen zur Kontextverwaltung ist ALPACA somit das einzige, dass eine Lösung für die Herstellung einer Ausgewogenheit im Informationsaustausch zwischen Peers anbietet. In [145] wird diese Anforderung als Ergänzung für die Statusanzeige von Nutzern innerhalb einer Instant Messaging-Anwendung realisiert. Die Umsetzung findet dort jedoch nicht clientseitig statt, sondern basiert auf der Verwendung einer TTP, welche die Statusinformationen eines Nutzers nur an jene Kontakte weiterleitet, die ihren eigenen Status auf sichtbar gestellt haben. Das oft zitierte System in [34] hingegen ermöglicht es dem Kontextinhaber z.B. nicht einmal einzusehen, ob, wann und von wem welche Informationen über ihn abgefragt wurden.

Angesichts der unterschiedlichen Ausprägungen sowohl von Kontextinformationen als auch von kontextbezogenen Diensten wurde die Anforderung der Allgemeingültigkeit der Kontextverwaltung gestellt. Die von ALPACA eingesetzte ontologiebasierte Modellierung von Kontextinformationen entspricht dabei der in der Literatur am häufigsten zu findenden Herangehensweise und

ermöglicht die Modellierung unterschiedlichster Typen von Kontextinformationen und deren Metadaten sowie die logische Inferenz auf den modellierten Informationen. In Kapitel 3.4.4.2 wurde beschrieben, wie sich ALPACA für die Nutzung unterschiedlicher Ausprägungen (visionärer) kontextabhängiger Anwendungen einsetzen lässt, die entweder zentralisiert oder Peer-to-Peer-basiert umgesetzt sein können. Zudem wurde in Kapitel 3.4.4.3 aufgezeigt, wie sich der Privacy Manager für bestehende Android-Anwendungen transparent in das Betriebssystem einbinden lässt, um personalisierten, kontextabhängigen Drittanbieter-Anwendungen situationsabhängig genau die Kontextinformationen zukommen zu lassen, die der Nutzer möchte.

Im Unterschied insbesondere zu den in Kapitel 3.3.2.1 beschriebenen Systemen zur TTP-basierten Kontextverwaltung lässt sich ALPACA damit für alle auf einem Endgerät installierten Anwendungen verwenden, die auf Kontextinformationen des Nutzers zugreifen und nicht für jeweils einen Dienst und dessen Broker-Komponente. Um eine generische Verschleierung verschiedenster Typen von Kontextinformationen zu ermöglichen, wurden zudem unter den in Kapitel 2.2.2 vorgestellten Verschleierungmechanismen jene identifiziert und in CoRe integriert, die sich z.B. wie die Herausgabe von veralteten oder als unzuverlässig gekennzeichneten Informationen allgemeingültig anwenden lassen.

Die Weiterentwicklung sowohl von mobilen Endgeräten als auch von Methoden zur Kontexterkennung und -verschleierung schreitet stetig voran. Vor diesem Hintergrund wird an ein System zur Verwaltung von Kontextinformationen die Forderung nach möglichst flexibler Erweiterbarkeit gestellt. Beim Entwurf der Systemarchitektur von ALPACA in Kapitel 3.4.4 wurde daher ein modularer Aufbau verfolgt, der sich einfach um entsprechende Komponenten ergänzen lässt. Die zentrale Komponente stellt der Privacy Manager dar, der das Kontextmodell und die Freigaberegeln des Nutzers verwaltet. Diese Schicht trennt die Kontextquellen von den Kontextempfängern, welchen der Privacy Manager jeweils passende Schnittstellen zur Verfügung stellt, um Informationen in das Kontextmodell zu schreiben oder daraus abzufragen.

Die auf dem Endgerät zur Verfügung stehenden Sensoren, Algorithmen zur Kontexterkennung und zur Verschleierung von Kontextinformationen gehören bewusst nicht zum Kern des präsentierten Systems, sondern werden vom Privacy Manager entsprechend der vom Nutzer aufgestellten Regeln orchestriert. So lässt sich z.B. ein neues Verfahren zur Verschleierung eines bestimmten Kontexttyps einfach integrieren, indem dieses nachträglich installiert und ein entsprechender *ObfuscationLevel* in CoRe hinzugefügt wird.

Erstellt der Nutzer nun eine Freigaberegel, in der dieser Verschleierungsmechanismus zum Einsatz kommt, aktiviert der Privacy Manager die neue Komponente und überführt dessen Ergebnisse in das Kontextmodell, um sie auf Anfrage dem entsprechenden Kontextempfänger zur Verfügung zu stellen. Somit können auch bei der für eine Dienstnutzung notwendigen Preisgabe von Informationen komplexe Schutzziele verfolgt werden, die sich mit generischen, bereits in CoRe integrierten Verschleierungstechniken nicht umsetzen lassen.

Im gesamten Systementwurf von ALPACA werden privatsphärerelevante Aspekte als Querschnittsthema berücksichtigt. Der Forderung aus [193], Mechanismen zum Schutz der Privatsphäre nicht erst auf Anwendungsebene, sondern auch in jeden relevanten Teilschritt der Kontextermittlung und -verwaltung zu integrieren, wird Rechnung getragen: Als exklusive Schnittstelle zu den Sensoren und Kontextinformationen des Endgeräts überwacht der Privacy Manager die Privatsphärepräferenzen des Nutzers. Sowohl die Akquise und Erzeugung von Kontextinformationen (über eine kontextabhängige Blacklist) als auch die Herausgabe dieser Daten an dritte Parteien (über eine kontextabhängige Whitelist) findet gemäß dieser individuellen Regeln statt.

Das für die formale Modellierung des Nutzerkontexts vorgestellte CoRe-Modell wurde so konzipiert, dass alle für die Erstellung feingranularer Freigaberegeln nötigen (Meta-)Informationen zur Verfügung stehen, mit deren Hilfe sich z.B. Aktualität und Detailgrad von herausgegebenen Informationen festlegen lassen. Zudem sieht sowohl die Systemarchitektur als auch das Kontextmodell die Integration spezialisierter Verschleierungstechniken für bestimmte Typen von Kontextinformationen vor, die als jeweils eigener *ObfuscationLevel* im Modell referenziert werden können.

Um je nach Situation und Vertrauenswürdigkeit unterschiedlicher Kontextempfänger maßgeschneiderte Informationen ausgeben zu können, werden in CoRe parallel unterschiedliche Repräsentationen des Nutzerkontexts gespeichert, die über den vorgestellten Trigger-Mechanismus kontextabhängig einem bestimmten Empfänger zugeordnet werden können. Im Gegensatz zu den bekannten Ansätzen nimmt CoRe nicht nur die funktionale Kontextmodellierung vor, sondern modelliert auch privatsphärerelevante Aspekte. Hierfür wurde das Modell derart mit Restriktionen hinsichtlich der Verknüpfbarkeit von Klassen, Relationen und Individuen versehen, dass auch die Konsistenz der kontextabhängigen Regeldefinitionen auf Basis des Modells zum Zeitpunkt der Regelauswertung validiert werden kann. Die Privatsphärebedürfnisse eines Nutzers werden von ALPACA somit bei jedem Teilschritt der kontextabhängigen Diensterbringung berücksichtigt.

3.5.2 Flexibilität versus Benutzerfreundlichkeit

Aufgrund der konservativen Herangehensweise bei der Verwaltung von Kontextinformationen geht der Einsatz von ALPACA unweigerlich mit einem erhöhten Konfigurationsaufwand für den Nutzer einher: Jede Information, die der Nutzer mit anderen Parteien zu teilen bereit ist, muss explizit freigegeben und ggf. noch genauer charakterisiert werden. Dies steht der Forderung nach einem hohen Maß an Benutzerfreundlichkeit grundsätzlich entgegen. Während sich z.B. die Studienteilnehmer in [159] unter Laborbedingungen zwar für die Erstellung individueller Regeln entschieden haben, ist es fraglich, ob eine Mehrheit der Nutzer gewillt oder dazu in der Lage ist, whitelistbasiert und situationsabhängig über die Freigabe von Informationen zu entscheiden. Neben einem

grundsätzlichen Interesse an Themen wie Datenschutz und Privatsphäre setzt dies zudem voraus, dass der Nutzer dazu bereit ist, sich aktiv darüber Gedanken zu machen, in welchen Situationen er welche Informationen über seinen aktuellen Kontext freigeben möchte. Viele Nutzer werden sich für den Charme einer unkomplizierten Dienstnutzung entscheiden, wenn die Alternative in der manuellen Definition von Freigaberegeln zu sehen ist. Gleichzeitig zeigen Arbeiten wie [24] deutlich, wie wichtig die Ausdrucksstärke bei der Definition von Freigaberegeln ist, um die oft sehr individuellen Privatsphärebedürfnisse eines Nutzers abbilden zu können.

Es wurden aus diesem Grund verschiedene Mechanismen in ALPACA integriert, um diese Hürden für den Nutzer möglichst klein zu halten und ihn nur bei Bedarf in die Entscheidungsfindung einzubinden. Als Beispiel hierfür kann die Einführung der drei verschiedenenen Ebenen angesehen werden, in die sich Kontextinformationen und -empfänger einteilen lassen, sowie der Quarantänestatus, in den Apps bei der Installation zunächst versetzt werden ohne dass der Nutzer hierbei eingreifen muss. Somit werden keine unnötigen Entscheidungen vom Benutzer verlangt, sondern nur solche, die Regeln für Kontextempfänger festlegen, die von ALPACA anhand ihrer Weitergabe von Kontextinformationen an die Netzwerkschnittstelle als echte Gefahr für die Privatsphäre des Nutzers klassifiziert werden.

Ein weiterer Aspekt, über den der Nutzer wie in Kapitel 3.4.3.3 argumentiert aus Sicht der Privatsphäre stets selbst entscheiden muss, ist die Auflösung widersprüchlicher Freigaberegeln. Auch hierfür wurde durch die dynamische Erkennung solcher Konflikte zur Anfragezeit dafür gesorgt, dass der Nutzer nur bei tatsächlichem Auftreten solcher Fälle eine Entscheidung fällen muss.

Wie von Bokhove et al. [35] gefordert, lassen sich mit ALPACA auch mit wenig Aufwand für den Benutzer unterschiedliche Grundeinstellungen für unterschiedliche Benutzertypen anlegen. Ein unbesorgter Nutzer kann alle Anwendungen auf die private Ebene einordnen, um jeden kontextabhängigen Dienst für eine optimale und unkomplizierte Nutzungserfahrung mit allen Informationen auszustatten, die dieser anfragt. Gleichermaßen kann ein nur auf Privatsphäre bedachter Nutzer, der sicherstellen möchte, dass keinerlei Kontextinformationen preisgegeben werden, auf die Angabe jedweder Freigaberegeln verzichten und alle Anwendungen somit der öffentlichen Ebene zuweisen und somit nichtssagende Default-Werte zurückgeben.

Zwischen diesen beiden extremen Ausprägungen ermöglicht ALPACA die Angabe beliebig komplexer, situations- und rezipientenabhängiger Freigaberegeln. Bereits heute schalten viele privatsphärebewusste Nutzer die GPS-Ortung an ihrem Smartphone ab und deaktivieren das WLAN-Modul, wenn sie unterwegs sind. Es bleibt abzuwarten, wie sich die Bereitschaft der Nutzer entwickelt, sich aktiver mit Themen wie Datenschutz und Privatsphäre auseinanderzusetzen, wenn neben den heute schon flächendeckend verfügbaren ortsbezogenen Diensten auch kontextbezogene Anwendungen populär werden und noch unmittelbarer persönliche Kontextinformationen wie die aktuelle Aktivität, der

Fitness-, Gesundheits- oder Gemütszustand standardmäßig und kontinuierlich zur Verfügung stehen.

Aus heutiger Sicht können mehr Typen von Kontextinformationen auf dem mobilen Endgerät ermittelt werden, als für die kontextabhängige Diensterbringung an externe Parteien preisgegeben werden müssen. Gleichzeitig war vor gut zehn Jahren – also unmittelbar vor der Einführung des ersten iPhones – auch nicht absehbar, mit welch großem Appetit mobile Anwendungen den Standort ihrer Nutzer abfragen [76]. Visionäre soziale Anwendungen, wie das in [164] skizzierte *MoodSharing* deuten jedoch an, dass es prinzipiell für fast alle Typen von Kontextinformationen Anwendungsfelder gibt, die einen Austausch dieser Informationen mit externen Parteien nötig machen.

Der *Privacy Manager* ist selbst eine kontextabhängige Anwendung, die lokal auf dem Endgerät des Nutzers läuft. Wie verschiedene Studien [142, 30, 265] gezeigt haben, spielen unterschiedlichste Aspekte der eigenen Situation eine große Rolle für die Freigabe von Kontextinformationen wie dem aktuellen Standort. Mit Hilfe von ALPACA lassen sich nicht nur alle Typen von Kontextinformationen den Privatsphärebedürfnissen des Nutzers entsprechend verwalten, sondern es können auch alle zur Verfügung stehenden Kontextinformationen für die kontextabhängige Freigabe berücksichtigt werden.

Die vorliegende Arbeit präsentiert mit ALPACA somit die technische Grundlage, um die individuellen Privatsphärebedürfnisse eines Nutzers bei der Verwaltung seiner persönlichen Kontextinformationen kontextabhängig, effektiv und flexibel durchzusetzen. Um den Benutzer bei der Formulierung entsprechender Freigaberegeln zu unterstützen bzw. um diesen Vorgang teilweise zu automatisieren, wurden in der Litaratur verschiedene Ansätze vorgestellt: So kann z.B. auf Basis des kollaborativen Filterns, das häufig für die Umsetzung personalisierter Empfehlungssysteme eingesetzt wird [55, 75], versucht werden, die Privatsphärepräferenzen eines Nutzers anhand eines Ähnlichkeitsvergleichs mit anderen Teilnehmern eines Systems vorherzusehen.

In Abhängigkeit von Informationen über die Semantik von Orten (Wohnsitz, Einkaufen, Essen, Universität, etc.) und der Tageszeit (Morgen, Mittag, Abend, Nachmittag, Nacht) untersuchen Zhao et al. ob sich mit Hilfe des kollaborativen Filterns nutzerübergreifend Empfehlungen für die kontextabhängige, binäre Freigabe von Standortinformationen erzeugen lassen [265]. In dem simplifizierten Szenario gelingt eine korrekte Vorhersage in 73 % der Fälle.

Xie et al. [256] gehen einen Schritt weiter und beziehen in ihren Betrachtungen zusätzlich den aktuellen Gemütszustand des Kontextinhabers sowie die soziale Beziehung zu verschiedenen Kontextempfängern mit ein. Die Empfänger werden dabei den Gruppen Familie, Freunde und Kollegen zugewiesen, für die unterschiedliche Regeln definiert werden können. In diesem komplexeren Beispiel bestätigt sich zum einen der Bedarf an ausdrucksstarken Freigaberegeln, zum anderen erreichen die Autoren bei der Empfehlung entsprechender Regeln immerhin noch Erfolgsraten um die 60 %.

Ein Nachteil solcher Verfahren ist, dass die Privatsphärepräferenzen anderer Nutzer bekannt sein müssen, was ein Datenschutzproblem darstellt. Um dieses Problem zu vermeiden, schlagen Bigwood et al. [30] den Einsatz maschinellen Lernens vor, um auf Basis von beobachteten Freigabeentscheidungungen korrekte Vorhersagen für zukünftige Situationen zu treffen. Mit Klassifikationsergebnissen von bis zu 83 % bei der Verwendung der Ensemble-Lerntechnik *Rotation Forests* [201] stellt dies einen vielversprechenden Ansatz dar, der jedoch eine umfangreiche Trainingsphase benötigt, in der der Nutzer alle Entscheidungen manuell treffen muss. Die Suche nach Möglichkeiten zur benutzerfreundlichen Formulierung von kontextabhängigen Freigaberegeln stellt somit ein wichtiges Forschungsfeld für zukünftige Arbeiten dar.

Zusammenfassend stellt ALPACA einen allgemein einsetzbaren Ansatz zur feingranularen, situations- und rezipientenabhängigen Verwaltung von Kontextinformationen auf einem mobilen Endgerät dar. Der Funktionsumfang geht deutlich über die aus Literatur und Praxis bekannten Systeme hinaus. Es werden alle für die Nutzung kontextbezogener Dienste nötigen Teilschritte berücksichtigt und hinsichtlich privatsphärerelevanter Aspekte optimiert, von der Akquise von Kontextinformationen über deren Modellierung und Inferenz, bis hin zur letztendlichen Weitergabe an einen Kontextempfänger.

Der vorgestellte Ansatz kann für aktuelle mobile Betriebssysteme eingesetzt werden, um Benutzern flexible Einstellungsmöglichkeiten für den individuellen Schutz ihrer Privatsphäre an die Hand zu geben. Hierfür werden ausdrucksstarke Freigaberegeln ermöglicht, die auf alle zur Verfügung stehenden Kontextinformationen Bezug nehmen können und die feingranulare Herausgabe einzelner Daten erlauben. Die Erkennung von widersprüchlichen Freigaberegeln wird in Echtzeit durchgeführt und der Nutzer somit nur bei tatsächlichem Bedarf um die Auflösung solcher Konfliktsituationen gebeten.

Zudem bietet ALPACA durch den Einsatz des in Abschnitt 3.4.4.2 eingeführten, rundenbasierten Kommunikationsprotokolls als einziges unter den bekannten Systemen zur Kontextverwaltung eine Lösung, die ein gewisses Maß an Symmetrie beim Austausch von Kontextinformationen zwischen Peers bietet. Der Preis für diesen hohen Grad an Allgemeingültigkeit, Ausdrucksstärke und Flexibilität ist jedoch in der Komplexität zu sehen, die bei der Erstellung situationsabhängiger Freigaberegeln u.U. auf den Nutzer zukommt.

3.6 Zusammenfassung

In diesem Kapitel wurde mit ALPACA ein umfassender Ansatz für die privatsphärezentrische Verwaltung von Kontextinformationen auf einem mobilen Endgerät vorgestellt. Zu diesem Zweck wurden verschiedene Ansätze zum Schutz der Privatsphäre in kontextbezogenen Anwendungen aus der Literatur zusammengetragen und zu einer Lösung integriert.

Im ersten Schritt wurde eine intuitive Abstraktion der Privatsphärebedürf-

nisse eines Nutzers entworfen. In den folgenden Abschnitten wurden technische Lösungen zur Umsetzung einer daran angelehnten Kontextverwaltung präsentiert. Hierfür wurde zunächst ein ontologiebasiertes und auf den Schutz der Privatsphäre ausgerichtetes Kontextmodell entworfen. Der Modellentwurf unterstützt die Einhaltung aus Privatsphäresicht unbedingt notwendiger Restriktionen, z.B. um die Herausgabe widersprüchlicher Informationen an denselben Kontextempfänger zu verhindern. Zur Umsetzung der situationsabhängigen Freigabe von Kontextinformationen an verschiedene Empfänger wurde aufbauend auf dem Kontextmodell ein kontextbezogener Trigger-Mechanismus vorgestellt. Hiermit lassen sich abhängig vom eigenen Kontext sowie ggf. dem des Empfängers flexible Freigaberegeln erstellen, deren Konsistenz zur Laufzeit überprüft wird. Im nächsten Schritt wurde die Systemarchitektur von ALPACA vorgestellt sowie der Kommunikationsablauf bei der Anfrage von Kontextinformationen durch eine dritte Partei gezeigt. Am Beispiel von Android wurde gezeigt, wie sich die Kontextverwaltung mit ALPACA nahtlos und bei Bedarf transparent in ein modernes mobiles Betriebssystem einfügt.

Das vorgestellte System bietet eine ausdrucksstarke, stark formalisierte und funktionale Grundlage für die clientseitige Umsetzung einer privatsphärezentrischen Kontextverwaltung. Die Erstellung geeigneter Freigaberegeln bleibt jedoch eine zeitaufwändige und komplexe Aufgabe, die den Nutzer womöglich überfordert. Die Komplexität entsteht dabei aus dem grundsätzlichen Wunsch, kontextabhängige Dienste mit ansprechender Qualität zu nutzen, ohne dabei Informationen preiszugeben, die der Nutzer individuell als privat oder einem bestimmten Empfänger gegenüber als unangemessen einstuft.

Ein vielversprechender, zur Regeldefinition orthogonaler Lösungsansatz besteht deshalb in der Entwicklung intelligenter Verschleierungstechniken, die mit minimalem Konfigurationsaufwand selbständig dazu in der Lage sind, die Privatsphäre des Nutzers einem objektiven Ziel entsprechend zu schützen.

Stehen solche Verfahren zur Verfügung, muss der Nutzer nicht mehr für unterschiedliche Situationen festlegen, wer in welcher Form auf seine Informationen zugreifen darf. Stattdessen kann er sich einmalig für ein bestimmtes Schutzziel entscheiden, das ihm z.B. gegenüber einem bestimmten Dienstanbieter wichtig ist und das fortan automatisch eingehalten wird.

Für die spezielle Ausprägung der ortsbezogenen Anwendungen werden im nächsten Kapitel Mechanismen vorgestellt, die genau diese Herangehensweise für die unkomplizierte, privatsphäreschonende Nutzung kontextabhängiger Dienste verfolgen und zudem versuchen, die erreichbare Dienstqualität trotz des Einsatzes einer effektiven Standortverschleierung zu optimieren.

4 Privatsphäre in ortsbezogenen Diensten

Im vorangegangenen Kapitel wurde ein generischer Ansatz zur Kontextverwaltung auf mobilen Endgeräten vorgestellt, der sich für verschiedenste Ausprägungen von Kontext und kontextabhängigen Diensten einsetzen lässt. Der Nutzer kann damit situationsabhängig festlegen, ob und mit wem bestimmte Informationen geteilt werden dürfen. Um den Detailgrad preisgegebener Informationen auch bei der Nutzung kontextabhängiger Dienste auf ein den Privatsphärebedürfnissen des Nutzers angemessenes Maß zu begrenzen, kann neben binären Freigabeentscheidungen zudem festgelegt werden, wie diese verschleiert werden sollen. Dies stellt jedoch eine komplexe Aufgabe dar, da hierfür informierte Entscheidungen hinsichtlich des durch einen bestimmten Verschleierungsmechanismus erreichbaren Grades an Privatsphäre sowie eine Abwägung bzgl. Datenschutz und Servicequalität erfolgen müssen. Um den Nutzer bei diesem nicht trivialen Vorgang zu unterstützen, werden maßgeschneiderte Verschleierungstechniken benötigt, welche die Kontextinformationen vor der Herausgabe an einen Empfänger geeignet, d.h., dem jeweiligen Kontext- und Anwendungstyp entsprechend, verfälschen.

Die am häufigsten genutzten kontextabhängigen Anwendungen sind ortsbezogene Dienste. Für die privatsphäreschonende Nutzung derartiger Dienstangebote werden im Folgenden clientseitige Verfahren vorgestellt, die nur das Endgerät des Nutzers als vertrauenswürdig erachten und sich direkt mit heute verfügbaren, kommerziellen Diensten nutzen lassen.

Ein stets auf aktuelle und präzise Standortdaten angewiesener Vertreter derartiger Dienste ist die verkehrsadaptive Ermittlung schnellster Routen mit Hilfe von online Routenplanern. Für dieses konkrete Einsatzszenario wird untersucht, wie sich der Ortskontext eines Benutzers unter Berücksichtigung eines Angreifers mit umfangreichem Kartenwissen sinnvoll verschleiern lässt und welche Auswirkungen der Einsatz solcher Verfahren auf die Nutzungserfahrung und Qualitätseigenschaften derartiger Dienste hat.

Für die Erhöhung der Effizienz des Systems werden unterschiedliche Optimierungen bei der kartenbasierten Verschleierung von Routenanfragen vorgeschlagen und hinsichtlich der auftretenden Trade-Offs evaluiert. Aus den hierbei gewonnenen Erkenntnissen ergibt sich schließlich ein neues Verfahren für die kontinuierliche, karten- und topologiebezogene Standortverschleierung, die sich wieder generisch für unterschiedliche LBS-Typen einsetzen lässt.

4.1 Vorveröffentlichungen

Teile des nachfolgend beschriebenen Ansatzes wurden bereits in einem internationalen Konferenzbeitrag [68] vorveröffentlicht. Es handelt sich dabei insbesondere um die Textabschnitte, die den Aufbau des Basissystems PrOSPR beschreiben sowie die einfachen Heuristiken für die Auswahl von Anfragepunkten. Die dazugehörige Evaluation wurde für die vorliegende Arbeit um zusätzliche Aspekte erweitert und auf Basis aktuellen Kartenmaterials neu durchgeführt.

Originär im Rahmen der vorliegenden Arbeit werden u.a. intelligente, topologiebezogene Heuristiken für die Auswahl von Dummy-Punkten und alternative Herangehensweisen bei der Routenvervollständigung eingeführt. Zudem werden ein neues Verfahren für die topologiebezogene Erzeugung von Verschleierungszonen sowie, darauf aufbauend, Strategien zur Herausgabe von Positionsdaten für verschiedene Ausprägungen ortsbezogener Dienste entwickelt.

4.2 LBS, Routenplanung und Privatsphäre

Die online Routenplanung stellt eine der populärsten Ausprägungen ortsbezogener Dienste dar, deren Nutzung längst zum privaten und beruflichen Alltag gehört. Unter Berücksichtigung der aktuellen Verkehrslage ermitteln entsprechende LBS die derzeit schnellste Route $P_{S \to D}$ sowie die dazugehörige Fahrzeit $t_{S \to D}$ von einem Startpunkt S zu einem Ziel D. Letzteres wird dabei i.d.R. in Form einer menschenlesbaren Adresse angegeben wie *Oettingenstr. 67, 80538 München*.

Für die folgenden Überlegungen wird o.B.d.A. davon ausgegangen, dass die Zieladresse des Nutzers bereits in dieser Form vorliegt. Der Startpunkt einer Routenanfrage kann entweder ebenfalls auf diese Weise angegeben werden oder durch die vom GPS-Modul des mobilen Endgeräts automatisch ermittelten WGS84-Koordinaten gegeben sein.

Im Gegensatz zu offline Navigationssoftware zeichnen sich online Routenplaner dadurch aus, dass sie schnellste Routen auf Basis tagesaktuellen Kartenmaterials und – was den größten Vorteil darstellt – mit Echtzeit-Verkehrsdaten, Stau-, Baustellen- und Umleitungsinformationen berechnen.

Aus Sicht der Privatsphäre werden bei der Verwendung eines online Routenplaners mehr personenbezogene Informationen preisgegeben als bei der Nutzung anderer ortsbezogener Dienste. So enthält die Anfrage typischerweise nicht nur den aktuellen Standort eines Nutzers, sondern auch die exakte Beschreibung seines nächsten Aufenthaltsorts. Selbst wenn der Startpunkt einer Routenanfrage nicht mit dem aktuellen Standort übereinstimmt, beschreibt dieser Punkt mit hoher Wahrscheinlichkeit eine Adresse, an der sich der Nutzer entweder in Kürze aufhalten wird oder an der er – weil dieser Ort in irgendeiner Form für ihn wichtig ist – sogar regelmäßig anzutreffen ist. Analog verhält es sich mit dem Ziel einer Routenanfrage, das dieselben personenbezogenen Informationen transportiert.

Im Vergleich zu anderen Typen ortsbezogener Dienste kommt hinzu, dass der Routenplaner für die von ihm erwartete Dienstqualität stets mit präzisen Ortsinformationen versorgt werden muss: Während eine um wenige hundert Meter verzerrt formulierte POI-Suche nach „*Apotheken in meiner Nähe*" aufgrund quasi wegfallender Bandbreitenbeschränkungen abhängig von der jeweiligen POI-Dichte brauchbare Ergebnisse liefert, kann ein Versatz der GPS-Position um nur wenige Meter (z.B. auf die falsche Fahrtrichtung einer Autobahn) bei einer Routenanfrage erhebliche und offensichtliche Fehler verursachen.

Angesichts der nachfolgend zusammengefassten Eigenschaften kann ein online Routenplaner somit als überdurchschnittlich komplexer Vertreter ortsbezogener Dienste angesehen werden – sowohl was die eigentliche Diensterbringung betrifft als auch hinsichtlich der Anforderungen an Mechanismen, die den Schutz der Privatsphäre im Rahmen der LBS-Nutzung gewährleisten sollen:

- Die Suche nach interessanten Orten in der Nähe des Nutzers kann durch die euklidische Distanz zweier Punkte auf einer Freifläche berechnet werden. Demgegenüber muss die Antwort auf eine Routenanfrage als kürzester Pfad innerhalb eines gerichteten, gewichteten Graphs ermittelt werden, der das Straßennetz unter Berücksichtigung aller Fahrgebote, Geschwindigkeitsbegrenzungen, Verkehrsbehinderungen, etc. modelliert.

- Im Gegensatz zu anderen LBS enthält eine Anfrage an einen online Routenplaner nicht nur den aktuellen Standort des Nutzers, sondern auch die exakte Angabe seines nächsten Ziels. Um die Privatsphäre der Standortdaten eines Nutzers effektiv zu schützen, müssen daher beide Endpunkte einer Routenanfrage wirkungsvoll verschleiert werden.

- Schon geringe Ungenauigkeiten bei der Angabe von Start oder Ziel können in Form von erheblichen Umwegen zu einer drastischen Verschlechterung der Dienstqualität führen. Übliche Verschleierungsmechanismen, die z.B. wie [6] auf einer zufälligen, geometrischen Translation von Koordinaten basieren, können deshalb nicht angewendet werden [78].

- Anders als andere LBS-Ausprägungen beruht die verkehrsadaptive online Routenplanung nicht allein auf statischen Karteninformationen und POI-Datenbanken, sondern auf volatilen Echtzeitdaten über die aktuelle Verkehrslage. Dies erhöht den Kommunikations- und Rechenaufwand des Dienstanbieters. Zudem lassen sich deshalb bekannte Techniken zum Schutz der Privatsphäre in ortsbezogenen Diensten wie z.B. das Caching von Antworten [4] nicht anwenden.

In den folgenden Abschnitten soll das gewählte Szenario daher aus unterschiedlichen Gesichtspunkten betrachtet werden: Den Anfang bilden Angriffe, die auf Basis der im Rahmen der Routenplanung übermittelten Standortinformationen möglich sind. Im Anschluss werden die algorithmischen Grundlagen der verkehrsadaptiven Routenplanung erläutert sowie darauf aufbauend die Fähigkeiten des Angreifers und das angestrebte Schutzziel dargelegt.

4.2.1 Angriffe auf Basis von Standortinformationen

Wie bei jedem pseudonym genutzten LBS ist auch der Anbieter einer online Routenplanung dazu in der Lage, seine Nutzer durch simple Archivierung der eingehenden Anfragen und Standortinformationen über die Zeit hinweg zu verfolgen und somit eine detaillierte Historie der von ihnen aufgesuchten Orte anzulegen. Im Folgenden wird eine Übersicht über mögliche Angriffe gegeben, die sich aus der Erhebung bzw. dem Besitz solcher Daten ergeben.

Dass eine solche anbieterseitige Speicherung von Nutzerdaten im Rahmen von Navigationsanwendungen erfolgt, ist meist den Nutzungsbedingungen entsprechender Dienste zu entnehmen. Ein reales Negativbeispiel dafür, wie solche Daten durch einen Dienstanbieter monetarisiert werden, ist z.B. der Datenskandal des niederländischen Anbieters TomTom. Dieser wurde 2011 dabei ertappt, die gespeicherten Fahrten seiner Nutzer an die niederländische Regierung und Polizei zu verkaufen, die daraus eine optimale Platzierung von Blitzer-Anlagen ableitete [113].

Ein neugieriger Dienstanbieter kann aus den gesammelten Daten jedoch noch weitere Schlüsse über die meist nur pseudonym bekannten Nutzer ziehen: Verschiedene Angriffe zielen dabei auf die De-Anonymisierung von Benutzern, die Profilerstellung oder die kontinuierliche Verfolgung und Vorhersage des aktuellen Standorts ab [79].

4.2.1.1 De-Anonymisierung von Benutzern

Mit Hilfe sogenannter Inferenz-Attacken (engl. *inference attacks*) ist jeder, der in Besitz derartiger Daten ist, u.U. dazu in der Lage, die regelmäßigen Aufenthaltsorte eines Nutzers wie z.B. sein Zuhause oder seine Arbeitsstelle zu identifizieren. Wie Krumm zeigt, lässt sich dies sogar mit relativ einfachen Mitteln erreichen [148]. So kann das Zuhause eines Nutzers einfach anhand des am häufigsten oder des zuletzt an einem Tag aufgesuchten Ortes ermittelt werden. Für den Arbeitsplatz existieren naturgemäß ähnliche Gesetzmäßigkeiten zu anderen Tageszeiten.

Krumm, der für seine Analyse die aus Fahrzeugen gesammelten GPS-Tracks von 172 Nutzern verwendet, gelingt es zwar nur in 5% der Fälle korrekt auf den Namen der Person zu schließen, hat dabei allerdings auch mit drei Problemen zu kämpfen. Zum Ersten unterliegen die verwendeten GPS-Messungen einer gewissen Ungenauigkeit, was eine zuverlässige Ermittlung der zugehörigen Adresse erschwert. Um dieses Problem in den Griff zu bekommen, schlagen diverse Arbeiten aufwändigere Methoden zur Extraktion wichtiger Orte aus GPS-Tracks vor, wie z.B. zeit- oder dichtebasiertes Clustering [137, 268, 269, 129]. Zum Zweiten hat sich sowohl der verwendete inverse Geocoder, mit dessen Hilfe von WGS84-Koordinaten auf Adressen geschlossen werden sollte, als auch die zur Verfügung stehende Adressdatenbank als sehr lückenhaft erwiesen. Zum Dritten stellen viele Personen ihr Fahrzeug nicht unmittelbar vor ihrem Haus ab, sondern auf einem freien Parkplatz in der Nähe.

In der Arbeit von Krumm führt allein der letzte Punkt zwangsweise dazu, dass das Wohnhaus des Nutzer an einer falschen Adresse vermutet wird.

Diese Schwachstellen bei der Identifizierung von Individuen entfallen jedoch bei der Nutzung eines online Routendienstes. So wird der Dienst anstelle von womöglich verrauschten GPS-Messungen im Rahmen einer Routenanfrage direkt mit den exakten Start- und Zieladressen versorgt, wodurch zudem der u.U. fehlerbehaftete Geocoding-Schritt entfällt. Das Problem, dass eine Route nicht zwangsweise direkt vor der Zieladresse endet, gilt für den Anbieter eines Routendienstes ebenfalls nicht: Bereits durch die Zieleingabe einer Routenanfrage ist die relevante Adresse unmittelbar bekannt. Unabhängig davon, ob der Nutzer am Ende ein wenig abseits parken muss oder nicht, kennt der Dienstanbieter längst die eigentliche Zieladresse. In seiner vom Autor betreuten Masterarbeit [74] hat Werner Eckert anhand eines mehrwöchigen Experiments mit echten Nutzern gezeigt, dass dieser Nachteil auch entfällt, wenn anstelle des Fahrzeugs das Smartphone des Nutzers als Messinstrument dient.

Aus Sicht des Anbieters eines online Routingdienstes ist demnach eine sehr viel höhere Trefferrate bei der De-Anonymisierung seiner Nutzer zu erwarten – in der Studie von Krumm würden sich somit lediglich die 13% der Teilnehmer, die in einem Mehrparteienhaus leben, nicht mehr eindeutig identifizieren lassen. Auch in solchen Fällen ist die Identifizierung von Personen unter Berücksichtigung weiterer Informationen jedoch eindeutig möglich – z.B., wenn neben deren Privatadresse zusätzlich der Ort ihrer Arbeitsstelle bekannt ist.

Dies zeigen Golle und Partridge in [97], indem sie die statistische Verteilung von Adresspaaren untersuchen, die Privatanschrift und Arbeitsplatz von mehr als 100 Mio. amerikanischen Arbeitnehmern beschreiben. Ihre Datenquelle ist eine synthetisch erstellte Übersicht über die täglichen Pendlerbewegungen, die vom statistischen Bundesamt der USA veröffentlicht wurde. Diese Daten geben zwar nicht die tatsächlichen Orte wieder, haben aber dieselbe statistische Verteilung wie die nicht offen zugänglichen Originaldaten. Die Autoren kommen zu dem Ergebnis, dass sich erst durch eine Verschleierung auf County-Ebene eine eindeutige Identifizierung zuverlässig verhindern lässt. Für LBS, die wie die online Routenplanung auf präzise Standorte angewiesen sind, stellt eine solche Vergröberung auf Landkreis-Ebene freilich ein unbrauchbares Vorgehen dar. Im Durchschnitt entstehen bei einer derart groben Verschleierung riesige *Anonymity Sets* mit einer Größe von knapp 35.000 Personen.

4.2.1.2 Erstellung von Persönlichkeitsprofilen

Während die wichtigen Orte eines Nutzers geschützt werden müssen, um eine Identifizierung zu verhindern, gibt es noch weitere Angriffe, die auf Basis der gesammelten Standortinformationen durchgeführt werden können. Sog. *Profiling Attacks* nutzen alle von einem Benutzer besuchten Orte und zielen nicht unmittelbar auf die De-Anonymisierung ab, sondern werden für die automatische Extraktion weiterer persönlicher Informationen genutzt [79].

Angriffe dieser Kategorie basieren auf der Tatsache, dass verschiedene Adressen meist mit einer eindeutigen Semantik versehen sind wie z.B. ein Krankenhaus oder sich wie ein Wohnhaus einer stark begrenzten Anzahl an Personen zuordnen lassen [258].

Durch Kenntnis der von einer Person besuchten Adressen lassen sich daher Lebensstil, Konsumverhalten, politische Meinungen und Religionszugehörigkeit ableiten. Der Besuch bei Fachärzten ermöglicht den Rückschluss auf gesundheitliche Beschwerden, Krankheitsbilder und -verläufe, usw. [83]. Vor diesem Hintergrund transportieren auch Adressen, die nur unregelmäßig oder einmalig beobachtet werden, aussagekräftige Hinweise auf verschiedene Eigenschaften einer Person und lassen sich daher zur Erstellung eines detaillierten Persönlichkeits- und Interessenprofils verwenden.

Darüber hinaus können durch die unbedachte Verwendung von online Routenplanern auf dieselbe Weise auch im beruflichen Umfeld ungewollt Informationen preisgegeben werden. Die Hausbesuche eines Arztes zeigen, wer bei ihm in Behandlung ist, die Termine von Versicherungsvertretern offenbaren die Kunden der Versicherung, Geschäftspartner verraten, zu welchen anderen Unternehmen neue Beziehungen aufgebaut werden, etc.

4.2.1.3 Echtzeitverfolgung von Benutzern

Eine dritte Angriffsform besteht in der kontinuierlichen Verfolgung eines Nutzers (engl. *tracking*) sowie der darauf basierenden Vorhersage, wo er sich zu einem bestimmten Zeitpunkt aufhält oder aufhalten wird – selbst wenn er den Dienst aktuell nicht nutzt. In diesem Zusammenhang können z.B. Scellato et al. mittels nicht-linearer Zeitreihenanalyse [141] der früheren Besuche wichtiger Orte durch einen Benutzer seinen nächsten Aufenthaltsort sowie die ungefähre Ankunftszeit und Aufenthaltsdauer mit Erfolgsraten von bis zu 90% vorhersagen [210].

Der Grund für die überaus erfolgreiche Durchführbarkeit solcher Vorhersagen ist in den Regelmäßigkeiten menschlicher Mobilität zu sehen [98], die sich individuenübergreifend und unabhängig von Alter, Geschlecht, Bevölkerungsdichte und Wohnort beobachten lassen [224]. Im Rahmen der vorliegenden Arbeit spielt diese Angriffsform jedoch nur eine untergeordnete Rolle, da sich aus Sicht des Dienstanbieters kaum Vorteile durch die ständige Verfolgung eines Nutzers ergeben. Für andere Angreifertypen, z.B. Stalker, ist dies von größerer Bedeutung.

Im Fall eines LBS-Anbieters erscheinen die ersten beiden, auf Identifikation und Profilerstellung ausgerichteten Angriffe deutlich interessanter, weil diese rein maschinell umsetzbar und direkt monetarisierbar sind. Das nachfolgend vorgeschlagene Verfahren zur privatsphäreschonenden Umsetzung von online Routenanfragen nimmt sich daher insbesondere des Problems an, eine eindeutige Exktraktion der Ziele und Aufenthaltsorte eines Nutzers durch den Dienstanbieter zu verhindern. Naturgemäß wird dadurch zugleich die aus der

Beobachtung von Routenanfragen ableitbare Kenntnis und Vorhersagbarkeit der exakten Aufenthaltsorte eines Nutzers auf Adressebene verhindert.

Ohne geeignete Gegenmaßnahmen ist es dem Anbieter eines online Routenplaners also möglich, durch simple Archivierung aller Anfragen eines Nutzers an eine Vielzahl von persönlichen Informationen über ihn zu gelangen. Die korrekte Zuordnung einzelner Routenanfragen zu einem Benutzer wird durch die Tatsache erleichtert, dass der Benutzer dem Dienstanbieter in der Regel über ein Pseudonym bekannt ist und sich verschiedene Routenanfragen somit trivial zu einem Profil kombinieren lassen. Solche Informationen können dazu dienen, ein detailliertes Bewegungs- und Persönlichkeitsprofil eines Nutzers zu erstellen, denn sie geben Auskunft über regelmäßig besuchte Orte, Interessen und Vorlieben des Nutzers sowie u.U. sogar seine Identität.

In diesem Kapitel wird ein neues Verfahren zur privatsphäreschonenden Nutzung der online Routenplanung vorgestellt. Die Gefahren der Identifizierung, Profilerstellung und genauen Verfolgbarkeit von Nutzern können damit auf eine individuell einstellbare Wahrscheinlichkeit reduziert werden, während weiterhin eine qualitativ hochwertige Dienstnutzung ermöglicht wird.

4.2.2 Grundlagen der online Routenplanung

In diesem Abschnitt wird kurz auf das der Routenplanung zugrundeliegende mathematische Problem sowie auf aktuelle Umsetzungen verkehrsadaptiver Routingverfahren eingegangen. Eine Übersicht über den aktuellen Stand der Technik bzgl. statischer, verkehrsadaptiver und verkehrsmittelübergreifender Routingmechanismen, die dabei relevanten Designentscheidungen und den daraus jeweils resultierenden Trade-Offs findet sich z.B. in [20].

Algorithmisch gesehen, handelt es sich bei der Ermittlung der schnellsten Route zwischen einem Startpunkt S zu einem Zielpunkt D um das Finden eines entsprechenden Pfades P in einem gerichteten Graph $G(V, E)$, der das Straßennetz modelliert. Die Routenplanung betrachtet das *Point-to-Point-Shortest-Path*-Problem, das versucht, den Pfad zwischen S und D in G zu ermitteln, der das niedrigste Gesamtgewicht w aller am Pfad beteiligten Kanten aufweist. Im Falle verkehrsadaptiver Routenplaner ist dabei insbesondere die Fahrtzeit $t_{S \to D}$ von Interesse, da sich diese dynamisch ändert und die wichtigste Unterscheidung zu offline Karten darstellt. Die Kanten E des Graphen modellieren Straßensegmente, die z.B. mit der realen, nicht-negativen Länge des Segments l als Kantengewicht sowie der erlaubten Fahrtrichtung markiert sind und zudem mit verschiedenen Metainformationen wie der zulässigen Maximalgeschwindigkeit annotiert sein können. Die Knoten V des Graphen stellen Anschlussstellen zweier Strassensegmente, Abbiegemöglichkeiten oder Kreuzungen dar, die mindestens zwei unterschiedliche Kanten miteinander verbinden.

Dass zwei Kreuzungspunkte im realen Straßennetz durch eine beidseitig befahrbare Straße physisch miteinander verbunden sind, bedeutet jedoch nicht, dass diese Verbindung stets benutzt werden darf. So schränken z.B. Abbie-

geverbote, die an einer Kreuzung aus einer gewissen Fahrtrichtung kommend gelten, abhängig von der zuletzt besuchten Kante die Menge an möglichen Folgekanten ggf. ein. Solche komplexen Eigenschaften des Straßennetzes lassen sich z.B. durch die Duplizierung bzw. das Splitting von Knoten und dem Einfügen entsprechender Kanten zu Nachbarknoten auf eine reguläre Graphdarstellung abbilden [213]. Durch diese Transformation des mit komplexen Abbiegeverboten versehenen Straßennetzes lassen sich schließlich Standardverfahren zur Ermittlung kürzester Wege in einem Graphen einsetzen wie der Algorithmus von Dijkstra [61] oder der A*-Algorithmus [112].

Einige Routingverfahren nehmen eine mehrere Minuten bis Tage dauernde Vorverarbeitung (engl. *pre-processing*) des Kartenmaterials in Kauf, um selbst auf kontinentaler Ebene die Antwort auf eine online Routenanfrage innerhalb weniger Nanosekunden bereitstellen zu können. Zu diesem Zweck werden in diesem Schritt z.B. „virtuelle Abkürzungen" durch Vorberechnung ausgewählter Teilrouten in den Graph eingefügt [56] oder Hierarchien unterschiedlich komplexer Graphen aufgebaut, die der natürlichen Ordnung von Straßen (Autobahn, Bundes-, Land- und Wohnstrassen, etc.) nachempfunden sind [209, 88]. Je kürzer die Antwortzeit, desto länger dauert meist die Preprocessing-Phase und desto größer fällt der Speicherbedarf des vorberechneten Graphen aus [20]. Im Fall verkehrsadaptiver Routingdienste sind die Kantengewichte jedoch dynamisch und müssen laufend der Verkehrslage angepasst werden, sodass ein zeitaufwändiges Preprocessing hier nicht anwendbar ist.

Die verkehrsadaptive Echtzeit-Planung von schnellsten Strecken, wie Google, TomTom und andere sie anbieten, verlässt sich längst nicht mehr auf die Vorhersage der aktuellen Verkehrslage auf Basis historischer Daten oder auf die Beobachtungen punktuell installierter Messstationen. Solche Strategien haben sich als zu ungenau herausgestellt und werden – maßgeblich unterstützt durch die immense Verbreitung und technologische Weiterentwicklung mobiler Endgeräte und dem Ausbau der mobilen Kommunikationsnetze – durch bessere Alternativen ersetzt [107]. So werden heute die von der Masse an Nutzern gemessenen Live-Daten dazu verwendet, Verkehrshindernisse wie Unfälle, Staus und überlastete Streckenabschnitte in Echtzeit zu erkennen [233] und geeignete Umfahrungen anbieten zu können [102]. Aufgrund dieser Herangehensweise sind diese Dienste zu überaus präzisen, hochauflösenden Aussagen bezüglich der aktuell vorherrschenden Verkehrslage imstande und bieten somit einen klaren Mehrwert gegenüber der rein auf statischem Kartenmaterial basierenden Offlinenavigation, aber auch gegenüber älteren, verkehrsadaptiven Systemen wie dem *Traffic Message Channel* (TMC) [81].

Um eine Balance zwischen der Dauer des Preprocessings und der individuellen Antwortzeit auf eine Routenanfrage zu erreichen, können für die Integration dieser Echtzeitdaten in den Routingmechanismus verschiedene Strategien verfolgt werden: So wird die Preprocessing-Phase von Delling et al. in zwei Schritte geteilt, wobei sich die verkehrsadaptive Aktualisierung sehr effizient umsetzen lässt und der zeitaufwändige, Metrik-unabhängige Teil nur einmalig

ausgeführt werden muss [57]. In [58] setzen die Autoren darauf, die Wiederholung des Preprocessings ganz zu vermeiden und die Verkehrsadaptivität in die Logik der Anfragebearbeitung zu integrieren, was jedoch zu Lasten der individuellen Antwortzeiten geht [20].

Für die benutzerfreundliche Visualisierung der schnellsten Route ist natürlich nicht nur deren Dauer, sondern v.a. auch deren Verlauf von Interesse. Im Rahmen der Routenplanung findet daher eine geographische Einbettung der Graphknoten in ein entsprechendes Koordinatensystem statt. Darüber hinaus wird eine Geocoding-Komponente benötigt, die auf Basis einer Adressdatenbank die Zuordnung von menschenlesbaren Adressen auf z.B. WGS84-Koordinaten ermöglicht, die der Routenplaner zum Ermitteln des schnellsten Weges schließlich auf die nächstgelegene Kante im Strassengraph mappt.

Aus der Summe dieser Funktionalitäten ermöglichen online Routenplaner die qualitativ hochwertige, verkehrsadaptive Ermittlung schnellster Strecken anhand von Punkt-zu-Punkt-Anfragen. Die Nutzer müssen den Dienstanbieter hierfür jedoch mit exakten Angaben bezüglich ihres aktuellen Standorts bzw. der Start- und Zieladresse ihrer Anfrage versorgen. Wie eingangs beschrieben, handelt es sich bei diesen Informationen jedoch um schützenswerte, persönliche Daten, deren Herausgabe an Dritte eine unmittelbare Gefährdung der Privatsphäre darstellen kann. Die sich hieraus ergebende Problemstellung, Annahmen über Identität und Fähigkeiten des Angreifers sowie weitere Anforderungen an die privatsphäreschonende Umsetzung der online Routenplanung werden im Folgenden genauer spezifiziert.

4.2.3 Problemstellung und Ziel der Verschleierung

Angesichts der unterschiedlichen Angriffsmöglichkeiten, die sich auf Basis präziser Standort- und Zielinformationen für die Privatsphäre eines Nutzers ergeben, soll der Dienstanbieter insbesondere daran gehindert werden, die einzelnen Aufenthaltsorte seiner Benutzer anhand beobachteter Routenanfragen exakt nachvollziehen zu können. Aus dem Besitz derartiger Informationen ergibt sich wie beschrieben die Gefahr der De-Anonymisierung sowie die Möglichkeit zur Erstellung detaillierter Persönlichkeitsprofile durch Dritte. Tauchen dieselben Adressen wiederholt und zu typischen Tageszeiten in den Routenanfragen eines Benutzers auf, kann daraus geschlossen werden, dass es sich dabei um einen wichtigen Ort wie z.B. sein Zuhause oder seine Arbeitsstelle handelt.

Die entsprechende Klassifizierung lässt sich automatisch und selbst durch vergleichsweise einfache Mittel erreichen [148]. So wäre es der niederländischen Polizei z.B. möglich, aus den von TomTom erhaltenen Daten [113] auf die Identitäten der Fahrer zu schließen, um Geschwindigkeitsüberschreitungen nachträglich zu ahnden. Neben der Identifizierung auf Basis wichtiger Orte können Adressen, die sporadisch oder einmalig vorkommen, mit Hilfe von Adress- und Unternehmensverzeichnissen zu Bekanntenkreis und Lebensstil des Nutzers sowie zu von ihm besuchten Geschäften, Ärzten, etc. aufgeschlüsselt werden [258].

Ermöglicht werden all diese Angriffe durch die Tatsache, dass verschiedene Orte stets mit einer bestimmten Bedeutung oder einer sehr begrenzten Zahl an unterschiedlichen Interpretationsmöglichkeiten verbunden sind. Gruteser et al. bezeichnen Standortdaten, die sich einem abgrenzbaren räumlichen Bereich wie z.B. einem Wohngebäude zuordnen lassen, daher als *restricted space information* [104]. Diese beschreiben Orte, die in ihrer Zugänglichkeit eingeschränkt (engl. *restricted*) sind und somit unmittelbare Rückschlüsse auf die Personen zulassen, die sich dort aufhalten. Im Rahmen dieser Arbeit wird zusätzlich die Semantik verschiedener räumlicher Einheiten berücksichtigt, woraus sich die folgende Definition *privatsphäresensibler Orte* ergibt:

Definition Privatsphäresensibler Ort:
Ein *privatsphäresensibler Ort* ist ein räumlich-geographischer Bereich, der eindeutig identifiziert werden kann und dem sich ein Besitzer, eine Besitzergruppe oder eine semantische Bedeutung eindeutig zuordnen lässt.

Bei Adressangaben, wie sie bei der Routenplanung verwendet werden, handelt es sich somit just um eine weit verbreitete Darstellungsform solcher privatsphäresensibler Orte. Ziel der vorliegenden Arbeit ist es, die Identifikation solcher Orte aus den für die Dienstnutzung zwangsläufig an einen LBS-Anbieter übertragenen Anfragen zu verhindern. Im Rahmen der online Routenplanung kann dies erreicht werden, indem sowohl Start- als auch Zieladresse einer Routenanfrage in passend zusammengestellten Verschleierungsmengen verborgen werden. Diese Vorgehensweise stellt eine Adaption des von Sweeney entwickelten Prinzips der k-Anonymität [229] dar, das für die datenschutzkonforme Publikation von Patientendaten formuliert wurde (vgl. Kapitel 2.2.2).

Dieses Konzept wurde bereits mehrfach für die Herstellung von Kommunikationsanonymität (engl. *query anonymity*) in ortsbezogenen Diensten verwendet [104, 177, 87, 92, 136]: Durch Rückgriff auf eine TTP als sog. *Anonymizer* versuchen diese Ansätze unter der Annahme, dass der nicht vertrauenswürdige Dienstanbieter ohnehin die genauen Aufenthaltsorte aller Nutzer kennt, den wahren Absender einer LBS-Anfrage unter k Teilnehmern zu verbergen. Hierfür wird unterschiedlichen Strategien folgend der Ursprungsort einer Anfrage so vergrößert oder deren Zeitpunkt derart verzögert, dass diese von k Benutzern stammen könnte. Nachteilig an diesen Verfahren sind u.a. die Abhängigkeit von der TTP sowie von der Verfügbarkeit einer ausreichend hoher Teilnehmerzahl in der Nachbarschaft des anfragenden Nutzers. Auch die Verzögerung von Anfragen stellt aus Sicht der Benutzerfreundlichkeit für viele LBS-Ausprägungen keine gangbare Lösung dar.

Im Rahmen dieser Arbeit wird angestrebt, den Einsatz einer TTP zu vermeiden. Auch soll angesichts der beschriebenen Möglichkeiten zur Identifizierung von Benutzern anhand ihrer Standortdaten nicht die Anonymisierung der Kommunikation erreicht werden, sondern die effektive Verschleierung der in Routenanfragen in Form von Start- und Zieladresse enthaltenen Orte.

Im Gegensatz zu vielen anderen Arbeiten, die sich mit dem Schutz von Standortinformationen in LBS beschäftigen (vgl. Kapitel 2.2.3), soll bei der Verschleierung zudem das Wissen des Angreifers um die Verteilung privatsphäresensibler Orte berücksichtigt werden. Nachfolgend wird davon ausgegangen, dass es sich bei jeder Adresse um einen solchen Ort handelt. Aus Sicht der Privatsphäre kann dies als konservative Interpretationsweise angesehen werden, da einem durch eine Adresse gegebenen Gebäude i.d.R. mindestens ein Besitzer bzw. mindestens eine semantische Bedeutung zugeordnet werden kann. Gleichzeitig impliziert dies die Berücksichtigung von Kartenwissen bei der Verschleierung von Standortinformationen. Dieser Arbeit liegt somit dasselbe Verständnis hinsichtlich der Privatsphäre von Standortinformationen zugrunde wie dem *k-Area*-Verfahren von Gruteser und Liu [106] oder dem *Silent Zone*-Ansatz von Wiesner et al. [253], die beide die Erzeugung von Verschleierungszonen vorschlagen, die k Gebäude beinhalten. Xue et al. formalisieren dieses Verständnis, indem sie das ebenfalls aus der Datenbankforschung bekannte Prinzip der l-Diversität [170] als sog. *location diversity* auf ortsbezogene Dienste übertragen und fordern, dass als Ursprung jeder LBS-Anfrage mindestens l verschiedene Orte in Frage kommen müssen [258].

Diese Arbeiten unterscheiden sich jeweils in den Algorithmen zur Erzeugung entsprechender Verschleierungszonen und Einsatzszenarien: Sowohl [106] als auch [253] nutzen die erzeugten Zonen, um die Standortpreisgabe innerhalb dieser Zonen vollständig zu unterbinden. Keiner dieser Ansätze stellt eine Lösung für die privatsphäreschonende Umsetzung von Routenanfragen dar, für die sowohl der Startpunkt als auch das Ziel der Anfrage gleichermaßen geschützt und an den Dienstanbieter übermittelt werden müssen. In Anlehnung an die etablierten Konzepte der k-Anonymität und Ortsdiversität gilt es für dieses spezielle Einsatzszenario daher, für Start- und Zieladresse angemessene Verschleierungszonen zu erzeugen, um darauf aufbauend die Umsetzung von *k-immunen Routenanfragen* zu ermöglichen.

Definition k-immune Routenanfragen:
Eine Anfrage an einen online Routenplaner ist *k-immun* gegenüber Inferenzangriffen auf Basis privatsphäresensibler Orte, wenn sich sowohl die Startadresse als auch die Zieladresse der Routenanfrage nicht von jeweils mindestens $k-1$ anderen semantischen Orten unterscheiden lassen.

Diese Anforderung lässt sich ohne den Einsatz einer TTP und unabhängig von der Verfügbarkeit weiterer Teilnehmer in der Umgebung des anfragenden Nutzers umsetzen. Für das Szenario der online Routenplanung, in dem der Nutzer stets eine unmittelbare Antwort auf seine Anfragen erwartet, stellt dies somit eine praktikable Herangehensweise dar, die durch den Einbezug von Kartenwissen dennoch effektiven Schutz gegenüber Angriffen wie die De-Anonymisierung und Profilerstellung bietet.

Die Wahrscheinlichkeit, Start- oder Zieladresse einer Routenanfrage eindeu-

tig identifizieren zu können, liegt damit jeweils maximal bei $\frac{1}{k}$. Diese Definition ermöglicht somit ein den Privatsphärebedürfnissen des Nutzers entsprechendes, individuelles Maß an Verschleierung, das ohne komplizierte Regelangaben durch den Nutzer selbständig, effektiv und konsequent bei der Nutzung entsprechender Dienste durchgesetzt werden kann.

4.2.3.1 Angreiferbeschreibung, Verhalten und Fähigkeiten

Wie in der Literatur üblich [63, 181, 104, 177, 87, 92, 136] wird auch im Rahmen der vorliegenden Arbeit davon ausgegangen, dass der Anbieter des ortsbezogenen Dienstes selbst eine Partei darstellt, welcher der Nutzer nicht vollständig vertraut. Der LBS-Anbieter wird somit als passiver, nicht aktiv bösartig agierender Angreifer angesehen, der seinen Dienst korrekt erbringt und nicht versucht, den Nutzer z.B. durch gezielt manipulierte Antworten in irgendeiner Form zu benachteiligen. Auch unternimmt der LBS-Anbieter keine aktiven Schritte, um an zusätzliche Informationen über seine Nutzer zu gelangen, wie dies z.B. im Rahmen der *observation identification* [104] oder bei dem Versuch, einen Nutzer physisch anzutreffen [168] der Fall wäre.

Er verhält sich seinen Nutzern gegenüber jedoch neugierig und versucht auf Basis der beobachteten Anfragen, mehr über deren Identität und Alltag sowie über ihre Interessen in Erfahrung zu bringen. Zusätzlich wird vorausgesetzt, dass die beobachteten Routinganfragen die einzige Quelle von Standortinformationen für den LBS-Anbieter über seine Nutzer darstellen. Zu Vermeidung von Angriffen über mögliche Seitenkanäle wie die geografische Rückverfolgbarkeit von IP-Adressen wird deshalb angenommen, dass vom Nutzer geeignete Gegenmaßnahmen wie z.B. die Verwendung zuverlässiger *Onion-Routing*-Protokolle [62] ergriffen werden.

Im Gegensatz zu [258] wird zudem ausdrücklich angenommen, dass der LBS-Anbieter dazu in der Lage ist, die einzelnen Anfragen eines Benutzers miteinander in Verbindung zu bringen. Verschiedene Teilnehmer sind für den Dienstanbieter demnach einfach unterscheidbar und über eindeutige Pseudonyme bekannt – entweder explizit in Form eines Benutzernamens oder implizit durch die Verwendung geräte- oder browserspezifischer Identifikatoren. Wie verschiedene Arbeiten gezeigt haben, stellen auch bereits Teilmengen der von einem Nutzer besuchten Orte selbst Quasi-Identifier dar [148, 97, 179, 168]. Durch die bloße Archivierung von Routenanfragen ist der Dienstanbieter daher dazu in der Lage, eine Historie der von einem Nutzer besuchten Adressen anzulegen.

Darüber hinaus besitzt der LBS-Anbieter naturgemäß umfangreiches Kartenwissen über die Verteilung und Erreichbarkeit privatsphäresensibler Orte. Naive, z.B. auf geometrischer Translation basierende Formen der Verschleierung von Standort- und Zielinformationen sind daher u.U. wirkungslos, da der Dienstanbieter große Teilbereiche der Karte unmittelbar ausschließen kann. Es gilt daher eine kartenbasierte Form der Verschleierung zu verwenden, um effektive Schutzmaßnahmen gegen einen derartigen Angreifer zu gewährleisten.

Neben dem LBS-Anbieter selbst kann gemäß dieser Beschreibung natürlich

auch jede andere Partei, die z.B. durch Überwachung der Kommunikation oder durch Kollaboration mit dem Routenplaner an die Routenanfragen eines Nutzers gelangt, gleichermaßen als Angreifer betrachtet werden.

4.2.3.2 Weitere Anforderungen und Annahmen

Im Folgenden wird davon ausgegangen, dass der Nutzer die schnellste Route $P_{S \to D}$ im Straßennetz von S nach D von einem heute verfügbaren, kommerziellen online Routenplaner anfragen möchte. Angesichts der unterschiedlichen Metriken und Echtzeitinformationen, die bei der verkehrsadaptiven Routenplanung zum Einsatz kommen können, entspricht die Antwort des Routingdienstes dabei nicht zwangsläufig der streckenmäßig kürzesten Verbindung von S nach D, sondern – unter Berücksichtigung aller dem Dienst bekannten Informationen – der aktuell als zeitmäßig am kürzesten erkannten Route.

Zur Formulierung der Routenanfrage an den online Dienst verwendet der Nutzer sein mobiles Endgerät. Dieses kann den Startpunkt der Routenanfrage mit Hilfe des eingebauten GPS-Empfängers in Form von WGS84-Koordinaten automatisch zur Verfügung stellen. Zudem bietet der Routendienst dem Nutzer über eine grafische Benutzerschnittstelle die Möglichkeit zur manuellen Eingabe von Start- und Zieladresse. Im Rahmen der vorliegenden Arbeit wird davon ausgegangen, dass der Nutzer diese Adressen vollständig angibt, d.h., es werden Straße, Hausnummer, Postleitzahl und Ort von ihm eingetragen. Die privatsphärekonforme Beschaffung unbekannter Adressen wird im Rahmen dieser Arbeit nicht weiter untersucht. Stattdessen wird angenommen, dass der Nutzer die Adressen kennt, offline ermittelt oder geeignete Verfahren zur privatsphäreschonenden Suchmaschinenennutzung wie z.B. *h(k)-PIR* nutzt [64].

An das Endgerät des Nutzers werden abgesehen von einer mobilen Datenverbindung und einem geringen Maß an Rechenleistung keinerlei Anforderungen gestellt. Insbesondere wird nicht vorausgesetzt, dass es von vornherein mit detaillierten Karten- oder Geocodinginformationen ausgestattet ist. Stattdessen strebt das hier vorgestellte Verfahren eine spontane Einsatzfähigkeit an, sodass entsprechende Daten unter Beibehaltung aller Anforderungen an den Schutz der Privatsphäre ad-hoc vom Dienstanbieter akquiriert werden müssen.

Eine weitere Bedingung besteht darin, dass sich die privatsphäreschonende Umsetzung von online Routinganfragen allein auf Basis der standardmäßig verfügbaren Anfrage-Schnittstellen existierender Dienstangebote realisieren lassen muss. So wird im Folgenden nicht wie bei der Umsetzung anderer Verfahren vom Dienstanbieter erwartet, dass dieser kooperiert und spezielle Änderungen an der Funktionalität seiner Schnittstellen vornimmt. Auch soll, wie bereits erwähnt, der Einsatz einer TTP vermieden werden, sodass das Endgerät des Nutzers die einzige vertrauenswürdige Instanz darstellt. Die Verschleierung von Standort- und Zielinformationen muss daher – konsistent zum vorangehenden Kapitel – bereits unmittelbar auf dem Endgerät des Nutzers stattfinden, ohne präzise Kontextinformationen an jegliche fremde Partei weiterzugegeben.

Zu guter Letzt gilt es selbstverständlich eine Form der privatsphäreschonenden online Routenplanung zu entwickeln, die unter Durchsetzung der Privatsphärepräferenzen des Nutzers eine qualitativ hochwertige Dienstnutzung ermöglicht, um eine Akzeptanz entsprechender Verfahren zu erreichen. Es müssen die auftretenden Trade-Offs zwischen dem erreichbaren Maß an Privatsphäre, dem Verlust an Dienstqualität und dem durch die Verschleierung entstehenden Kommunikationsoverhead identifiziert und optimiert werden.

Für das hier untersuchte Einsatzszenario der online Routenplanung bedeutet dies insbesondere, dass die unter dem Einsatz der Verschleierung erzielten Routenergebnisse nicht merklich von der optimalen Route abweichen dürfen und kein übermäßig großer, zusätzlicher Kommunikationsaufwand entstehen sollte. Zudem soll ein möglichst großer Anteil der Route tatsächlich verkehrsadaptiv berechnet werden, d.h., dass die Strecke, die durch clientseitige Vervollständigung „blind" ergänzt werden muss, möglichst minimal sein soll.

4.3 Verwandte Arbeiten

Ein umfassender Überblick über Mechanismen zum Schutz der Privatsphäre in LBS wurde bereits in Kapitel 2.2.3 gegeben. Eine häufig getroffene Annahme ist dabei, dass die Aufenthaltsorte von Nutzern aus Privatsphäresicht relevanter sind als die Wege zwischen diesen Zielen. Dies liegt daran, dass die Orte mit bestimmten Semantiken versehen sind, die eine Identifizierung und Profilbildung des Nutzers ermöglichen [263, 127, 148, 97, 269, 178, 83, 85, 82, 186, 232]. Zudem empfiehlt es sich, Karteninformationen bei der Erstellung von Verschleierungszonen oder Dummy-Endpunkten einzubeziehen, da sonst unrealistische Positionen entstehen, die von einem Angreifer unmittelbar ausgeschlossen werden können oder Zonen, die nicht den Grad an Schutz bieten, den man sich von ihnen verspricht [251].

Im Folgenden werden einige Verfahren vorgestellt, die im direkten Zusammenhang mit den Inhalten dieses Kapitels stehen.

4.3.1 Bann-Zonen

Auf Basis einfacher Clustering-Verfahren und Heuristiken auf den gesammelten GPS-Trajektorien hunderter Nutzer untersucht Krumm in [148], mit welcher Genauigkeit sich die Privatadressen von Nutzern aus diesen Daten ermitteln lassen. Dies gelingt aufgrund des Versuchsaufbaus, bei dem die GPS-Geräte in Fahrzeugen montiert wurden, die nicht immer unmittelbar vor dem Wohnhaus geparkt wurden, immerhin in knapp 13% der Fälle. Bei der Nutzung von Smartphones als Messinstrument entfällt diese Ungenauigkeit [74].

Vor dem Hintergrund derartiger *Inferenz-Angriffe* mittels Standortinformationen wird davon ausgegangen, dass Personen eher dazu bereit sind, ihre aktuelle Position preiszugeben, wenn dadurch keine Aufenthalte an privatsphäresensiblen Orten verraten werden. Der Autor schlägt daher verschiedene Schutz-

maßnahmen vor, um die genaue Inferenz von Orten, die ein Nutzer besucht, zu unterbinden. Aus Sicht der Datenqualität ist der Einsatz sog. *Ban Zones* am vielversprechendsten. Dabei wird um die sensiblen Orte eines Nutzers ein Kreis mit Radius R gelegt, dessen Mittelpunkt zufällig um die eigentliche Adresse verschoben ist. Diese Verschiebung ist notwendig, da ein Angreifer sonst trivial auf den Kreismittelpunkt als Heimatadresse schließen kann. Innerhalb dieses Kreises werden dann überhaupt keine Standortdaten übermittelt, während außerhalb keine Inferenzangriffe stattfinden können und somit unverfälschte Daten preisgegeben werden können.

Überraschend ist bei den Ergebnissen von Krumm insbesondere die Größe der Ban Zone, die nötig ist, um die Ermittlung der korrekten Adresse aller Nutzer zuverlässig zu unterbinden. Erst ab einem Radius von $R = 2000m$ sind die einfachen Angriffe für keinen Teilnehmer mehr erfolgreich. Ein Nachteil dieser einfachen Verschleierungen ist darin zu sehen, dass keine Karteninformationen berücksichtigt werden. So sollte aus Sicht der Privatsphäre stets eine konservative Parametrisierung der Verschleierungsalgorithmen verfolgen, die auch in Extremfällen noch Schutz bietet. In interaktiven LBS vergrößert sich somit in vielen Fällen unnötig der Bereich, in der ein Dienst nicht genutzt werden kann. Im Rahmen von Messkampagnen wird damit oft ein zu großer Bereich ausgeblendet, wodurch die Datenqualität sinkt.

Gleichzeitig bietet ein derartiges Verfahren auch keine Garantien hinsichtlich der Unsicherheit eines Angreifers wie die l-Diversität, da insbesondere in locker besiedelten Gebieten unter Umständen nur sehr wenig Adressen als Wohnung in Frage kommen. Diesem Problem begegnen sowohl Gruteser und Liu [106] mit dem *k-Area*-Ansatz als auch Wiesner et al. [253] mit *Silent Zones*. In beiden Arbeiten wird vorgeschlagen, Bannzonen derart zu erzeugen, dass sie eine gewisse Anzahl an sensiblen Orten, d.h. verschiedenen Gebäuden beinhalten. Dadurch wird einerseits erreicht, dass die Zonen ähnlich der k-Anonymität gewisse Garantien hinsichtlich des erreichten Grades an Privatsphäre bieten, andererseits aber auch nicht unnötig groß werden. Innerhalb dieser Zonen werden keine Standortangaben veröffentlicht. Gruteser et al. erlauben dies jedoch nachträglich, wenn kein Aufenthalt in einer Zone stattgefunden hat. In [253] werden verschiedene Strategien zur Erzeugung solcher Zonen vorgestellt.

Ein großer Vorteil all dieser Ansätze ist, dass – obwohl [106] hierfür einen zetralen Location Server vorsieht – die Verschleierung der Standortinformationen rein clientseitig erfolgen kann und somit keine TTP nötig ist, der der Nutzer alle seine Orte anvertrauen muss. Anders verhält es sich mit dem *Mix Zones*-Verfahren von Beresford und Stajano [26]. Hier werden ebenfalls Bannzonen definiert, in der alle an einer TTP registrierten Nutzer keine LBS-Anfragen stellen. Eine Mix Zone muss so gewählt sein, dass sie in kurzer Zeit von ausreichend vielen Nutzern betreten wird. Der zentrale Server weist jedem Nutzer bei Betreten der Zone ein neues Pseudonym zu, sodass er nach Verlassen der Zone nicht trivial wiedererkannt werden kann. Das Anonymitätsset schwankt hierbei jedoch in der Größe und ist abhängig von der Anzahl an Nutzern, die

sich aktuell in der Zone befinden. Das System eignet sich daher auch nur für die Nutzung anonymer LBS, da aufgrund der häufig wechselnden Pseudonyme keine personalisierte Diensterbringung möglich ist.

Für den Einsatz in Datensammel-Kampagnen sind diese Verfahren überaus geeignet, da die Datenmenge zwar gerinfügig verringert wird oder zeitlich verzögert eintrifft, die Korrektheit der freigegebenen Daten aber nicht darunter leidet [50, 150]. Die Nutzung interaktiver LBS innerhalb einer Bannzone wird jedoch von keinem dieser Verfahren ermöglicht. Da sich ein Nutzer zu einem Großteil der Zeit aber eben genau an solchen privatsphäresensiblen Orten wie seinem Zuhause aufhält, stellt dies einen erheblichen Nachteil dar.

4.3.2 Standortverzerrung

In jeder Situation eine privatsphäreschonende LBS-Nutzung ohne den Einsatz einer TTP zu ermöglichen, ist das Ziel der clientseitigen Standortverschleierung. Anders als bei den dummybasierten Verfahren wird hierbei kein Set an diskreten Orten an den LBS übermittelt, in dem der tatsächliche Standort enthalten ist, sondern entweder eine kontinuierliche Fläche, die den Ort beinhaltet oder ein geeignet transformiertes Koordinatenpaar.

Der Vorteil dieser Verfahren ist daher, dass der tatsächliche Ort nicht in der Anfrage auftaucht. Zudem ist eine personalierte Dienstnutzung möglich, da sich ein Nutzer gegenüber dem LBS-Anbieter z.B. stets über ein Pseudonym identifizieren kann. Im Gegenzug leidet u.U. die Dienstqualität, da je nach Maß der vorgenommenen Verschleierung die LBS-Antwort vom optimalen Ergebnis abweichen kann. Grundlegende Annahme hinter diesen Verfahren ist jedoch, dass sich eine Vielzahl an LBS-Typen auch mit verzerrten Standortinformationen ohne merkliche Qualitätseinbußen nutzen lässt [11].

Die hierbei zum Einsatz kommenden Verfahren unterscheiden sich hauptsächlich in der Herangehensweise zur Erzeugung geeignet verschleierter Standortangaben, der Berücksichtigung kontinuierlicher Anfragen und dem Hintergrundwissen, das für die Verschleierung ggf. miteinbezogen wird.

Ardagna et al. stellen in [11] verschiedene geometrische *Obfuscation Operators* vor. Diese basieren auf der Beobachtung, dass die Messung des aktuellen Standorts z.B. mit GPS ohnehin schon einem gewissen Fehler unterliegt, der meist in Form eines Kreises um die tatsächliche Nutzerposition angegeben wird. Durch die dort eingesetzten Verschleierungmechanismen wird der durch Messung verursachte Fehler vor der Übermittlung des Standorts an den LBS bewusst verstärkt, um ein höheres Maß an Privatsphäre zu erreichen.

Zur eigentlichen Verschleierung des Standorts schlagen die Autoren die Vergrößerung und Verkleinerung des Kreisradius sowie das Verschieben des Mittelpunktes vor. Durch geeignete Kombination dieser drei Operatoren kann erreicht werden, dass die Schnittmenge des durch die Originalmessung erhaltenen Kreises mit dem letztendlich an den LBS übermittelten Kreis in einem beliebig kleinen Verhältnis zu dessen Fläche steht. Dieses Verhältnis interpretieren

die Autoren als *Relevanz* $\mathcal{R}$ der erzeugten Fläche, woraus sich der Grad an Privatsphäre als $1 - \mathcal{R}$ ergibt.

Nachteile an diesem Verfahren sind, dass es sich nur für sporadische LBS-Anfragen eignet und keine Karteninformationen berücksichtigt. So kann z.B. nicht gewährleistet werden, dass alle Orte in dem vom Nutzer an den LBS übermittelten Umkreis plausibel oder erreichbar sind, oder dass Inferenzangriffe auf Basis einer dafür nötigen Zahl an im Kreis enthaltenen semantischen Orte verhindert werden können. In [10] wird dasselbe Prinzip auf Informationen hinsichtlich der grundsätzlichen Begehbarkeit von Kartenbereichen erweitert. Insbesondere Straßen und Gebäude werden dabei als begehbar gekennzeichnet und der Radius R im Rahmen der Standortverschleierung kontinuierlich vergrößert, bis die oben genannten Eigenschaft der Relevanz auch auf die an begehbaren Bereichen abgedeckte Fläche zutrifft und somit vom Angreifer nicht auf Basis eines kartenbasierten Inferenzangriffs verringert werden kann.

Wie zuvor schon Dewri [59] (Kapitel 2.2.3.1) versuchen auch Andrés et al. ohne Verschleierungszonen auszukommen und stellen in [6, 43] eine clientseitige Standortverschleierung auf Basis der Differential Privacy vor. Genau wie in [59] wird hierbei eine zufällige Koordinatentranslation aus einer Laplace-Verteilung mit Skalenparemeter r gezogen. Dieser legt erneut fest, wie weit zwei Punkte voneinander entfernt sein können, dass sich die Wahrscheinlichkeiten, den entstandenen Punkt erzeugt zu haben, maximal um den Faktor e^{ϵ} unterscheiden. Im Umkehrschluss ist es einem Angreifer durch die Beobachtung des verschleierten Standorts kaum möglich Rückschlüsse ziehen, welcher Punkt im Abstand r um den echten Standort diese Verschleierung verursacht hat. Dieses Maß für Privatsphäre umschreiben die Autoren mit *Geo-Indistinguishability*.

Im Gegensatz zu der Arbeit von Dewri ist r dabei jedoch nicht durch den maximalen Abstand der Nutzer im Anonymitätsset festgelegt, sondern kann vom Benutzer individuell gemäß seiner Bedürfnisse gewählt werden. Auch dieses Verfahren bezieht keinerlei Kartenwissen mit ein und lässt sich nicht für kontinuierliche LBS-Anfragen einsetzen, da das Privacy-Budget mit jeder weiteren Anfrage schrittweise aufgebraucht wird. In [53] wird zudem als offene Forschungsfrage formuliert, ob eine solche Auflockerung des Differential Privacy-Prinzips effektiv ist und dieselben formalen Privacy-Garantien bietet.

Ardagna et al. stellen ein System zur Verschleierung sensibler Aufenthaltsorte und Bewegungen durch das Einfügen von Cover-Stories vor, die an das in einer Trainingsphase beobachtete Mobilitätsverhalten des Nutzers angelehnt sind [12]. Der Nutzer muss hierbei zunächst manuell alle Orte und Kanten eines Strassengraphs hinsichtlich ihrer Privatsphärerelevanz bewerten. Stellt der Algorithmus fest, dass sich der Nutzer an einem Ort befindet, den er nicht veröffentlichen möchte, erzeugt das Verfahren ein Alibi, das einen Aufenthalt in der Nähe, jedoch in einem als privatsphäretechnisch unbedenklichen Ort vorgaukelt.

Ähnlich den dummybasierten Verfahren stellt sich somit auch hier das Problem der realistischen, nicht angreifbaren Erzeugung gefälschter Trajektorien.

Im Gegensatz zu reinen Dummy-Ansätzen sinkt hier jedoch die Dienstqualität, da die echte Position bewusst verschwiegen wird und nur das Cover an den LBS übermittelt wird. Zudem wird eine lange Trainingsphase für die Erzeugung des Mobilitätsmodells des Nutzers benötigt, in der keine Dienstnutzung möglich ist. Darüber hinaus liegen für bisher unbesuchte und vom Nutzer nicht klassifizierte Gebiete keine Mobilitätsdaten oder Privatsphärepreferenzen vor, die zur Cover-Erstellung verwendet werden können. Die Autoren schlagen als Rückfalllösung die Verwendung der Privatsphärepreferenzen anderer Nutzer vor, was jedoch nur mit Hilfe einer TTP funktioniert. Die dafür notwendige Bereitschaft der Nutzer, ihre individuellen Privatsphärepräferenzen auf Ortsbasis mit anderen zu teilen, kann selbst hinterfragt werden.

In einer Serie von Publikationen beschäftigen sich auch Ghinita sowie Damiani et al. mit Möglichkeiten zur Verschleierung von Standortinformationen und schlagen verschiedene Verfahren vor, die unterschiedliches Hintergrundwissen und Vorgehensweisen des Angreifers berücksichtigen. Im Kern beruhen all diese Ansätze auf der Hinzunahme von Karteninformationen und der selektiven Verschleierung sensibler Aufenthaltsorte. Anstelle der tatsächlichen Position eines Nutzers wird dabei – sofern er sich irgendwo darin aufhält – jeweils eine sog. *Cloaking Region* (CR) übermittelt. Diese Regionen werden nutzerspezifisch um die individuell als privatsphärekritisch eingestuften Orte definiert. Außerhalb dieser Zonen wird der exakte Standort preisgegeben.

Im *PROBE*-Framework von Damiani et al. [54] werden unerreichbare Gebiete wie ein See mit einer Aufenthaltswahrscheinlichkeit von 0 belegt und Gebäude als semantische Orte eines bestimmten Typs interpretiert. Der Nutzer kann ein Privatsphäreprofil anlegen, das aussagt, welche Typen von Orten aus seiner Sicht privatsphärekritisch sind und welche nicht. Jede Klasse von semantischen Orten wird dabei als ein *Feature* interpretiert. Für jedes Feature muss ein Schwellwert τ für die *Sensitivität* P_{Sens} angegeben werden, der besagt, mit welcher Wahrscheinlichkeit ein Aufenthalt an einem Ort des jeweiligen Typs von einem Angreifer vermutet werden darf.

In einer Offline-Phase wird auf Basis des Nutzerprofils schließlich eine Partitionierung der Karte vorgenommen, in der um jeden als sensibel eingestuften Ort eine CR erzeugt wird. Dabei wird analog zu [10] darauf geachtet, dass die räumliche Ausdehnung der privatsphäresensiblen Orte im Verhältnis zur gesamten Fläche der CR τ nicht übersteigt. Die erzeugten CRs müssen zudem paarweise disjunkt sein. Die Autoren setzen auf eine offline Berechnung der CRs, um die Gefahr des Reverse-Engineerings auf die erzeugten Zonen zu verhindern, die bei einer online Berechnung gegeben wäre [231].

Ein Nachteil dieser CR-Erzeugung ist, dass nicht davon ausgegangen werden kann, dass gleich große Flächen dieselbe Aufenthaltswahrscheinlichkeit besitzen. So ist ein Gebäude am Ortsrand ein plausiblerer Aufenthaltsort als anliegende Grünflächen, was hier jedoch noch nicht berücksichtigt wird.

In [89] wird ein Angriff auf Basis des zeitlichen Abstands aufeinanderfolgender Standortupdates in Form von CRs beschrieben. Da die maximale Ge-

schwindigkeit, mit der sich ein Nutzer bewegt, von einem Angreifer abgeschätzt werden kann, ist es u.U. möglich, Teilbereiche der aktuellen oder der vorangehenden CR eindeutig auszuschließen. Konnten in der seit dem letzten Standortupdate vergangenen Zeitspanne nicht alle Punkte innerhalb der CR, in der sich der Nutzer jetzt befindet, erreicht werden, wird die LBS-Anfrage so lange hinausgezögert, bis dies der Fall ist. Die Autoren gehen hierbei davon aus, dass sich die Teilnehmer auf einer freien Fläche bewegen und berechnen die Erreichbarkeit von Orten anhand einer angenommenen Maximalgeschwindigkeit des Benutzers und der maximalen Hausdorff-Distanz der beiden CRs.

Da dies jedoch eine unrealistische Annahme darstellt, wird derselbe Angriff in [260] für Umgebungen formuliert, in denen Bewegungen durch ein Straßennetz eingeschränkt sind. Zudem wird hier die CR-Erzeugung aus [54] ebenfalls an das Straßennetz angepasst, was in Form des *Annotated City Networks* geschieht, das neben der Begehbarkeit von Orten auch Annahmen über deren Popularität trifft. Auf Basis des Nutzerprofils, das wie zuvor über die Features und entsprechende Schwellwerte definiert ist, werden anschließend wieder CRs erzeugt. Nur, wenn seit dem letzten veröffentlichten Positionsupdate so viel Zeit vergangen ist, dass alle Punkte in der aktuellen CR oder der aktuelle Punkt von allen Punkten der vorangehenden CR hätten erreicht werden können, wird die aktuelle Zone oder der genaue Standort preisgegeben. Andernfalls wird die LBS-Anfrage so lange verzögert, bis dieser Zustand eintritt.

Ein weiterer Nachteil all dieser Verfahren ist, dass der Angreifer aus den erzeugten CRs direkt das Privatsphäre-Profil des Nutzers ableiten kann. Da stets dieselben Typen von Orten in solchen Zonen auftauchen, wird dadurch verraten, an welchen semantischen Orte der Nutzer Besuche verbergen möchte.

Darüber hinaus stellt die Erstellung eines solchen Nutzerprofils und die Wahl geeigneter Schwellwerte einen komplizierten, manuellen Vorgang dar, den der Nutzer weit im Voraus planen muss. Zudem wird für diese Form der CR-Erzeugung eine große Menge an schwierig zu erlangendem Hintergrundwissen über die einzelnen Orte benötigt: So muss nicht nur der genaue Typ jedes Gebäudes bekannt sein, sondern auch Statistiken über deren Popularität vorliegen, die mit dem Hintergrundwissen des Angreifers übereinstimmen müssen.

Ein weiteres Problem der CR-Erzeugung in [260] ist, dass die Autoren das Straßennetz explizit als ungerichteten Graphen modellieren. Durch die Existenz von Einbahnstrassen und Abbiegeverboten ist das jedoch eine stark vereinfachte Annahme, die je nach Lage solcher gerichteter Kanten innerhalb einer Zone große Teile der CR unerreichbar machen können, je nachdem, aus welcher Richtung der Nutzer die Zone betritt. In solchen Fällen können die versprochenen Privatsphäregarantien nicht aufrechterhalten werden.

4.3.3 Privatsphäreschonende Routenplanung

Obwohl es sich bei der online Routenplanung um eine der am häufigsten genutzten Ausprägungen ortsbezogener Dienste handelt, haben sich bisher ver-

hältnismäßg wenige Arbeiten mit einer Lösung dieses Problems beschäftigt. Die Systeme, die sich unmittelbar mit dieser Problematik auseinandersetzen, werden im Folgenden vorgestellt. Keiner der bekannten Ansätze ist jedoch dazu in der Lage, alle Schritte bei der Ermittlung der schnellsten Route von A nach B ohne den Einsatz eines vertrauenswürdigen Dritten oder die Kooperation des LBS-Anbieters umzusetzen.

Mouratidis [181] stellt ein PIR-basiertes Verfahren zum Herunterladen der Kartenausschnitte von einem Kartenanbieter vor, die den kürzesten Weg von einem in Form von Koordinaten angegebenen Paar aus Start- und Zielposition beinhalten. Zu diesem Zweck wird ein manipulationssicherer, voll vertrauenswürder *sicherer Co-Prozessor* (SCP) als Modul an der Dienstschnittstelle des LBS-Anbieters installiert.

Der Autor schlägt zwei verschiedene PIR-Protokolle vor, die unterschiedlich großen Rechen- und Kommunikationsaufwand auf Seiten des Dienstanbieters verursachen. Die Kanten des Straßennetzes werden dabei unter Vorberechnung kürzester Wege im Rahmen eines Pre-Processings so auf die Speicherseiten verteilt, dass sich durch das Herunterladen einer bestimmten Seite keine Hinweise auf die Routenendpunkte des Nutzers ergibt. Für diesen Zweck werden außerdem zusätzliche Dummy-Seiten angefragt, u.a. um die Länge der gesuchten Route zu verbergen. Clientseitig muss hierfür zunächst eine Header-Datei heruntergeladen werden, die relevante Informationen über die PIR-bezogene Indexierung der Karte speichert. In mehreren Runden werden dabei die relevanten Seiten aus dem Speicher der Kartenanbieters an den Nutzer übertragen.

Wie bei allen PIR-basierten Methoden ist der große Vorteil dieses Ansatzes, dass der Dienstanbieter keinerlei Hinweise auf die tatsächlichen Routenendpunkte des Nutzers erhält. Auf der anderen Seite lässt sich das vorgestellte Verfahren nicht mit standardmäßig verfügbaren Routenplanern verwenden, da der Schutz der Privatsphäre nicht clientseitig gewährleistet werden kann und nur durch den Einsatz des SCP beim Dienstanbieter sichergestellt ist. Ob letzteres eine realistische Annahme ist, bezweifelt Mouratidis in [181] selbst.

Darüber hinaus eignet sich ein solcher PIR-Ansatz nicht für die Verwendung in verkehrsadaptiven Routenplanern, da das zeitaufwändige Preprocessingdurch aufgrund der häufig wechselnden Kantengewichte ständig aufs Neue durchgeführt werden müsste. Ein weiteres Problem ist, dass der Nutzer bereits im Vorfeld die Koordinaten seines Ziels kennen muss. Während dies für den Routenstartpunkt mittels des in mobilen Endgeräten verbauten GPS-Moduls eine realistische Annahme ist, bleibt offen, wie diese Information über den Zielpunkt privatsphäreschonend akquiriert werden kann.

Lee et al. stellen *OPAQUE* [161, 160] vor, wobei eine TTP für alle am System angemeldeten Benutzer als *Path Obfuscation Server* dient. Diese vertrauenswürdige Komponente empfängt die original Routenanfragen der Nutzer und generiert gemäß der individuellen Privatsphäre-Einstellungen eine sog. *obfuscated Path Query* und übermittelt diese an den Routenplaner.

Hierfür erzeugt die TTP für die beiden echten Routenendpunkte eine ge-

wisse Anzahl an gefälschten Positionen, für die ebenfalls Routen angefragt werden. Eine „verschleierte" Routenanfrage ist daher eine Menge aus Anfragen, die neben der echten Routenanfrage des Nutzers eine definierbare Menge an künstlich erzeugten Anfragen enthält. Orientiert man sich an der in der Literatur üblichen Begrifflichkeit, handelt es sich somit genau genommen nicht um einen verschleierungsbasierten Ansatz, sondern um ein Dummy-Verfahren, da die tatsächliche Routenanfrage des Nutzers stets mit übermittelt wird. Aus diesem Grund können die Autoren jedoch garantieren, dass in den Antworten des Routenplaners stets die optimale Antwort enthalten ist.

Dieser Ansatz weist jedoch auch einige Nachteile auf. Neben den grundsätzlichen Problemen, die mit dem Einsatz einer TTP einhergehen, berücksichtigen die Autoren bei der Erstellung der Dummy-Positionen keine Karteninformationen, sondern wählen diese zufällig aus der Nachbarschaft der echten Routenendpunkte [160]. Hierbei besteht die Gefahr, dass sich die gewählten Dummy-Positionen trivial als solche erkennen lassen.

Ein weiteres Problem ist, dass die Dummy-Positionen bei jeder neuen Routenanfrage zufällig neu erzeugt werden. Über eine Frequenzanalyse der beobachteten Anfragepunkte kann ein Angreifer nach Beobachtung mehrerer Routenanfragen somit die wiederholt auftauchenden Orte von den zufälligen Dummies unterscheiden. Ein dritter Nachteil ist, dass sich der erreichbare Grad an Privatsphäre – wie bei allen Dummy-basierten Ansätzen – nur durch die Übermittlung zusätzlicher Routenanfragen erhöhen lässt. Dieses Problem fangen die Autoren durch den Einsatz der TTP ab, die sich neben der Erstellung der Dummy-Positionen auch um das Filtern der Antwortmenge kümmert und dem Benutzer nur die für ihn relevante Antwort zukommen lässt.

In dieser Arbeit soll ein Verfahren gefunden werden, dass ohne den Einsatz einer TTP und mit den Standardschnittstellen tatsächlich verfügbarer online Routenplaner funktioniert. Der einzige Ansatz aus der Literatur, der diese Anforderungen zum Teil verfolgt, ist der von Vicente et al. [236]. Dieser ist daher auch am nächsten mit der vorliegenden Arbeit verwandt.

Auch dort verfolgen die Autoren das Ziel, die kürzeste Route zwischen zwei Punkten s und d von einem online Routenplaner zu beziehen, ohne exakte Hinweise auf diese beiden Orte zu geben. Die gesamte Karte ist dabei in ein reguläres Gitternetz in einzelne Zellen mit fester Kantenlänge aufgeteilt. Um die Koordinaten der Zieladresse in Erfahrung zu bringen, fragt der Client zu Beginn der Anfrage diese Information von einem vertrauenswürdigen *Location Obfuscator* ab, der auf Basis der Adresse die gesuchte Zelle ausgibt.

Die beiden Routenendpunkte werden somit auf zwei Zellen S und D abgebildet, für die der Client im nächsten Schritt das Kartenmaterial von einem Kartenanbieter herunterlädt. Im nächsten Schritt wird eine Liste mit allen paarweisen Kombinationen aus Knoten der Straßengraphen von Start- und Zielzone erstellt. Im Rahmen der eigentlichen Routenermittlung schickt der Client so lange Anfragen für diese Knotenpaare an den Routenplaner, bis ein zuvor festgelegter Anteil dieser Listeneinträge c über die extern in Erfahrung

gebrachten kürzesten Routen beantwortet ist.

Um die Anzahl an hierfür nötigen Routenanfragen zu reduzieren, beobachten die Autoren, welche noch ausstehenden Knotenpaare durch die aktuell eingehende Routenanfrage beantwortet werden, da sie sich auf der Antwortroute befinden. Es werden verschiedene Heuristiken für die Reihenfolge der Routenanfragen vorgeschlagen, die darauf abzielen, möglichst viele solcher implizit beantworteten Paare zu erzeugen. In den von Vicente et al. durchgeführten Experimenten wird ab einer Abdeckung von $c = 75\%$ der Knotenpaare zuverlässig eine vollständige Route gefunden, jedoch ist nur eine der vorgeschlagenen Heuristiken (*BoundariesDistance*) dazu in der Lage, dies zu garantieren. In über 80% der Fälle ist die gefundene Route streckenmäßig nicht länger, als die bei der unverschleierten Routenanfrage ermittelte Antwort. Es wird jedoch nicht untersucht, wie sich die ermittelten Routen bzgl. ihrer Fahrzeiten verhalten.

Während dieses Verfahren den Vorteil hat, dass es ausschließlich die Standardschnittstellen existierender Routenplaner nutzt, erzeugt es auf der anderen Seite eine hohe Anzahl an Routenanfragen. Bei einer Zellgröße von 1000m müssen für die Beantwortung einer verschleierten Anfrage im Durchschnitt über 60 Routenanfragen an den LBS übermittelt werden. Die hohe Zahl an Anfragen, die für die zuverlässige Beantwortung einer Routenanfrage nötig ist, erklärt sich dadurch, dass in [236] das lokal bekannte Straßennetz nicht genauer analysiert wird. Wie im weiteren Verlauf dieser Arbeit gezeigt wird, lassen sich damit eine Vielzahl an überflüssigen Anfragen einsparen, ohne den Grad an Privatsphäre oder die erreichbare Dienstqualität zu beeinträchtigen.

Wie schon in der Diskussion der verschiedenen Ansätze zur Erzeugung von Bann-Zonen ist ein weiterer Nachteil dieses Systems, dass zur Erzeugung der Verschleierungszonen keine Karteninformationen berücksichtigt werden. So kann auch bei diesem Verfahren nicht garantiert werden, dass in allen Zonen eine ausreichend hohe Anzahl an semantischen Orten, d.h. plausiblen Start- und Zielpunkten vorliegt. Dies wird auch noch einmal in Kapitel 4.5 gezeigt.

Ein weiterer Nachteil ist, dass die Autoren keine privatsphäreschonende Lösung für den notwendigen Geocoding-Schritt anbieten. Stattdessen definieren sie die Rolle eines vertrauenswürdigen *Location Obfuscator*, der diese Aufabe übernimmt. Die Autoren schlagen hierfür den Einsatz der mittlerweile eingestellten LBS *Google Latitude* vor – gleichzeitig wird Google Maps als Beispiel eines neugierigen Routenplaners genannt, vor dem Start und Ziel der Anfrage verborgen werden sollen. Die größte Schwäche dieses Verfahrens liegt jedoch in der hohen Anzahl an Routenanfragen, die für das garantierte Finden einer Route an den LBS übermittelt werden müssen.

Xue et al. beschreiben die Routenplanung als Beispielanwendung für ihr Konzept der *Location*-Diversity [258]. Auf die mögliche privatsphäreschonende Umsetzung dieser LBS-Anwendung gehen die Autoren nicht ein, schlagen zur Umsetzung jedoch ebenfalls die Verwendung einer TTP vor und berücksichtigen lediglich die l-diverse Verschleierung des Startpunktes, nicht aber der Zieladresse.

Viele Ansätze aus der Literatur gehen somit entweder davon aus, dass eine TTP exisitert oder dass der Dienstanbieter in Form spezieller Anwendungsschnittstellen kooperiert, um beliebig verschleierte Anfragen korrekt zu beantworten. Andere Verfahren vermeiden diese Annahme – immerhin stellt der Dienstanbieter selbst den als nicht vertrauenswürdig eingestuften, potentiellen passiven Angreifer dar. In diesem Fall gilt es den Tradeoff zwischen Privatsphäre und Dienstqualität zu ermitteln und zu optimieren, da die von einem unmodifizierten LBS-Anbieter erhaltenen Antworten auf verschleierte Standort-Queries von den optimalen Ergebnissen abweichen können.

Ein solches Verständnis liegt auch der vorliegenden Arbeit zugrunde. Es soll daher ein System entwickelt werden, das mit möglichst wenig Kommunikationsaufwand alle Teilschritte einer qualitativ hochwertigen, verkehrsadaptiven online Routenplanung clientseitig und privatsphäreschonend umsetzt.

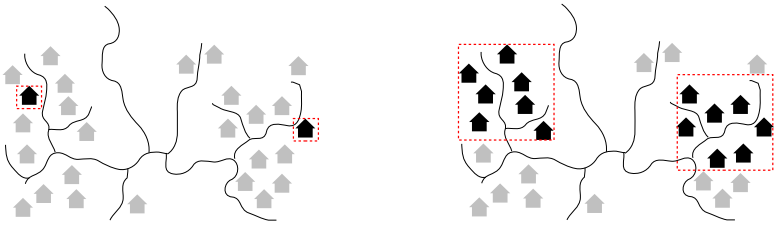
4.4 Privatsphäreschonende Umsetzung der online Routenplanung

In diesem Abschnitt wird mit PrOSPR (engl. *n̆derlinePrivacy-preserving Online Shortest Path Retrieval*) ein clientseitiges Verfahren für die privatsphäreschonende Nutzung verkehrsadaptiver online Routenplaner präsentiert. Angefangen beim Geocoding von Start- und Zieladresse bis hin zur Erzeugung einer vollständigen, verkehrsregelkonformen Ergebnisroute, werden für alle bei der Routenermittlung nötigen Teilschritte Methoden vorgestellt, welche die Standort-Privatsphäre des Nutzers schützen. Zu keinem Zeitpunkt werden die tatsächlichen Endpunkte der Routenanfrage, die personenbezogene Informationen darstellen, an den Routenplaner oder sonstige Parteien kommuniziert. Das Endgerät des Nutzers ist somit wie schon bei ALPACA die einzige als vertrauenswürdig eingestufte Systemkomponente, die im Rahmen der LBS-Nutzung in Besitz der exakten Kontextinformationen gelangt.

Das vorgestellte System orientiert sich rein an den standardmäßigen Web-Schnittstellen bestehender LBS und bedarf daher keinerlei Kooperation seitens des Dienstanbieters, die über die reguläre Diensterbringung hinaus geht. Die Vermeidung von ortsbasierten Inferenzangriffen durch den Dienstanbieter wird durch das in Kapitel 4.2.3 eingeführte Konzept der *k-immunen Routenanfragen* angestrebt. Die konsequente Einhaltung dieses Prinzips garantiert, dass sich weder Start- noch Zieladresse einer Routenanfrage auf weniger als k Kandidaten eingrenzen lassen.

Wie im weiteren Verlauf dieses Kapitels gezeigt wird, erlaubt der Parameter k eine für den Nutzer einfach nachvollziehbare Abwägung zwischen dem gewünschten Maß an Standortanonymität und der im Mittel dadurch zu erwartenden Verschlechterung bzgl. der Qualität der Routenplanung.

Vor dem Hintergrund der in Kapitel 4.2 beschriebenen Angriffsvektoren zielt



(a) Standardanfrage (b) PrOSPR mit $k = 7$

Abbildung 4.1: Die unterschiedlichen Detailstufen von Ortsinformationen, die bei einer Standard-Routenanfrage und mit PrOSPR preisgegeben werden. Im Normalfall lernt der Dienstanbieter die exakten Adressen einer Anfrage, bei PrOSPR nur eine grobe Region.

PrOSPR darauf ab, Inferenzattacken auf Basis der vom Routing-Anbieter sammelbaren Informationen über die von einem Nutzer besuchten Orte zu verhindern. Im Gegensatz zu heute verfügbaren LBS-Implementierungen, bei denen der Nutzer wie in Abb. 4.1a eine Route vom tatsächlichen Start- zu seinem tatsächlichen Zielpunkt anfragt, soll in dieser Arbeit eine alternative Vorgehensweise untersucht werden, bei der der Routingdienst diese präzisen Ortsinformationen nicht erhält. Stattdessen werden die Routenendpunkte mit Hilfe eines kartenbasierten Verschleierungsmechanismus derart verschleiert, dass sich hieraus keine Rückschlüsse auf die tatsächlichen Orte ziehen lassen.

Aus Sicht des Dienstanbieters verschwinden Start und Ziel einer Routenanfrage dabei wie in Abb. 4.1b dargestellt in zwei Zonen, die jeweils eine Verschleierungsmenge von mindestens k verschiedenen Adressen beinhalten. Da online Routenplaner standardmäßig jedoch keine Schnittstellen für die Unterstützung derartiger Region-zu-Region-Anfragen bereitstellen, kümmert sich der PrOSPR-Client selbst um die zur Lösung dieses Problems nötigen Schritte.

Die wichtigsten Beiträge, die das hier vorgestellte System zur privatsphäreschonenden Nutzung bestehender online Routenplaner liefert, lassen sich wie folgt zusammenfassen.

Zum Ersten wird ein umfassender Ad-hoc-Mechanismus für die kartenbasierte Verschleierung von Adress- und Standortinformationen im Rahmen von Routenanfragen vorgeschlagen, der auch die Schritte des Geocodings und der Akquise des für die lokale Routenvervollständigung notwendigenden Kartenmaterials privatsphärekonform umsetzt.

Zum Zweiten wird auf Grundlage des hierfür eingeführten Prinzips der k-immunen Routenanfragen ein intuitiv parametrisierbares und rein clientseitig umsetzbares Verfahren präsentiert, das die Privatsphäre eines Nutzers bei der online Routenplanung ohne den Einsatz einer TTP effektiv schützt.

Drittens werden unterschiedliche Heuristiken und Optimierungen des Basissystems vorgestellt und evaluiert, die eine Balance zwischen dem Grad an Privatsphäre, den Qualitätseigenschaften der so ermittelten Routenergebnisse

und dem dafür notwendigen Kommunikationsaufwand sicherstellen sollen.

Darüber hinaus wird das eingesetzte Verfahren zur Erstellung der Verschleierungszonen hinsichtlich einiger beobachteter Schwachstellen analysiert. Auf Basis dieser Bewertung wird im Anschluss ein neues, deterministisches Verfahren zur topologiebasierten Erstellung reziproker Verschleierungszonen vorgestellt, das nicht nur die Verteilung privatsphäresensibler Orte berücksichtigt, sondern auch deren gegenseitige Erreichbarkeit im Straßennetz bei der Zonenerstellung einbezieht. Wie gezeigt wird, lassen sich die so erstellten Verschleierungszonen auch für andere Ausprägungen ortsbezogener Dienste sinnvoll einsetzen und weisen dabei bessere QoS-Eigenschaften als bestehende Verschleierungsmechanismen auf.

Im Folgenden wird zunächst der vollständige Systemaufbau und Kommunikationsablauf von PrOSPR vorgestellt. Aufbauend auf der modular gehaltenen Gesamtkonzeption werden daraufhin verschiedene Optimierungsmöglichkeiten präsentiert, um die erreichbare Dienstqualität unter Beibehaltung der geforderten Privatsphäregarantien zu verbessern und den zusätzlich entstehenden Kommunikationsaufwand zu verringern. Im Anschluss daran wird eine umfangreiche empirische Evaluation des Systems durchgeführt.

4.4.1 Systementwurf und Kommunikationsablauf von PrOSPR

Der Systementwurf von PrOSPR folgt einem modularen Aufbau, der in mehreren Teilschritten die Ermittlung der schnellsten Strecke zwischen zwei Punkten von einem verkehrsadaptiven online Dienst privatsphäreschonend ermöglicht. Die einzelnen Teile des Algorithmus lassen sich durch alternative Umsetzungskonzepte und Implementierungen austauschen, was die einfache Integration und Evaluierung unterschiedlicher Herangehensweisen erlaubt.

Um die Privatsphäre der Nutzer im Hinblick auf ortsbasierte Inferenzangriffe bei der Verwendung eines online Routenplaners zu schützen, führt PrOSPR auf abstrakter Ebene die in Abb. 4.2 skizzierten Schritte aus:

1. Initiiert der Nutzer eine Routenanfrage an den online Dienst, wird clientseitig zunächst eine konservative, übertrieben grobe Verschleierung auf die Endpunkte angewendet. Ziel ist die Erzeugung zweier geografischer Regionen, die den jeweiligen Anfragepunkt enthalten und deren enthaltene Adressanzahl die vom Nutzer geforderte Größe der Verschleierungsmengen garantiert übertrifft.

2. Im Anschluss werden geokodierte Informationen über die exakte Verteilung privatsphäresensibler Orte innerhalb dieser Regionen von einem ebenfalls als nicht voll vertrauenswürdig eingestuften Kartenanbieter heruntergeladen. Unabhängig davon, ob es sich dabei um den Routenplaner selbst oder um eine dritte Partei handelt, gelten bei diesem Schritt somit dieselben Privatsphäreanforderungen wie zuvor.

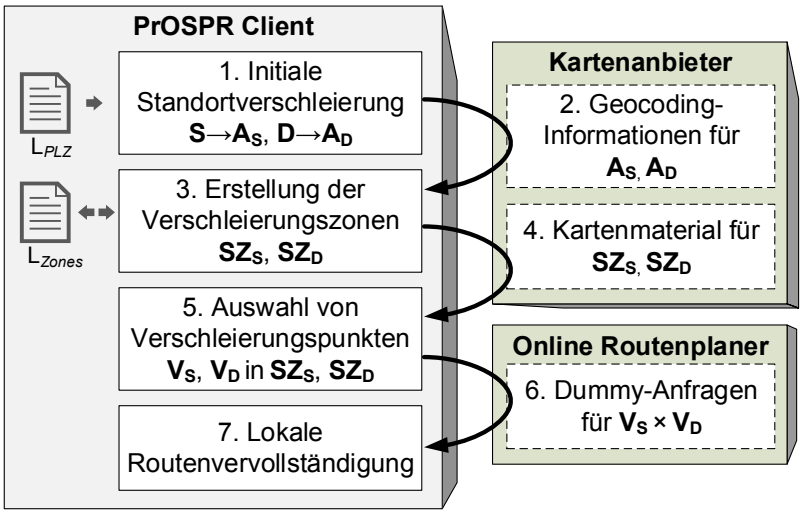


Abbildung 4.2: Grundsätzlicher Ablauf einer Routenanfrage mit PrOSPR.

3. Die clientseitig verwaltete Liste L_{Zones} enthält die Beschreibungen früher bereits verwendeter Verschleierungszonen. Fallen Start- oder Zieladresse der aktuellen Anfrage in eine solche Zone, wird diese erneut verwendet. Ist dies nicht der Fall, führt der Client eine kartenbasierte Verfeinerung der in Schritt 1 erstellten Regionen durch, wobei die Größe der Zonen unter Einhaltung der Privatsphäreanforderungen des Nutzers optimiert wird, um eine möglichst hohe Dienstqualität zu erlauben.

4. Im nächsten Schritt werden die Straßennetzwerke der beiden soeben erstellten Verschleierungszonen vom Kartenanbieter heruntergeladen.

5. Gemäß verschiedener Heuristiken werden anschließend unterschiedlich viele und unterschiedlich charakterisierbare Punkte innerhalb der Straßennetzwerke von Start- und Zielzone ausgewählt.

6. Diese dienen als Endpunkte für eine entsprechende Anzahl von privatsphäretechnisch unbedenklichen Dummy-Anfragen an den Routenplaner, welche die optimale Routenantwort approximieren sollen.

7. Da eine vollständige Route zwischen den tatsächlichen Endpunkten der Routenanfrage hierbei nicht garantiert werden kann, versucht der Client im letzten Schritt, aus den vom Routenplaner in Erfahrung gebrachten Dummy-Routen eine möglichst optimales Ergebnis zu erzeugen.

Somit vermeidet PrOSPR auch in der Kommunikation mit dem Kartenanbieter die Weitergabe präziser Angaben bzgl. der Start- oder Zieladresse einer Routenanfrage an externe Komponenten. Selbst wenn es sich bei Routenplaner und Kartenanbieter um kollaborierende oder identische Parteien handelt, ist die Privatsphäre des Nutzers somit stets gemäß des Parameters k geschützt.

Durch die geeignete Aufgabenverteilung zwischen Client und Dienstanbieter verbleibt das Wissen um diese privatsphäresensiblen Orte auf dem Endgerät des Nutzers. In den folgenden Abschnitten werden die einzelnen Schritte im Detail erklärt sowie die im Rahmen der vorliegenden Arbeit verwendeten Heuristiken und Implementierungen beschrieben.

4.4.1.1 Initiale Vergröberung der Routenendpunkte

Da die exakten Beschreibungen von Start- und Zielpunkt einer Routenanfrage das persönliche Endgerät nicht verlassen sollen, muss zunächst eine effektive, clientseitig durchführbare Verschleierung auf die beiden Endpunkte S und D angewendet werden. Um die Privatsphäre des Nutzers bzgl. seines aktuellen oder nächsten Aufenthaltsorts gemäß seinen Anforderungen zu schützen, gilt es zwei Regionen A_i, $i \in \{S, D\}$ zu finden, die jeweils mindestens k verschiedene privatsphäresensible Orte beinhalten, um die Wahrscheinlichkeit einer eindeutigen Identifizierbarkeit von S und D entsprechend gering zu halten.

Der Startpunkt einer Routenanfrage ist entweder durch eine vom Nutzer manuell eingetragene Adresse definiert oder durch die vom Endgerät ermittelten GPS-Koordinaten des derzeitigen Standorts. Für den Zielort wird angenommen, dass dieser grundsätzlich in Form einer vollständigen Adresse wie *Oettingenstrasse 67, 80538 München* angegeben wird. Die initiale Vergöberung derartiger Adresseingaben wird bei PrOSPR auf Basis des hierarchisch aufgebauten Postleitzahlsystems umgesetzt, das in vielen Ländern zur Verfügung steht.

Eine Postleitzahl beschreibt eindeutig ein räumlich zusammenhängendes Gebiet, dessen grobe geografische Einordnung anhand der ersten Ziffern der Zahl möglich ist. Die entsprechende Verschleierung lässt sich clientseitig trivial erreichen, indem Straße, Hausnummer und Ort der angegebenen Adresse einfach verworfen werden. Für Länder, in denen keine derartigen Systeme existieren oder diese wie in Großbritannien nicht öffentlich sind[1], können ähnliche Verfahren z.B. auf Basis von Städte- oder Gemeindegrenzen durchgeführt werden.

Die Verwendung statisch definierter Regionen für die initiale Adressverschleierung gibt eine obere Grenze für den maximalen Wert von k vor. Da ein solches Gebiet typischerweise jedoch mehrere 100 bis 1000 Adressen beherbergt, wird argumentiert, dass dieses Vorgehen für das hier verfolgte Einsatzszenario einen geeigneten Grad an Privatsphäre bietet. In Deutschland besitzen lediglich drei Gebäude in Frankfurt am Main und die Zugspitze eigene Postleitzahlen [215]. Derartige Sonderfälle können im PrOSPR-Client hinterlegt und einfach dem jeweils umschließenden bzw. einem angrenzenden Gebiet zugeschlagen werden.

Ist der Startpunkt einer Routenanfrage als Koordinatenpaar gegeben, ließen sich darauf theoretisch einfach zu berechnende, geometrische Verschleierungsmechanismen unmittelbar anwenden. Derartige Techniken werden u.a.

[1] `http://freethepostcode.org/`

von Ardagna et al. [9] und Krumm [148] beschrieben und basieren z.B. auf der durch einen maximalen Radius nach oben begrenzten Erzeugung eines zufällig verschobenen Umkreises, der die tatsächliche Nutzerposition beinhaltet. Ähnlich wie gitternetzbasierte Partitionierungsverfahren [236] schützen derart simple Verschleierungsansätze jedoch nicht zuverlässig vor Angreifern, die auf die Extraktion privatsphäresensibler Orte abzielen und über entsprechendes Kartenwissen verfügen. Da bei der Erzeugung der verschleierten Zonen die dafür nötigen Hintergrundinformationen nicht mit einbezogen werden, ist es in beiden Fällen je nach Besiedlungsdichte der Umgebung wahrscheinlich, dass die erzeugten Zonen oft weniger als die vom Nutzer geforderten k Adressen beinhalten, die zur effektiven Verhinderung von Inferenzangriffen benötigt werden. Natürlich sollen jedoch auch solche, nur über Koordinaten bekannten Standortinformationen zur Erzeugung k-immuner Routenanfragen führen.

Zu diesem Zweck umfasst der PrOSPR-Client eine Liste L_{PLZ}, die für jede relevante Postleitzahl unterschiedliche Beschreibung deren Grenzlinie bzw. deren geografischer Ausdehnung enthält. Einmal als detailliertes Polygon, das den Umriss des Postleitzahlgebeits präzise und überlappungsfrei nachzeichnet. Dies ermöglicht eine rudimentäre, speicherplatzschonende Form des clientseitigen Geocodings, indem einem berechneten Koordinatenpaar eindeutig das Postleitzahlgebiet zugeordnet werden kann, das diesen Ort enthält. Findet die Angabe des Startpunkts über die aktuellen GPS-Koordinaten des Endgeräts statt, ermittelt PrOSPR daher das Postleitzahlgebiet als Startregion A_S, in dem sich der Nutzer derzeit aufhält. Das Nachschlagen des jeweils passenden Postleitzahlgebiets lässt sich unter Verwendung geeigneter Indexstrukturen wie z.B. einem R-Baum [110] sehr effizient umsetzen. Durch den erneuten Rückgriff auf das Postleitzahlensystem kann damit genau wie bei der adressbasierten Verschleierung auch bei der Verwendung von GPS-Koordinaten sichergestellt werden, dass die initial erzeugte Verschleierungsregion eine unbedenklich große Zahl an unterschiedlichen privatsphäresensiblen Orten beinhaltet.

Anbieter von online Kartenmaterial wie OSM[2] erlauben das Herunterladen von Kartenausschnitten meist über die Angabe eines minimal umgebenden Rechtecks (MUR) oder eines beliebig komplexen Polygons, das die Umrisse des betreffenden Gebiets beschreibt. Je einfacher diese Form ausfällt, desto effizienter lassen sich i.d.R. die Anfragen beantworten. Aus diesem Grund enthält L_{PLZ} zudem eine auf wenige Stützpunkte simplifizierte Beschreibung der konkaven Hülle [191] jedes Postleitzahlgebiets, um eine effiziente Anfragebearbeitung durch den online Kartenanbieter zu ermöglichen. Diese Darstellung von Start- und Zielregion wird als Ergebnis dieses Schritts zurückgegeben.

Davon unabhängig, ob die Startposition in Form einer Adresse oder als Koordinatenpaar angegeben wird, ist der PrOSPR-Client nach diesem Schritt in Besitz der konservativ approximierten geografischen Umrisse zweier – im Hinblick auf k – ebenfalls konservativ verschleierter Regionen, A_S und A_D, welche die Privatsphäreanforderungen des Nutzers garantiert erfüllen.

[2] `http://www.openstreetmap.org/`

4.4.1.2 Privatsphärekonforme Akquise von Geocoding-Informationen

Die so ermittelten Regionen schützen die Privatsphäre des Nutzers zuverlässig gegenüber den beschriebenen Inferenzangriffen, da sie so gewählt sind, dass die Anzahl an enthaltenen Adressen das geforderte k jeweils deutlich übersteigt. Ohne das angestrebte Schutzziel zu gefährden, können diese Ortsangaben daher auch an externe Parteien kommuniziert werden. Im Gegenzug fallen diese Regionen jedoch auch räumlich gesehen übertrieben groß aus, was entweder zu einer schlechten durchschnittlichen Dienstqualität führt oder eine Vielzahl an Anfragen an den Routenplaner erfordert, um mit hoher Wahrscheinlichkeit eine hochwertige Routenantwort zu erhalten. Aus diesem Grund sollen diese Regionen im weiteren Verlauf derart verfeinert werden, dass sie unter Einhaltung der Privatsphärebedürfnisse des Nutzers eine qualitativ hochwertige Dienstnutzung bei vertretbarem Kommunikationsoverhead ermöglichen.

In Übereinstimmung[251] wird bei der Umsetzung von PrOSPR davon ausgegangen, dass eine effektive Verschleierung von Ortsinformationen nicht ohne die Berücksichtigung von Kartenwissen durchgeführt werden kann. Ein realer Angreifer, der wie der Anbieter eines online Routenplaners naturgemäß über derartiges Wissen verfügt, ist mit einfachen Mitteln dazu in der Lage, die realistischen Kandidaten für den tatsächlichen Ziel- oder Aufenthaltsort des Nutzers stark einzugrenzen. Hierbei können unwahrscheinliche oder unzugängliche Kartenbereiche wie z.B. Grün- und Freiflächen, Gewässer oder Straßen als mögliche Ziele trivial ausgeschlossen werden. Insbesondere vor dem Hintergrund der hier adressierten, ortsbasierten Inferenzattacken bleiben i.d.R. nur Gebäude als plausible Ziele übrig. Rein geometrische Verschleierungstechniken laufen daher immer Gefahr, einen grundsätzlich schwankenden und teilweise deutlich geringeren effektiven Grad an Privatsphäre zu bieten, als dies z.B. der gewählte Radius der Verschleierung zunächst impliziert. Dieser Nachteil kartenunabhängiger Verschleierungsmechanismen wird im weiteren Verlauf auch noch anhand einer beispielhaften Auswertung demonstriert.

Um die Verschleierung privatsphäresensibler Orte auf Basis von Adressangaben umzusetzen, muss stattdessen auf Geocoding-Informationen zurückgegriffen werden, wie sie von OSM und anderen Kartenanbietern zur Verfügung gestellt werden. Diese ermöglichen das Mapping von Adressen auf die zugehörigen Koordinaten eines geeigneten Referenzsystems und vice versa. Ein solcher Schritt ist einerseits notwendig, da es – anders als z.B. für WGS84-Koordinaten – keine geometrischen Verfahren für die Verschleierung von Adressen gibt. Diese Informationen sind andererseits aber auch unabdingbar für die Erzeugung einer vollständigen Route von Start- zu Zieladresse, da ohne sie keine Zuordnung dieser Punkte auf die Kanten des Straßengraphs möglich wäre. Angesichts der Ergebnisse von Krumm, der die schlechte Qualität einfacher Geocoding-Verfahren wie die lineare Interpolation von Hausnummern entlang einer Straße aufzeigt [148], verwendet PrOSPR hierfür exakte Informationen.

Der nächste Schritt des Algorithmus besteht daher im Herunterladen geokodierter Informationen B_{A_i} über die in den initial verschleierten Regionen

enthaltenen Adressen von einem Kartenanbieter. Dies geschieht durch die Formulierung geeigneter Anfragen an dessen Dienstschnittstelle, die mit den in L_{PLZ} über A_i, $i \in \{S, D\}$ jeweils enthaltenen Umrissbeschreibungen entsprechend parametrisiert werden, um die Privatsphäre des Nutzers zu wahren. Nach diesem Schritt ist die Verteilung privatsphäresensibler Orte in Start- und Zielregion bekannt, sodass dem Client nun die Hausnummern, Straßennamen und WGS84-Koordinaten aller in A_S und A_D enthaltenen Adressen vorliegen.

4.4.1.3 Gebäudebasierte Konstruktion von Verschleierungszonen

Als nächstes werden die im ersten Schritt grob ermittelten Verschleierungsregionen auf Basis der soeben akquirierten Geocoding-Informationen derart verfeinert, dass sie den tatsächlichen Privatsphärebedürfnissen des Nutzers genauer entsprechen und die geforderte Anzahl an möglichen Ausgangsadressen weniger stark überbieten. Dies geschieht, um den im weiteren Verlauf auftretenden Kommunikationsoverhead einzuschränken. So muss das Straßennetz nur für die kleineren Zonen und nicht für ein gesamtes Postleitzahlgebiet heruntergeladen werden. Eine weitere Motivation für die privatsphärekonforme Verkleinerung der Verschleierungszonen ist, mit möglichst wenig Anfragen möglichst große Bereiche der gefundenen Zonen verkehrsadaptiv vom Routenplaner berechnen lassen zu können, um eine hohe Dienstqualität zu erreichen. Daher werden auch die bereits erwähnten Dummy-Anfragen erst auf Basis der optimierten Verschleierungszonen formuliert und an den Routenplaner gesendet.

Um das Risiko ortsbasierter Inferenzangriffe effektiv und messbar einschränken zu können, sollen k-immune Routenanfragen ermöglicht werden. Dafür gilt es bei dem anstehenden Verfeinerungsschritt, sicherzustellen, dass jede Verschleierungszone mindestens k verschiedene Adressen enthält, an denen sich der Nutzer derzeit aufhalten könnte oder die sein nächstes Ziel darstellen können. Die hier vorgestellte Basisversion von PrOSPR setzt zu diesem Zweck den auf Gebäudeinformationen basierenden *Silent Zone*-Ansatz (SZ) ein, der von Wiesner et al. in [253] unter Mitwirkung des Autors vorgestellt wurde. Ursprünglich für den Erhalt der Privatsphäre im Kontext partizipativer Sensornetze entwickelt [50, 252], ist eine SZ als eine Zone um einen wichtigen Ort eines Benutzers definiert, in der er keine Messwerte an den Datenaggregator übermittelt, um seinen genauen Standort zu verbergen.

[253] stellt verschiedene Algorithmen zur nicht-deterministischen Erzeugung einer SZ vor und vergleicht deren Performanz. Im Gegensatz zu früheren Lösungsansätzen wie dem *Ban Zone*-Verfahren von Krumm [148] orientiert sich das Silent Zone-Konzept bei der Zonenkonstruktion an der Verteilung von Gebäuden in der Umgebung und kann daher garantieren, dass eine erfolgreich erstellte SZ mindestens k verschiedene Gebäude enthält. Der Einsatz zufälliger Prozesse bei der Zonenerstellung soll eine Rekonstruierbarkeit des eigentlichen Ausgangspunkts verhindern.

[253] stellt mehrere Bedingungen an eine SZ z, die für einen beliebigen privatsphäresensiblen Ort $p \in z$ erzeugt wird. Im Rahmen der vorliegenden Ar-

beit werden diese zu den folgenden beiden, für das Einsatzszenario der online Routenplanung de facto privatsphärerelevanten Anforderungen umformuliert:

1. z muss mindestens k verschiedene privatsphäresensible Orte beinhalten.
2. Anhand der Grenzen von z darf nicht mit einer Wahrscheinlichkeit größer als $\frac{1}{k}$ auf die Identität bzw. Adresse des Ausgangspunkts p geschlossen werden können.

Auf Basis der im vorigen Schritt akquirierten Geocoding-Informationen schlägt der PrOSPR-Client zunächst in der lokal verwaltenen Liste L_{Zones} nach, ob S oder D in eine früher bereits verwendete SZ fallen. Trifft dies zu, so wird dieselbe Zone für die aktuelle Routenanfrage wiederverwendet. Vor dem Hintergrund des Privatsphäreschutzes bietet dieses Vorgehen unter der getroffenen Annahme, dass der Dienstanbieter verschiedene Anfragen eines Nutzers durch die Existenz eines expliziten oder impliziten Pseudonyms eindeutig zuordnern kann, einen wichtigen Vorteil. Würde bei jeder Routenanfrage mit einem früher bereits verwendeten Endpunkt stets eine neue, zufällige SZ erzeugt, kann der Dienstanbieter durch Bildung der Schnittmenge der über die Zeit hinweg beobachteten Verschleierungszonen die Kandidatenmenge eingrenzen und im Worst-Case sogar eine eindeutige Identifizierung der echten Adresse erreichen. Auch nahegelegene Adressen, die rein zufällig in einer früheren SZ liegen, führen demnach zur Wiederverwendung dieser Zone. Aus Sicht der Privatsphäre ergibt sich des Weiteren der Vorteil, dass derart benachbarte Ziele eines Nutzers für den Dienstanbieter ununterscheidbar sind und nicht nachvollziehbar ist, wie viele Adressen ein Nutzer innerhalb einer Zone besucht.

Fallen S oder D jedoch nicht in eine früher bereits verwendete Zone, wird für den entsprechenden Endpunkt eine neue SZ erzeugt. Um eine möglichst hohe Dienstqualität zu gewährleisten, sollten diese Zonen den geforderten Wert für k idealerweise mit der kleinstmöglichen räumlichen Ausdehnung gewährleisten. Um allerdings zu verhindern, dass sich anhand der Lage einer derart optimalen SZ u.U. eindeutige Rückschlüsse auf die Ausgangsadresse ziehen lassen, zielt der Silent Zone-Ansatz darauf ab, Zonen zu erzeugen, welche die geforderte Anzahl an Gebäuden lediglich von oben her annähern, sich im Gegenzug aber mit Hilfe zufälliger, nicht eindeutig umkehrbarer Prozesse erzeugen lassen. Die Konstruktion der Zonen findet bei PrOSPR auf Basis des *Random-Rect*-Verfahrens (RR) aus [253] mit *k-Based Increase* (KBI) statt. Der genaue Ablauf der verwendeten Implementierung ist in Algorithmus 1 abgedruckt.

RR erzeugt auf Basis der aus B_{A_i} ermittelbaren Koordinaten loc_i iterativ SZ-Kandidaten für S bzw. D. Ein solcher Kandidat beschreibt ein zufällig um den Ausgangspunkt der Verschleierung versetztes Rechteck b, das diesen Punkt enthält. $loc.x$ und $loc.y$ stellen den Breiten- und Längengrad des zu verschleiernden Ortes dar. Die Funktion `rand` erzeugt gleichverteilte Zufallswerte zwischen den beiden angegebenen Grenzen. Diese werden dazu verwendet, den Zonenmittelpunkt zufällig um die eigentliche Ausgangsposition loc zu versetzen, um eine triviale Rückverfolgbarkeit zu verhindern.

Algorithmus 1 RandomRect (RR)

Require: $loc \wedge initsize > 0 \wedge k > 0 \wedge B \neq \emptyset \wedge \rho >= 1.0$
  $c \Leftarrow 0, area \Leftarrow initsize, f \Leftarrow 1.0$
  while $c < k$ **do**
    $width \Leftarrow \text{sqrt}(area/\text{rand}(1.0, \rho))$
    $height \Leftarrow (area/width)$
    $\text{shuffle}(width, height)$
    $x \Leftarrow loc.x - \text{rand}(0, width)$, $y \Leftarrow loc.y - \text{rand}(0, height)$
    $b \Leftarrow \text{rect}(x, y, x + width, y + height)$
    $c \Leftarrow \text{countBuildings}(B, b)$
    $f \Leftarrow \text{kbi}(c, k)$
    $area \Leftarrow f * area$
  end while
  return b

Der Wert für die initiale Zonengröße *initsize* kann ohne Gefahr für die Privatsphäre beliebig klein gewählt werden, da der gesamte Prozess lokal auf dem Endgerät des Nutzers abläuft. Um die Erzeugung „degenerierter", d.h. großer, extrem schmaler Zonen zu verhindern, wird mit ρ ein neuer Faktor eingeführt, der das maximal erlaubte Seitenverhältnis der Zone festlegt. Auf die Angabe einer minimalen Seitenlänge wird im Vergleich zu [253] hingegen verzichtet.

Sobald in einem Durchlauf mit b ein neuer SZ-Kandidat erzeugt wurde, zählt der Algorithmus die darin enthaltenen Gebäude $c = |B_{A_i}(b)|$. Ist der Wert von c kleiner als das geforderte k, vergrößert der Algorithmus iterativ die Fläche des in der nächsten Runde zu erzeugenden Kandidaten. Der dabei angewendete Vergrößerungsfaktor wird bei der Verwendung von KBI abhängig von der bereits erreichten Menge an Gebäuden bestimmt und nimmt ab, je mehr man sich der Zielgröße nähert, vgl. [253].

Erreicht bzw. übertrifft c in einem Durchlauf den gewünschte Wert von k, ist der Algorithmus beendet und die Grenzen b des aktuellen SZ-Kandidaten werden als Ergebnis zurückgegeben. Die so erzeugten Zonen SZ_S und SZ_D werden zudem in L_{Zones} gespeichert, um den eingangs beschriebenen Lookup bestehender SZs im Rahmen zukünftiger Routenanfragen zu ermöglichen.

4.4.1.4 Herunterladen von Kartenmaterial für Start- und Zielzone

Sobald auf diese Weise die exakten Grenzen der SZs für Start- und Zieladresse der Routenanfrage bekannt sind, werden die lokalen Straßennetze der beiden Zonen vom Kartenanbieter akquiriert. Im Gegensatz zu den Grenzen einer SZ wird deren Straßennetz jedoch nicht für künftige Anfragen gespeichert, sondern jedes Mal neu heruntergeladen, um stets aktuelle Karten zu verwenden.

Um für Sonderfälle vorzusorgen, die durch in den Randbereichen einer Zone liegende Gebäude entstehen, werden die soeben ermittelten SZ-Grenzen vor der Anfrage noch um einen konstanten Wert `BORDER` nach außen versetzt. Da-

durch werden bei der Kartenanfrage auch solche Straßensegmente berücksichtigt, die in der Realität die nächstgelegene Kante einer Adresse im Straßennetz darstellen, durch unglückliche SZ-Erstellung andernfalls jedoch nicht heruntergeladen würden. Dies kommt vor, wenn eine Straße unmittelbar außerhalb der Zonengrenze liegt, während die entsprechende Gebäude gerade noch zur Zone gehören. Für solche Adressen kann ohne diese Vergrößerung der angefragten *BoundingBox* kein korrektes Mapping auf den Straßengraph durchgeführt werden und keine korrekte Route ermittelt werden. Diese zusätzliche Fläche trägt jedoch nicht zur Erhöhung der Privatsphäre bei, da die hinzukommenden Randbereiche vom Angreifer trivial wieder entfernt werden können.

Die Abfrage des Kartenmaterials erfolgt wie zuvor über die Dienstschnittstelle des Kartenanbieters. Nun werden jedoch keine geokodierten Adressinformationen angefragt, sondern das befahrbare Straßennetz G_i der beiden erzeugten Verschleierungszonen, inklusive aller Wege, die mindestens einen Knoten innerhalb der Zonengrenzen liegen haben. Nachdem sowohl SZ_S als auch SZ_D garantiert mehr als k verschiedene privatsphäresensible Orte beinhalten, stellt auch diese Herausgabe der exakten SZ-Grenzen an externe Parteien keine Gefahr hinsichtlich der Privatsphäreanforderungen des Nutzers dar.

Mit Hilfe der nun lokal verfügbaren Geocoding- und Karteninformationen werden die Endpunkte der Routenanfrage S und D auf den jeweils nächstgelegenen Punkt auf einer Kante des entsprechenden Straßennetzes von Start- und Zielzone projiziert. Hierfür werden die kleinsten Distanzen der beiden Routenendpunkte zu allen Knoten und Kanten in G_i ermittelt. Liegt der jeweilige Endpunkt näher an einer Kante als an einem Knoten, wird er mittels Lotbildung realitätsgetreu auf das entsprechende Straßensegment platziert. Durch Einfügen eines neuen Knotens an dieser Position in den Straßengraph wird die Kante geteilt, um die spätere Routenvervollständigung zu ermöglichen.

Nach Abschluss dieses Teilschrittes stehen dem PrOSPR-Client nun die lokalen Straßengraphen G_S and G_D der beiden Verschleierungszonen zur Verfügung sowie die zuvor bereits heruntergeladenen Geocoding-Informationen B_S und B_D – ohne dafür präzise Angaben hinsichtlich der Adressen oder Positionen von S oder D an externe Komponenten preisgegeben zu haben.

4.4.1.5 Heuristische Auswahl von Verschleierungspunkten

Um auch unter dem Einsatz der in Abschnitt 4.4.1.3 vorgestellten Verschleierung der Routenendpunkte eine sinnvolle Dienstnutzung zu gewährleisten, muss PrOSPR neben dem zuverlässigen Schutz der Privatsphäre die qualitativ hochwertige Ermittlung der jeweils schnellsten Route ermöglichen. Angesichts der gewählten Verschleierung bedeutet dies, dass zu jedem möglichen Startpunkt in der Startzone der aktuell schnellste Weg zu jeder Adresse der Zielzone in Erfahrung gebracht werden muss. Um dabei in hohem Maße von der Verkehrsadaptivität des Onlinedienstes zu profitieren, sollte im Idealfall zudem jeweils die komplette Strecke durch den Dienst berechnet werden, da jede lokale Ergänzung ohne die Berücksichtigung aktueller Verkehrsinformationen

geschieht. Da online Routenplaner jedoch keine Schnittstellen für die Beantwortung der dafür nötigen Region-zu-Region-Anfragen zur Verfügung stellen, muss clientseitig für die Beschaffung einer korrekten Antwort gesorgt werden.

Eine naive Herangehensweise, um garantiert eine gültige Route vom Start zum Ziel zu erhalten, wäre es, jeweils eine Route von jeder Adresse in der Startzone zu jeder Adresse der Zielzone vom Dienstanbieter anzufragen. Ein derartiges Verfahren wählen Vicente et al., die vom Routenplaner die Verbindungen von allen Knoten in einer Startzelle zu allen Knoten der Zielzelle anfragen [236]. Trotz der dort umgesetzten Umsortierung und weiteren Strategien zur Vermeidung redundanter Anfragen, führt dieses Vorgehen jedoch zu einem immensen Kommunikationsaufwand, der zudem quadratisch mit der Parameterbelegung von k anwächst.

Um den durch die Verschleierung entstehenden Kommunikationsoverhead gering zu halten, werden bei PrOSPR daher gemäß verschiedener Heuristiken zwei Sets V_i an sog. *Verschleierungspunkten* (VPs) auf dem Straßennetzwerk der beiden Verschleierungszonen SZ_i ausgewählt, die später als Dummies für die Formulierung privatsphäretechnisch unbedenklicher Anfragen an den online Routenplaner dienen. Im weiteren Verlauf werden die so erhaltenen Routenergebnisse dann mit Hilfe des heruntergeladenen Kartenmaterials von Start- und Zielzone lokal ergänzt, um eine vollständige Route von S nach D zu erhalten.

Dieser Schritt kann nach der zuvor durchgeführten Verkleinerung der Verschleierungszonen daher als weitere Maßnahme zur Reduzierung des Kommunikationsaufwands für Endgerät und Dienstanbieter angesehen werden. Anders als bei dem Verfahren von Lee et al. [161, 160], bei dem die Originalanfrage nur durch die zusätzliche Übertragung zufällig erzeugter Dummy-Anfragen versteckt wird, wird die Privatsphäre nicht allein durch die Anzahl an übertragenen Anfragen geschützt. Durch die dort umgesetzte kartenunabhängige Erzeugung zufälliger Dummy-Anfragen ist deren Strategie zudem sowohl anfällig gegen Angreifer mit Kartenwissen als auch gegenüber einer Frequenzanalyse und der Identifizierung der echten Routenanfragen durch häufig vorkommende Routenendpunkte, was bei PrOSPR ebenfalls vermieden wird.

Stattdessen soll erreicht werden, dass aus Sicht des Routenplaners und unabhängig von der Anzahl an Anfragen, die an diesen geschickt werden, tatsächlich jeweils mindestens k privatsphäresensible Orte als plausible Ausgangs- und als Zieladresse gleichermaßen wahrscheinlich erscheinen.

Während die hierfür konzipierte Strategie der Verwendung von VPs und lokaler Routenvervollständigung aus Sicht der Privatsphäre und zur Reduzierung des Kommunikationsaufwands vorzuziehen ist, kann aufgrund des heuristischen Vorgehens nicht garantiert werden, dass die optimale Route gefunden wird. Stattdessen werden Umwege produziert, bzw. kann in einigen Fällen überhaupt keine Lösung erzeugt werden. Dieser bewusst durchgeführte Schritt führt damit potentiell zu einer Verschlechterung der Dienstqualität, die es selbst wiederum zu minimieren gilt.

In der Vorveröffentlichung von PrOSPR [68] sind die folgenden Heuristiken

für die Erzeugung entsprechender VP-Sets V_i in Start- und Zielzone enthalten:

- *Random node* (`R1`): Wähle zufällig einen Punkt auf einer Kante des Straßennetzwerks der aktuellen Zone.
- *Center* (`C1`): Wähle den Punkt auf einer Kante des Straßennetzwerk der aktuellen Zone, der die kleinste euklidische Distanz zum geografischen Mittelpunkt von SZ_i besitzt.
- *Random nodes* (`RN`): Wähle zufällig N Punkte auf Kanten des Straßennetzwerks der aktuellen Zone.
- *All Entry/Exit-Points* (`EE`): Ermittle alle Punkte `EE`, die aus der Startzone heraus bzw. in die Zielzone hineinführen. Entsprechende Punkte liegen auf jenen Kanten von G_i, die einen Knoten innerhalb sowie einen Knoten außerhalb der Zonengrenze von SZ_i liegen haben.
- *Random Entry/Exit-Points* (`EEN`): Wähle zufällig N Punkte aus den oben beschriebenen `EE`.
- *Random nodes, one reachable* (`RNr`): Wähle einen Punkt aus der Zusammenhangskomponente des lokalen Straßengraphen, der vom Startpunkt S erreichbar ist (zum Zielpunkt D führt). Fülle mit $N-1$ zufällig aus allen Graphkomponenten gewählten Knoten auf, um die Privatsphäre des Nutzers durch die Abdeckung verschiedener Teilgraphen zu schützen.
- *Random Entry/Exit-Points, one reachable* (`EENr`): Ermittle die Zusammenhangskomponenten des lokalen Straßengraphen der jeweiligen Zone und wähle einen zufälligen Punkt aus dem Set EE der Komponente, zu welcher der Startpunkt S (der Zielpunkt D) gehört. Fülle mit $N-1$ zufällig gewählten EE-Knoten aller Graphkomponenten auf.

Die `EE`-basierten Methoden beruhen auf der auch in [236] beobachteten Tatsache, dass grundsätzlich jede Route aus der Startzone und in die Zielzone einen dieser speziellen Punkte passieren muss, sodass diesen besondere Aufmerksamkeit geschenkt werden soll. Ein Vorteil dieser Strategien ist, dass sie sich automatisch der Anzahl an Ausfall- bzw. Eintrittsstrassen einer Zone anpassen und damit dazu beitragen, überflüssige Anfragen zu vermeiden.

Um durch die Wahl der VPs, die später an den Routenplaner übermittelt werden, keine impliziten Hinweise auf die tatsächlichen Routenendpunkte zu geben, laufen die ersten fünf Heuristiken gänzlich ohne die Berücksichtigung von S bzw. D ab. Stattdessen besitzt jeder Punkt im Straßennetzwerk G_i von Start- und Zielzone exakt dieselbe Wahrscheinlichkeit, zu V_i hinzugefügt zu werden. Die *one reachable* Strategien versuchen, die Erzeugung einer vollständigen Route von S nach D zu garantieren, indem sie mindestens einen Punkt auswählen, der in derselben Graphkomponente wie S bzw. D liegt. Was damit verhindert werden soll, ist dass zufällig nur solche VPs ausgewählt werden, die

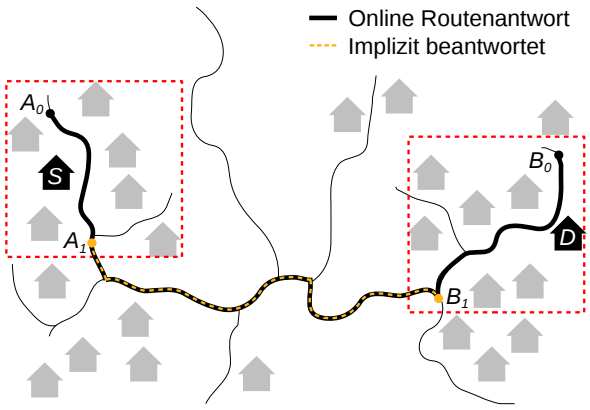


Abbildung 4.3: PrOSPR vermeidet redundante Routenanfragen, die durch vorangehende Antworten bereits implizit beantwortet wurden.

in nicht erreichbaren Zusammenhangskomponenten des lokalen Straßennetzes liegen und daher nicht zum Auffinden einer vollständigen Ergebnisroute von Start- zu Zielpunkt führen. Die Erreichbarkeit innerhalb des lokalen Straßen-Teilgraphen wird ad-hoc mit Hilfe des A*-Algorithmus [112] getestet.

Nach der Durchführung dieses Auswahlprozesses stehen zwei Mengen von VPs fest, V_S und V_D, die im Folgenden für die Formulierung von Dummy-Anfragen an den online Routenplaner verwendet werden. Welchen Einfluss die verschiedenen Auswahlheuristiken dabei auf die Optimalität und Vollständigkeit von Routenergebnissen haben, wird in Kapitel 4.5 evaluiert.

4.4.1.6 Übermittlung von Dummy-Anfragen an den Routenplaner

Um die vom Nutzer gewünschte Route qualitativ hochwertig annähern zu können, ohne dabei Hinweise auf die genauen Endpunkte zu geben, wird nun für jede mögliche paarweise Kombination von VPs in $V = V_S \times V_D$ eine Anfrage an den verkehrsadaptiven Routenplaner geschickt. Der dabei auftretende Kommunikationsaufwand entspricht daher $\mathcal{O}(|V|)$.

Abhängig von der im Schritt zuvor eingesetzten Auswahlstrategie werden hierfür unterschiedlich viele Verschleierungspunkte aus der Start- und Zielregion berücksichtigt. Unter der Verwendung geeigneter Belegungen für den Parameter N ist die Anzahl an VPs pro Zone jedoch bereits deutlich geringer als das geforderte k, wodurch sich unter Beibehaltung desselben Grades an Privatsphäre im Gegensatz zu [161] bereits eine Vielzahl an Anfragen einsparen lässt. Dennoch ist je nach VP-Heuristik mit einem Vielfachen des Kommunikationsaufwands zu rechnen, den eine normale Routenanfrage verursacht. Zwar gefährdet letztere u.U. die Privatsphäre des Nutzers, lässt sich dafür jedoch mit einer einzigen Anfrage zuverlässig und korrekt beantworten.

Neben der Verwendung intelligenter Selektionsstrategien kann eine Vermeidung überflüssiger Anfragen ferner durch die Berücksichtigung durch vorangehende Ergebnisrouten implizit beantworteter Routenanfragen erreicht werden.

Wie in Abb. 4.3 gezeigt, tritt dieser Fall ein, wenn beide Endpunkte einer noch geplanten Routenanfrage von A_1 nach B_1 auf einer Strecke $P_{A_0 \to B_0}$ liegen, die bereits als schnellste Route von A_0 nach B_0 ermittelt wurde. Aufgrund der Optimalität von Teilstücken kürzester Routen kann auf die Anfrage der schnellsten Route $P_{A_1 \to B_1}$ ohne der Gefahr von Qualitätseinbußen verzichtet werden. Analog zu [236] werden daher alle Antworten des online Routenplaners vor dem Absenden weiterer Anfragen auf die implizite Beantwortung ausstehender Routen hin überprüft und entsprechende VP-Paare aus V entfernt.

Dem Client stehen nach der Beantwortung der Dummy-Anfragen durch den Routenplaner nun eine Reihe von ungefähren Ergebnissen für die Anfrage des Nutzers zur Verfügung. Diese müssen abschließend noch sinnvoll zu einer Approximation der schnellsten Strecke zwischen den tatsächlich Routenendpunkten ergänzt werden, was im folgenden Schritt umgesetzt wird.

4.4.1.7 Ergebnisberechnung durch lokale Routenvervollständigung

Um dem Benutzer dieselbe Nutzungserfahrung wie bei der herkömmlichen Verwendung von online Routenplanern bieten zu können, muss ihm als Ergebnis der privatsphäreschonenden Routenplanung eine vollständige Route zwischen den von ihm angegebenen Endpunkten angezeigt werden. Nachdem zum Schutz der Privatsphäre die tatsächliche Route jedoch nicht explizit vom Routenplaner angefragt wird, muss PrOSPR auf Basis der erhaltenen Dummy-Antworten und den zuvor heruntergeladenen Karteninformationen durch lokale Vervollständigung sicherstellen, dass ein gültige Gesamtergebnis erzeugt wird.

Durch Zufall kann es eintreten, dass auch die tatsächliche Route durch eine Dummy-Anfrage bereits implizit beantwortet wurde. Daher wird jede Routenantwort $P_{A \to B}$, $A \in V_S$, $B \in V_D$ dahingehend untersucht, ob $P_{S \to D}$ eine Teilstrecke dieser Route ist. Trifft dies zu, ist damit nicht nur die optimale Route von S nach D gefunden, sie wurde auch noch vollständig verkehrsadaptiv vom online Routenplaner berechnet. In diesem Fall werden die überflüssigen Teilstrecken $P_{A \to S}$ und $P_{D \to B}$ einfach entfernt und die resultierende Strecke dem Nutzer als Ergebnis präsentiert. Dieser Vorgang kann parallel zur Übermittlung der noch verbleibenden Routenanfragen an den Dienstanbieter erfolgen. Um die Privatsphäre des Nutzers dadurch jedoch nicht einzuschränken, werden die ursprünglich geplanten Dummy-Anfragen konsequent weitergeführt, bis alle VP-Paare aus V korrekt abgearbeitet wurden. Dem Nutzer kann in diesem Fall jedoch schon im Vorfeld das optimale Ergebnis angezeigt werden.

Aufgrund der heuristischen Auswahl von VPs kann i.d.R. jedoch nicht davon ausgegangen werden, dass PrOSPR die Routenanfrage des Nutzers derart ideal beantworten kann. Für den Standardfall, dass die zuletzt erhaltene Dummy-Antwort $P_{A \to B}$ keine unmittelbare Lösung für $P_{S \to D}$ darstellt, wird daher gemäß einer der folgenden Strategien die lokale Vervollständigung der vom Routenplaner verkehrsadaptiv berechneten Ergebnisroute angestrebt:

- *Seek Dummy Points* (`SDP`): Vervollständige $P_{A \to B}$ zu einer gültigen Route

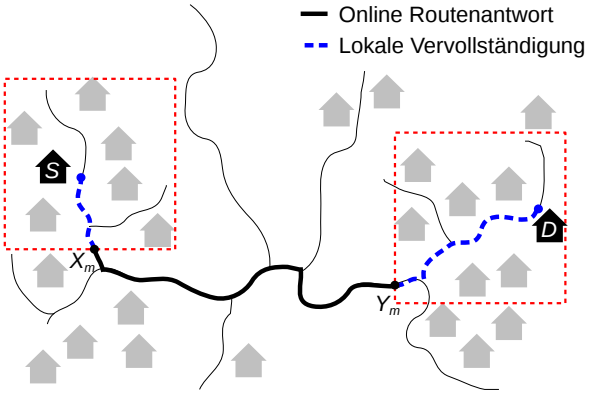


Abbildung 4.4: Innerhalb der erstellten Verschleierungszonen wird eine lokale Routenvervollständigung von S bzw. nach D durchgeführt.

von S nach D durch Hinzufügen der kürzesten Strecke $P_{S\to A}$ in G_S sowie der kürzesten Strecke $P_{B\to D}$ in G_D.

- *Seek Outer Bordercrossing* (`SOB`): Ermittle den ersten Austrittspunkt $E_S[1]$ aus der Startzone und den spätesten Eintrittspunkt $E_D[n]$ in die Zielzone auf $P_{A\to B}$. Entferne die Teilrouten $P_{A\to E_S[1]}$ und $P_{E_D[n]\to B}$ von $P_{A\to B}$. Vervollständige die resultierende Route $P_{E_S[1]\to E_S[n]}$ zu einer gültigen Route von S nach D durch Hinzufügen der kürzesten Strecke $P_{S\to E_S[1]}$ in G_S sowie der kürzesten Strecke $P_{E_D[n]\to D}$ in G_D. Im Gegensatz zu `SDP` ist diese Strategie dazu in der Lage, lokale Schleifen und Umwege innerhalb der Verschleierungszonen zu vermeiden, da direkt zum Zonenausgang geroutet wird.

- *Seek Inner Bordercrossing* (`SIB`): Ermittle den letzten von S in G_S erreichbaren Austrittspunkt $E_S[n]$ aus der Startzone und den ersten in G_D zu D führenden Eintrittspunkt $E_D[1]$ in die Zielzone auf $P_{A\to B}$. Vervollständige die Route wie oben.

- *Minimize Blindness* (`MB`): Ermittle die kürzeste Route in G_S von S zu jedem Punkt $p_i \in P_{A\to B}$, der in SZ_S liegt. Ermittle analog in G_D die kürzeste Route von jedem Punkt $q_i \in P_{A\to B} \cap \mathrm{SZ}_D$ zu D. Vervollständige die Route so, dass die jeweils kürzeste lokale Strecke an die entsprechend verkürzte Gesamtroute $P_{p_i\to q_i}$ angehängt wird, um dadurch den nichtverkehrsadaptiven Anteil der ermittelten Gesamtroute zu minimieren.

Bei extremer räumlicher Nähe von S und D kann es dazu kommen, dass keine Aus- und Eintrittspunkte in Start- und Zielzone existieren. In diesem Sonderfall überlappen sich die beiden Regionen und die verkehrsadaptiv berechnete, schnellste Verbindungsstrecke verläuft ausschließlich durch den von den Verschleierungszonen abgedeckten Bereich. Tritt dies ein, gilt bei den Strategien `SOB` und `SIB` daher $E_S = A$ und $E_D = B$, um auch in dieser Situation

ein vollständiges Ergbnis auf Basis der externen Routenantwort erzeugen zu können. Das Resultat ist hierbei dann äquivalent zur Strategie `SDP`.

Unabhängig davon, welche Strategie zur lokalen Routenvervollständigung eingesetzt wird, muss allerdings darauf geachtet werden, eine tatsächlich gültige Route zu erzeugen. Insbesondere darf dabei die „Nahtstelle" J_S, an der die lokale Route in der Startzone auf die extern berechnete Strecke trifft, nicht aufgrund eines Abbiegeverbotes regelwidrig sein. Dieser notwendige Schritt wurde z.B. in [236] nicht beachtet, sodass dort potentiell ungültige Ergebnisse entstehen können. In der Zielzone muss analog das Einbiegen von der externen Route an J_D auf die lokal ergänzte Route zu D erlaubt sein.

PrOSPR wendet daher das folgende Verfahren für die Erzeugung einer gültigen Gesamtroute an. Nach der Ermittlung von J_S gemäß einer der oben genannten Strategien wird mit A* und unter der Berücksichtigung von Einbahnstrassen und Abbiegeverboten die kürzeste Strecke in G_S von S über J_S zu einem interpolierten Punkt J'_S gesucht, der zwischen J_S und dessen Nachfolger entlang der extern berechneten Dummyroute liegt. Falls sich dabei keine Route erzeugen lässt, die verkehrsregelkonform auf die externe Route einbiegt, wird umgehend mit der Verarbeitung der nächsten Dummy-Antwort des Routenplaners begonnen.

Lässt sich in der Startzone jedoch eine gültige Route finden, wird die Vervollständigung anschließend analog in der Zielzone durchgeführt. Den Ausgangspunkt der lokal zu ermittelnden Strecke bildet hier nun ein interpolierter Punkt J'_D, der zwischen dem Vorgänger von $J_D \in P_{S\to B}$ und der Nahtstelle selbst liegt. Der tatsächliche Endpunkt D stellt das Ziel der lokalen Route dar, die wie in der Startzone einen verkehrsregelkonformen Übergang haben muss.

Das Ergebnis einer lokalen Routenvervollständigung ist in Abb. 4.4 dargestellt. Sobald eine vollständige Route von S nach D gefunden ist, schätzt PrOSPR die Dauer der approximierten Strecke. Hierfür werden die auf Basis der statischen Karteninformationen über G_S und G_D geschätzten Fahrzeiten der von $P_{A\to B}$ abgetrennten Teilstrecken $P_{A\to J_S}$ und $P_{J_D\to B}$ von $t_{A\to B}$ abgezogen und die der hinzukommenden Ergänzungen $P_{S\to J_S}$ und $P_{J_D\to D}$ addiert. Daraus ergibt sich aus jeder erfolgreich vervollständigbaren Dummy-Antwort eine geschätzte Gesamtdauer $t'_{S\to D}$. Die Länge der approximierten Gesamtroute kann anhand des Routenverlaufs exakt berechnet werden.

Nachdem eine verkehrsregelkonforme Route von S nach D gefunden wurde – oder wenn die Vervollständigung der aktuell untersuchten Dummy-Antwort $P_{A\to B}$ fehlschlägt – wird mit dem Verarbeitung der nächsten Routenantwort begonnen, bis alle VP-Paare V untersucht sind. Dem Nutzer wird anschließend die als zeitlich kürzeste geschätzte Gesamtroute als Ergebnis auf seine Routenanfrage präsentiert. Im Idealfall weicht das so erzielte Resultat nicht stark von der optimalen Route ab, die ohne den Einsatz der Endpunktverschleierung ermittelt worden wäre.

4.4.2 Anfragereduzierung mit PrOSPR+

Aufgrund des modularen Aufbaus der privatsphäreschonenden Routenplanung lassen sich einzelne Schritte einfach gegen andere Herangehensweisen austauschen. In diesem Kapitel werden zwei Erweiterungen für PrOSPR vorgestellt, die dazu beitragen, das Finden einer gültigen Routenantwort zu garantieren und die Anzahl an Anfragen, die zur Erzeugung einer gültigen Ergebnisroute nötig sind, zu verringern. Zu diesem Zweck werden neue, topologiebasierte Strategien zur initialen Ermittlung der Dummy-Anfragepunkte vorgeschlagen sowie das Prinzip der *Transitpunkte* eingeführt, deren Einsatz den Kommunikationsaufwand bei der Formulierung von Dummy-Anfragen reduziert.

4.4.2.1 Intelligente Heuristiken zur Auswahl von Verschleierungspunkten

Entscheidend für die Erzeugung einer korrekten Route von S nach D ist die Wahl geeigneter Dummy-Punkte für die späteren Routenanfragen. Die bisher verwendete Liste an Auswahlheuristiken aus Kapitel 4.4.1.5 wird daher nun um die beiden folgenden, komplexeren Strategien ergänzt. Im Gegensatz zu den bisher bekannten Arbeiten zum Schutz der Privatsphäre bei der online Routenplanung [68, 236, 161, 160] werden nun Details der lokalen bekannten Straßennetze analysiert, um eine bewusste Auswahl zu ermöglichen.

Um Anzahl und Lage der für das zuverlässige Finden einer gültigen Route benötigten VPs intelligent abzuschätzen, erzeugen sie sog. *Must-Have Components* (MHCs). Diese stellen problemspezifisch kombinierte Zusammenhangskomponenten der Straßengraphen dar, die wie folgt konstruiert werden:

- *Random from MHC* (`RMHC`): Ermittle für jede Adresse $ad \in \mathrm{SZ}_i$ das Set EP_{ad}, das die Menge an jeweils erreichbaren Austrittspunkten in der Startzone bzw. die Menge an zugehörigen Eintrittspunkten in der Zielzone beschreibt. Fasse alle Adressen mit identischen EP-Sets zu einer Komponente zusammen. Ist EP_a für zwei verschiedene Komponenten a, b eine Teilmenge von EP_b, fasse a und b zu einer MHC ab mit $EP_{ab} = EP_a \cap EP_b$ zusammen. Ist EP_a eine echte Teilmenge von EP_b, werden nur die Adressen von EP_a behalten, da ein VP später unbedingt aus diesen zu wählen ist. Komponenten, die sich nicht auf diese Weise mit anderen kombinieren lassen, stellen ebenfalls eine eigene MHC dar. Wähle anschließend einen zufälligen Punkt auf dem Straßennetz jeder MHC als VP.

- *Midmost from MHC* (`MMHC`): Wähle aus jeder MHC aus Start- und Zielzone den Punkt als VP, der unter allen Punkten der jeweiligen Komponente den größten minimalen Abstand zu den EPs der Zone aufweist.

Bei der Feststellung der gegenseitigen Erreichbarkeit von Ein- bzw. Ausgängen und den enthaltenen Adressen einer SZ muss eine große Anzahl an Routen

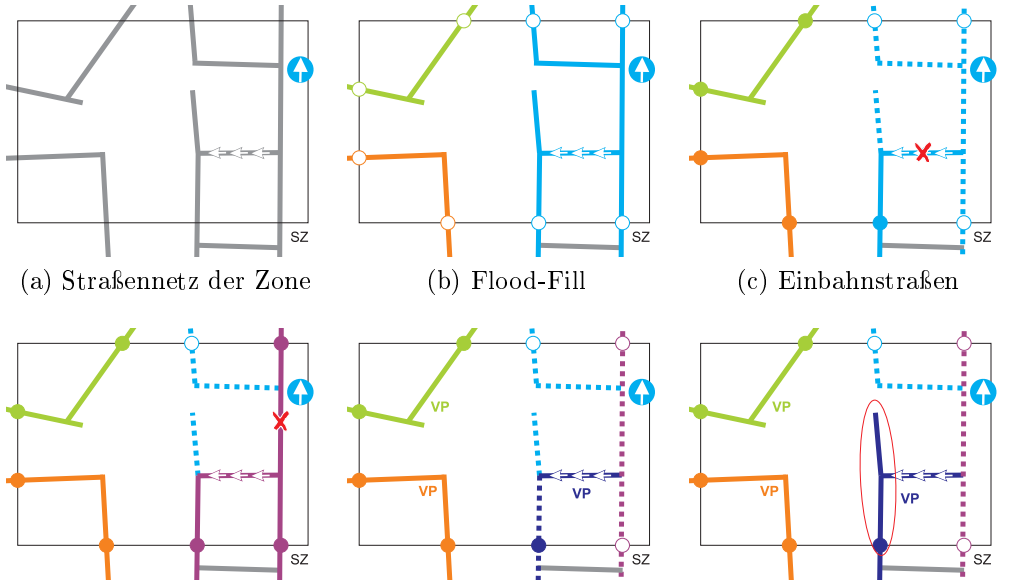
Algorithmus 2 Must-Have Components

Require: $b \wedge G_i \neq \emptyset \wedge source$
  $components \Leftarrow \emptyset, borderpoints \Leftarrow \emptyset$
  $components, borderpoints \Leftarrow$ findComponentsAndBorderpoints(G_i)
  for all $c \in components$ **do**
    $critical \Leftarrow$ extractOnewayAndTurnBanEdges(c, $source$)
    for all $j \in critical$ **do**
      for all $e \in borderpoints$ **do**
        if $source$ **then**
          $r \Leftarrow$ findRoute$(j.center, e, c)$
        else
          $r \Leftarrow$ findRoute$(e, j.center, c)$
        end if
        if $r \neq \emptyset$ **then**
          $j.ep \Leftarrow j.ep \cup e$
        end if
      end for
      $components \Leftarrow components \cup j$
    end for
  end for
  $mhc \Leftarrow$ mergeByReachableBorderpoints($components$)
  return mhc

lokal auf dem Endgerät des Nutzers berechnet werden. Um diesen Aufwand zu reduzieren, verwendet PrOSPR eine inhaltlich nahezu äquivalente, jedoch deutlich effizientere Umsetzungsvariante, die in Algorithmus 2 in Pseudocode beschrieben ist. Diese betrachtet neben schwachen Zusammenhangskomponenten ausschließlich „kritische" Stellen innerhalb des lokalen Straßengraphen, d.h. Einbahnstrassen und Abbiegeverbote, da diese – neben der trivialen Ursache, dass es innerhalb einer Verschleierungszone natürlich mehrere, nicht verbundene Teilgraphen geben kann – ausschlaggebend für die Nichterreichbarkeit bestimmter Teilbereiche des Graphen von verschiedenen Punkten aus sind. Eine grafische Darstellung der einzelnen Schritte ist in Abb. 4.5 zu sehen.

Im ersten Schritt des Algorithmus werden schwache Zusammenhangskomponenten in G_i durch ein einfaches Flood-Fill-Verfahren gefunden. Parallel dazu können die Ein- und Ausfallpunkte der Zone ermittelt und den entsprechenden Sets EP_j der jeweiligen Zusammenhangskomponenten hinzugefügt werden.

Als nächstes werden alle Einbahnstrassen und die an Abbiegeverboten beteiligten Kanten in den jeweiligen Zonen als mögliche Verursacher einer eigenen MHC innerhalb einer schwachen Zusammenhangskomponente darauf getestet, welche Grenzpunkte von diesen aus in der korrekten Fahrtrichtung erreichbar sind bzw. vice versa. Zu diesem Zweck wird mit dem A*-Algorithmus versucht, im lokalen Straßennetz eine verkehrsregelkonforme Route zwischen dem



(a) Straßennetz der Zone (b) Flood-Fill (c) Einbahnstraßen

(d) Abbiegeverbote (e) Merge-Schritt (f) Delta zur vollständigen Berechnung

Abbildung 4.5: Die einzelnen Schritte der MHC-Erzeugung.

Mittelpunkt eines Einbahnsegments bzw. der jeweils relevanten Kante eines Abbiegeverbots und den `EE`s der Zone zu erzeugen.

Der Eingabeparameter *source* gibt an, ob es sich beim Straßengraph G_i um die Startzone handelt. Im Graph G_S der Startzone wird hierfür von der Mitte aller Einbahnstrassen sowie von den initialen Kanten von Abbiegeverboten zu allen Ausgängen geroutet. In der Zielzone wird versucht, Routen von allen Eingängen zum Mittelpunkt der in G_D enthaltenen Einbahnstrassen sowie der von einem Abbiegeverbot betroffenen Zielkanten zu finden. Diese kritischen Stellen bilden jeweils einen eigenen MHC-Kandidaten j. Gelingt es, eine entsprechende Route zu finden, wird der zu untersuchte Austrittspunkt zu EP_j hinzugefügt. Das zugehörige Straßennetz stellen die an der untersuchten Stelle beteiligten Straßensegmente dar.

Im letzten Schritt werden die gefundenen MHC-Kandidaten auf Basis identischer EP-Mengen zusammengefasst, wodurch sich die Auswahl unnötiger VPs und damit auch das Versenden überflüssiger Anfragen an den Routenplaner vermeiden lässt. Da die eigentlichen Routenendpunkte hierbei – im Gegensatz zu den zuvor eingeführten *one reachable* Strategien – nie berücksichtigt werden, besticht dieser Ansatz aus Sicht der Privatsphäre zudem durch die Tatsache, dass garantiert jede Adresse aus der jeweiligen Zone eine der erzeugten Dummyrouten erreicht bzw. von dieser erreicht werden kann. Ein potentieller Angreifer kann somit keine Teilbereiche der SZ effektiv ausschließen.

Durch die derartige Konstruktion der *Must-Have Components* und deren Verwendung für die VP-Auswahl wird erreicht, dass PrOSPR immer eine gültige Route von S nach D mit einer sinnvoll minimierten Anzahl an VPs findet.

Dieser gesamte Vorgang lässt sich mit Hilfe der schon zur Verfügung stehenden Karteninformationen rein clientseitig umsetzen und verursacht keinerlei zusätzlichen Kommunikationsaufwand.

Es kann jedoch nicht garantiert werden, dass es sich dabei um die absolut minimale Anzahl an VPs handelt. So kann eine vom Routenplaner ermittelte Ergebnisroute z.B. die SZ zunächst verlassen, um diese für das Durchqueren einer anderen MHC c entlang der schnellsten Route erneut zu betreten.

Der hier vorgestellte Ansatz kann die Erzeugung eines eigenen – dann unnötig gewordenen – Verschleierungspunktes in c aufgrund der begrenzten Kenntnis des Straßengraphen nicht verhindern. Dies stellt jedoch einen selten zu erwartenden Spezialfall dar, der sich ohne Kenntnis des gesamten Straßengraphen nicht vermeiden lässt und zudem die exakte Lage von Start- und Zielpunkt sowie die aktuelle Verkehrslage berücksichtigen muss. Lassen es Lage der gewählten VPs und Reihenfolge der Dummy-Anfragen zu, kann dies jedoch mit dem in Kapitel 4.4.1.6 beschriebenen Verfahren auf Basis der implizit in den Routenantworten enthaltenen Informationen nachträglich erreicht werden.

4.4.2.2 Einsatz von Transitpunkten und Viarouten

Ein neuer Ansatzpunkt, um die Anzahl an Dummy-Anfragen an den Routenplaner zu reduzieren, lässt sich bei der Anfrageformulierung selbst umsetzen.

Seien V_S und V_D die Sets an Verschleierungspunkten der Start- und Zielzone einer verschleierten Routenanfrage. Um eine vollständige Route garantieren zu können, verlangt die bislang verwendete Strategie, dass für jedes Paar aus Verschleierungspunkten eine Anfrage an den LBS geschickt wird (vgl. Kapitel 4.4.1.6). Das Verfahren verursacht somit die Übermittlung und Beantwortung von $|V_S| \times |V_D|$ Routenanfragen.

Unter Inkaufnahme möglicher weiterer Umwege im Vergleich zur original Route lässt sich dieser Aufwand deutlich reduzieren. Zu diesem Zweck wird nicht mehr paarweise aus der Startzone direkt in die Zielzone geroutet, sondern über einen sog. *Transitpunkt*.

Die Idee bei der Verwendung von Transitpunkten beruht auf der Beobachtung, dass der Großteil an Routenantworten für die übermittelten Dummyanfragen ein gemeinsames Mittelstück aufweisen, das von vielen Routen besucht wird. Zudem erlauben alle heute verfügbaren Routenplaner die Angabe von Zwischenzielen entlang einer Route. Dieses Dienstangebot soll im Folgenden zur Einsparung von Routenanfragen eingesetzt werden.

Der Pseudocode der im Rahmen dieser Arbeit verwendeten Transitpunkt-Strategie ist in Algorithmus 3 zu sehen. Die erste Anfrage wird dabei wie bisher direkt von einem VP aus der Startzone zu einem VP aus der Zielzone berechnet. Aus der ersten Routenantwort des LBS wird ein Transitpunkt tp bestimmt, der auf dieser Route liegt. Die Methode `selectTP` wählt dabei den Punkt, der genau auf der Hälfte der Strecke liegt.

Bei allen folgenden Anfragen wird jeweils das nächste Paar an VPs ausgewählt und eine Anfrage mit tp als Viapunkt an den Routenplaner geschickt.

Algorithmus 3 Transitpunkt-Strategie

Require: $V_S \neq \emptyset \wedge V_D \neq \emptyset$
  $tp \Leftarrow \emptyset, resultset \Leftarrow \emptyset$
  $p \Leftarrow V_S.\text{pop}(), q \Leftarrow V_D.\text{pop}()$
  $route \Leftarrow \text{requestRoute}(p, q)$
  $tp \Leftarrow \text{selectTP}(route)$
  while $V_S \neq \emptyset \vee V_D \neq \emptyset$ **do**
    $p \Leftarrow V_S.\text{pop}() \vee p, q \Leftarrow V_D.\text{pop}() \vee q$
    $route \Leftarrow \text{requestViaRoute}(p, tp, q)$
    $resultset \Leftarrow resultset \cup route$
  end while
  return *resultset*

Somit kann auch hier garantiert werden, dass von jedem VP der Startzone eine vollständige Route zu jedem VP der Zielzone gefunden wird. Die Anzahl an nötigen LBS-Anfragen wird dabei von $|V_S| \times |V_D|$ auf $\max\{|V_S|, |V_D|\}$ reduziert.

Abschließend werden die gefundenen Via-Routen am Viapunkt aufgeteilt und so miteinander konkateniert, dass sich eine vollständige Route aus den `MHC`s des Start- und Zielpunkts ergibt, die dem Nutzer als Ergebnis präsentiert werden kann.

4.4.3 Zusammenfassung

In diesem Kapitel wurde PrOSPR vorgestellt, ein umfassender Ansatz für die clientseitige Umsetzung privatsphäreschonender Routenanfragen an die standardmäßig zur Verfügung stehenden Dienstschnittstellen existierender online Routenplaner. Die zuverlässige Verschleierung von Start- und Zieladresse einer Routenanfrage wird dabei in Anlehnung an das Prinzip der k-Anonymität und mit Hilfe eines kartenbasierten Verschleierungsverfahrens erreicht.

Durch den Einsatz geeigneter Teilschritte verlassen dabei genaue Angaben bezüglich der tatsächlichen Endpunkte der Routenanfrage zu keiner Zeit das Endgerät des Nutzers. Auf Basis der übermittelten Informationen sind externe Parteien und potentielle Angreifer wie der Kartenanbieter und Routenplaner selbst daher nicht dazu in der Lage, Aufenthalts- und Zielort des Nutzers mit einer Wahrscheinlichkeit größer als $\frac{1}{k}$ zu bestimmen. Ohne dafür auf eine TTP zurückgreifen zu müssen, ermöglicht PrOSPR somit die Umsetzug k-immuner Routenanfragen, die sowohl den Startpunkt als auch das Ziel von Routenanfragen effektiv verschleiern, um Inferenzangriffe wie die De-Anoymisierung und Profilerstellung auf Basis exakter Standort- und Zielangaben zu verhindern.

4.5 Evaluation der privatsphäreschonenden Routinganfragen

In diesem Kapitel werden die vorgestellten Verfahren einer empirischen Evaluation unterzogen und hinsichtlich verschiedener Aspekte analysiert. Wichtige Kennzahlen sind hierbei die erreichbare Dienstqualität sowie die durch die Verschleierung entstehenden, zusätzlichen Kommunikationskosten in Abhängigkeit des jeweils gewählten Verschleierungsgrades und der verschiedenen Heuristiken zur Auswahl der Anfragepunkte. Zudem wird untersucht, welcher effektive Grad an Standortanonymität bei der Verwendung k-immuner Routenanfragen im Vergleich zu bestehenden Verfahren zuverlässig erreicht wird.

Die erreichbare Dienstqualität wird anhand der Abweichung der verschleierten Routenergebnisse von den optimalen Routenantworten beurteilt, die bei der unverschleierten Verwendung der tatsächlichen Start- und Zieladressen erzielt worden wären. Der Vergleich betrachtet dabei nicht wie in [236] die Länge der verschleierten Routenergebnisse, sondern die entstehenden Abweichungen hinsichtlich der vom Routinganbieter geschätzten Fahrzeiten. So kann eine verschleierte Route streckenmäßig durchaus kürzer ausfallen als eine vom Routenplaner ermittelte Route. Wenn diese jedoch langsamere oder verkehrsbehinderte Strecken entlang führt, erhöht sich dadurch dennoch die Fahrzeit des Nutzers. Letztere stellt bei der Verwendung verkehrsadaptiver Routenplaner somit das entscheidende Kriterium dar und dient daher als Qualitätsmetrik.

Darüber hinaus wird analysiert, wie lang die jeweiligen Streckenabschnitte in Start- und Zielzone ausfallen, die mit Hilfe der lokalen Routenvervollständigung berechnet werden müssen. Auch dies stellt ein Qualitätsmerkmal dar, da für die entsprechenden Streckenabschnitte keine Echtzeit-Informationen des Dienstanbieters verwendet werden können, sondern diese rein auf Basis des statischen Kartenmaterials berechnet werden müssen.

4.5.1 Datengrundlage und Versuchsaufbau

Die Datengrundlage für die nachfolgend durchgeführten Experimente bildet das Kartenmaterial von OpenStreetMap (OSM) mit dem Stand vom 15.2.2016. OSM beinhaltet sowohl das Verkehrswegenetz als auch Gebäude- sowie Adressinformationen, die das notwendige Geocoding ermöglichen.

Für die Akquise von geokodierten Adressinformationen und Kartenausschnitten, die für die Erstellung der Verschleierungszonen und die lokale Routenvervollständigung notwendig sind, wird eine lokale Installation der *Overpass* API[3] verwendet, die eine eigene Anfragesprache für das selektive Herunterladen von OSM-Daten bereitstellt. Als Routingserver kommt eine eigene Instanz der *Open Source Routing Machine* (OSRM)[4] zum Einsatz. Die verwendete Implementierung von PrOSPR ist in *Python* umgesetzt.

[3] `http://wiki.openstreetmap.org/wiki/Overpass_API`
[4] `http://project-osrm.org/`

Die Kartendaten von OSM erheben trotz sehr aktiver Community keinen Anspruch auf Vollständigkeit. Einzelne Gebäude oder Adressangaben können in diesem Open-Source-Datensatz daher fehlen. Um dennoch realitätsnahe Ergebnisse erzeugen zu können, wurden durch visuelle Überprüfung drei verschiedene Regionen ausgewählt, die eine hohe Abdeckung an korrekt getaggten Gebäuden aufweisen. Um die Eigenschaften von PrOSPR in Hinblick auf unterschiedliche Besiedlungsdichten ermitteln zu können, wurden zudem solche Gebiete ausgewählt, die sich hierbei deutlich voneinander unterscheiden. Unter Berücksichtigung beider Aspekte fiel die Wahl auf die drei Städte München (4.601 Einwohner/km^2), Rosenheim (1.636) und Erding (655).

Als *semantischer Ort* gem. der Definition in Kapitel 4.2.3 – und damit als plausibler Endpunkt einer Routenanfrage – werden im Folgenden nur solche Gebäude berücksichtigt, die tatsächlich mit einer Adresse versehen sind.

Parameter	**Belegungen**
k	50, 100, 150, 200
`BORDER`	50 m
VP-Heuristik	`C1`, `R1`, `R5`, `R5r`, `EEP`, `EEP5`, `EEP5r`
Routenvervollständigung	`SDP`, `SOB`, `SIB`, `MB`
Routing-Szenario	in München, München→Erding, München→Rosenheim

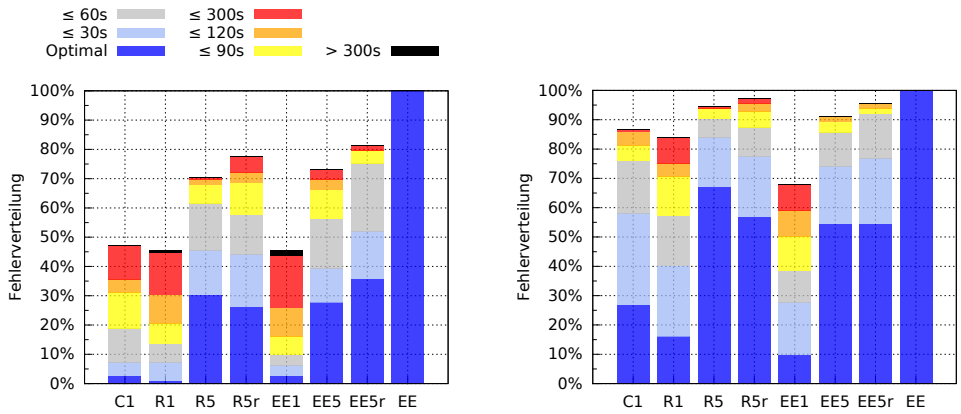
Tabelle 4.1: Verwendete Parameterbelegungen

Die Experimente werden gemäß der Parameterbelegungen aus Tabelle 4.1 durchgeführt. Um eine direkte Vergleichbarkeit der einzelnen Heuristiken und Vervollständigungsstrategien zu erlauben, werden für jede mögliche Kombination je 120 verschiedene Routen zwischen denselben, initial zufälllig ermittelten Start- und Zieladressen ermittelt und dieselben VPs verwendet.

4.5.2 Evaluation von PrOSPR und PrOSPR+

In den folgenden Abschnitten wird zunächst untersucht, wie sich das Originalverfahren PrOSPR in Sachen Dienstqualität und Standortanonymität verhält und welche Rolle dabei die verschiedenen Strategien zur Auswahl von Verschleierungspunkten und zur lokalen Routenvervollständigung spielen. Im Anschluss wird das Abschneiden der vorgeschlagenen Erweiterung PrOSPR+ untersucht und hinsichtlich Dienstqualität und Kommunikationsaufwand mit PrOSPR verglichen.

Die hierzu durchgeführten Experimente nutzen eine neue, umfangreichere Implementierung der Verfahren als im Rahmen der Vorveröffentlichung [68], wodurch nun detailliertere Betrachtungen möglich sind.



(a) Seek Dummy Points

(b) Minimize Blindness

Abbildung 4.6: Fehlerverteilung der verschleierten Routenplanung mit PrOSPR für $k = 100$ in München mit `SDP` und `MB`.

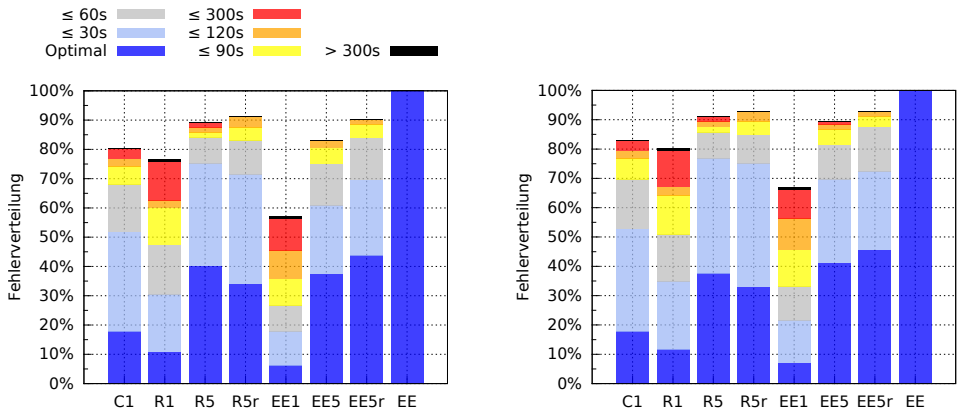
4.5.2.1 Qualität der gefundenen Routen

In diesem Schritt wird evaluiert, welche Dienstqualität sich bei der Verwendung der privatsphäreschonenden Routenplanung mit PrOSPR erreichen lässt. Um eine Balance zwischen Privatsphäre und Kommunikationsaufwand zu erreichen, wurden in Kapitel 4.4.1.5 verschiedene Heuristiken vorgeschlagen, die helfen sollen, LBS-Anfragen einzusparen. Im Idealfall weicht die Antwort auf eine verschleierte Routenanfrage nicht merklich von der optimalen Route ab, die bei der unverschleierten Anfrage zurückgegeben würde. Durch die heuristische Herangehensweise kann dies jedoch nicht garantiert werden.

Im Rahmen der vorliegenden Arbeit wird ein verschleiertes Routenergebnis als optimal gewertet, wenn es weniger als zehn Sekunden von der Dauer der unverfälschten Route abweicht. Zudem werden nur verkehrsregelkonforme Ergebnisrouten gewertet. Lassen sich Start- und Zielpunkt im lokal bekannten Straßennetz der verwendeten SZs überhaupt nicht oder nur unter Verstoß gegen geltende Verkehrsvorschriften (Wende- und Abbiegeverbote, Einbahnstraßen, etc.) mit einer Dummyroute verbinden, kann unter Verwendung dieses Dummys keine vollständige Route ermittelt werden.

In Abb. 4.6 ist die Fehlerverteilung der Ergebnisse der verschleierten Routenanfragen bei der Verwendung von PrOSPR für $k = 100$ innerhalb der Region München für verschiedene VP-Heuristiken und die unterschiedlichen Strategien zur Routenvervollständigung abgebildet. Bereits an dieser Stelle ist ersichtlich, dass – inklusive der *reachable* Ansätze – keine der vorgeschlagenen Heuristiken außer `EE` das Finden einer gültigen Ergebnisroute garantiert.

In Abhängigkeit von der verwendeten Strategie zur lokalen Routenvervollständigung schlägt das Auffinden einer vollständigen Route von S nach D mit unterschiedlicher Häufigkeit fehl. Die deutlichsten Unterschiede ergeben sich dabei zwischen den Methoden *Seek Dummy Points* (`SDP`, 4.6a) und *Minimize Blindness* (`MB`, 4.6b). Während `SDP` für die VP-Heuristiken `C1`, `R1` und `EE1`



(a) Seek Outer Bordercrossing

(b) Seek Inner Bordercrossing

Abbildung 4.7: Fehlerverteilung der verschleierten Routenplanung mit PrO-SPR für $k = 100$ in München mit `SOB` und `SIB`.

jeweils in mehr als 50 % der Fälle kein gültiges Ergebnis produzieren kann, gelingt dies bei `MB` in Kombination mit `C1` zu 86 % – obwohl beide Strategien dieselbe Ausgangslage hinsichtlich der SZs und gewählten VPs besitzen.

Die Erklärung für dieses Phänomen ist in der unterschiedlichen Arbeitsweise der beiden Strategien zur Routenvervollständigung zu sehen. `SDP` versucht, die externe Routenantwort durch das naive Anfügen zweier Teilrouten zu den Endpunkten der Dummyanfrage zu vervollständigen. Wie man sieht, lässt sich dabei jedoch in etwa der Hälfte aller Fälle kein verkehrsregelkonformer Pfad finden, der unter der Berücksichtigung der nötigen Fahrtrichtung und Verkehrsregeln korrekt auf die externe Route einbiegt. In 14,3 % der Fälle existiert unter Verwendung von `C1` überhaupt keine Route zu diesem Punkt im lokal bekannten Straßennetz, weitere 38,3 % lassen sich nur mit Hilfe eines in Deutschland meist verbotenen sog. „U-Turns" vervollständigen, also ein direktes Wenden auf einer Straße. Im Rahmen der vorliegenden Arbeit wird der zweite Fall als verkehrsregelwidriges Fahrmanöver gesehen, das dem Nutzer nicht als Route vorgeschlagen werden soll und daher als Fehlschlag gezählt wird.

Die Strategie `MB` hingegen versucht, eine Verbindung zu jedem lokal bekannten Punkt der externen Routenantwort zu finden und so früh wie möglich auf diese zu wechseln. Die kürzeste gefundene Strecke wird schließlich als Ergebnis ausgewählt. Dadurch, dass hierbei kein Besuch des gewählten Dummypunkts erzwungen wird, lassen sich hierbei sowohl mehr Routen überhaupt beantworten als auch optimal beantworten. Zudem werden, falls die externe Strecke lokal erreichbar ist, keine derartigen U-Turns verursacht. In 13,3 % der Fälle kommt es jedoch auch hier vor, dass keine Route zu einem lokal bekannten Punkt der externen Route gefunden werden kann, da der tatsächliche Start- oder Zielpunkt des Nutzers in einer Zusammenhangskomponente des Straßennetzes liegt, die keinen Punkt mit der Routenantwort des LBS gemein hat.

Die übrigen beiden Strategien, *Seek Outer Bordercrossing* (`SOB`, 4.7a) und *Seek Inner Bordercrossing* (`SIB`, 4.7b), die versuchen, eine Routenvervollständigung zu unterschiedlichen Aus- bzw. Eintrittspunkten der externen Routenantwort aus bzw. in die lokale Zone zu erzeugen, liegen fast gleichauf. Bei beiden Varianten fallen die erzielten Ergebnisse unabhängig von der VP-Heuristik jedoch schlechter aus als bei `MB`. Die `SIB`-Variante, die den spätest- bzw. frühestmöglichen Übergangspunkt als Nahtstelle auswählt, schneidet durchgängig geringfügig besser ab als `SOB`, bei der grundsätzlich nur der erste Aus- bzw. der letzte Eintrittspunkt der externen Route aus bzw. in die Verschleierungszone von Start und Ziel berücksichtigt wird. So findet `SIB` bei der Verwendung von `C1` z.B. in 17 % der Fälle kein gültiges Ergebnis, `SOB` bleibt in 20 % erfolglos. Was das Finden der optimalen Routenantwort betrifft, liegen beide Verfahren nahezu exakt gleich auf.

Insgesamt kann somit festgehalten werden, dass die Strategie *Minimize Blindness* den anderen Herangehensweisen zur lokalen Routenvervollständigung in allen Belangen überlegen ist. Sie findet – bei identischer Ausgangslage – stets am häufigsten eine vollständige, verkehrsregelkonforme Route von S nach D. In nur 5,3 % aller Fälle schlägt dies bei `R5` und 2,7 % bei `R5r` fehl. Auch die Ermittlung der optimalen Routenantwort gelingt hier mit Abstand am häufigsten: Bei Verwendung der VP-Heuristik `R5` handelt es sich in 67 % aller untersuchten Fälle um das optimale Ergebnis, das auch bei der unverschleierten Routenanfrage ermittelt worden wäre. Darüber hinaus verursacht `MB` nie Umwege, die länger als 5 Minuten andauern. In 90 % der Fälle bleibt der ingesamt durch die Verschleierung hervorgerufene Umweg hierbei sogar im Bereich von unter 60 Sekunden. Sofern nicht explizit anders kenntlich gemacht, wird im Folgenden daher nur noch diese Strategie berücksichtigt.

In den anderen Routingszenarien zeigt sich beim Vergleich der verschiedenen Konkatenationsstrategien dasselbe Bild mit lediglich kleinen Verschiebungen der Prozentwerte. Unabhängig von der Umgebung der Routenendpunkte ist somit stets der Einsatz der *Minimize Blindness*-Strategie zu empfehlen.

Innerhalb eines einzelnen Diagramms in den Abb. 4.6 und 4.7 lassen sich deutliche Unterschiede zwischen den verschiedenen Heuristiken zur Auswahl der Dummypunkte erkennen. Wie zu erwarten ist, schneiden die `*1`-basierten Verfahren, die nur eine verschleierte Routenanfrage an den LBS schicken, erheblich schlechter ab als jene Ansätze, die mehrere VPs auswählen. Unabhängig von der jeweiligen Strategie zur Routenvervollständigung äußert sich dies sowohl in einer geringeren Anzahl an optimalen Routenergebnissen als auch in einem höheren Anteil an überhaupt nicht gefundenen Routen.

Die Strategie `C1` sorgt stets für mehr optimale Routenergebnisse als `R1` und `EE1`. Dieses Verhalten ist darauf zurückzuführen, dass Routen, die im Mittelpunkt der Zone starten oder enden, im Durchschnitt eine höhere Wahrscheinlichkeit haben, den – aus Sicht der optimalen Route von S nach D – korrekten Aus- bzw. Eintrittspunkt aus der Zone zu wählen: Routen, deren Endpunkte

im Zentrum der Verschleierungszone liegen, besitzen eine global „günstigere" Ausgangslage, da durch den externen Routenplaner bereits innerhalb der Zone Entscheidungen getroffen werden, die für viele mögliche Start- bzw. Endpunkte ebenfalls sinnvoll sind. Werden die Dummypunkte jedoch wie bei `R1` zufällig gewählt, können diese nah am Rand der Zone liegen und letztere über den für sie idealen Ausgang verlassen oder betreten. Bei `EE1` liegen diese Punkte zwangsläufig am Rand und geben den Übertrittspunkt somit oft fest vor.

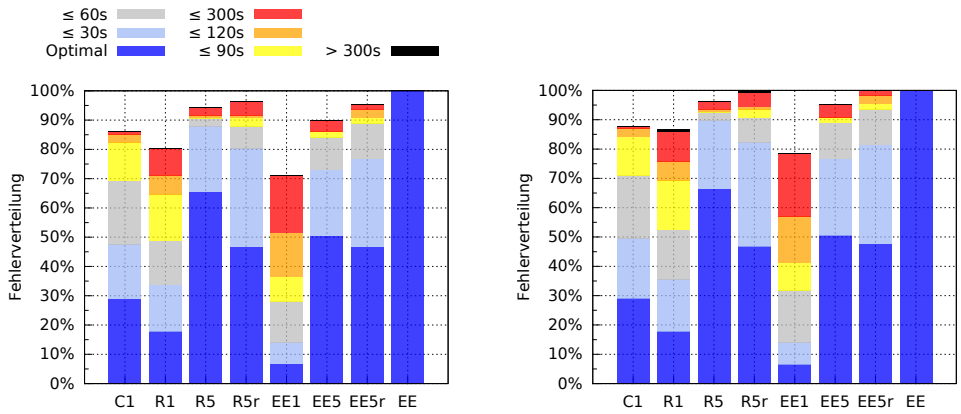
Unterscheidet man nur zwischen diesen Heuristiken, ist `C1` den anderen beiden somit eindeutig vorzuziehen. Praktisch einsetzbar ist jedoch keine davon, da sie selbst bei der Verwendung der ansonsten erfolgreichen `MB`-Strategie zwischen 13 (`C1`) und 32 % (`EE1`) der Routenanfragen nicht beantworten können. Ein Vorteil dieser Herangehensweisen ist, dass der Dienstanbieter nicht unterscheiden kann, ob es sich um eine verschleierte oder eine unverschleierte Routenanfrage handelt. Eine Verfügbarkeit von jeweils deutlich unter 90 % stellt jedoch mit Sicherheit keine praxistaugliche Lösung dar.

Am anderen Ende der Skala, was die Anzahl an gewählten VPs angeht, befindet sich die Strategie `EE`, welche Dummyanfragen für alle Paare an Ein- und Austrittspunkten von Start- und Zielzone an den LBS versendet. Diese ist die einzige von den in Kapitel 4.4.1.5 vorgestellten Heuristiken, die das Finden einer gültigen Ergebnisroute von S nach D garantiert. Unabhängig von der jeweiligen Strategie zur lokalen Routenvervollständigung handelt es sich dabei zudem stets um das optimale Ergebnis, da durch die paarweise Abfrage aller möglichen Übergangspunkte auch die optimale Route im Rahmen der Dummyanfragen garantiert enthalten ist. Wie im weiteren Verlauf gezeigt wird, erkauft man sich diese hohe Dienstqualität jedoch mit immensem Kommunikationsoverhead, sodass auch der Einsatz dieser Heuristik keine praxisfähige Lösung darstellt.

Einen Mittelweg für das Erzeugen einer vollständigen Ergebnisroute unter Eingrenzung der Anzahl an nötigen LBS-Anfragen stellen die `*n`-basierten Auswahlheuristiken dar. Im Rahmen dieser Arbeit wird deren Performanz examplarisch für $n = 5$ analysiert. Pro Route, die verkehrsadaptiv und privatsphäreschonend ermittelt werden soll, erzeugen diese Verfahren somit maximal 25 Anfragen an den LBS-Anbieter. Bei der Verwendung der Strategie `MB` sind sowohl die Heuristiken `R5` und `R5r` als auch `EE5` und `EE5r` in mehr als 90 % der Fälle in der Lage, dem Nutzer ein gültiges Routenergebnis zu präsentieren.

Die *Random*-Verfahren, die VPs aus dem gesamten bekannten Straßennetz wählen, erzielen dabei bessere Ergebnisse als die `EE`-basierten Heuristiken. Dieses Verhalten wie zuvor darauf zurückzuführen, dass die `EE`-basierten Herangehensweisen VPs am Rand der Zone auswählen, wodurch die Chance, den im Durchschnitt günstigsten Übertrittspunkt der Zone zu besuchen, sinkt. Das Auffinden der optimalen Routenantwort variiert dabei zwischen 54 % bei der Verwendung von `EE5` und 67 % mit `R5`.

Der Einsatz der *Reachable*-Erweiterung soll zu einer geringeren Anzahl



(a) Verbot von U-Turns

(b) Zulassen von U-Turns

Abbildung 4.8: Fehlerverteilung in Abhängigkeit der Akzeptanz von U-Turns in den Ergebnisrouten ($k = 200$, `MB`, in München).

an nicht auffindbaren Routen führen, kann den Erfolg der verschleierten Routenermittlung unter Einhaltung aller Verkehrsregeln jedoch auch nicht in jedem Fall sicherstellen. Akzeptiert man U-Turns auf einer Straße hingegen als erlaubtes Fahrmanöver, werden bei `R5r` und `EE5r` stets vollständige Routen gefunden. Die Fehlerverteilung unter Erlaubnis solcher U-Turn-Routen ist in Abb. 4.8 für $k = 200$ dargestellt. Wie eingangs argumentiert wurde, werden solche gefährlichen Navigationsanweisungen im Rahmen der vorliegenden Arbeit jedoch als ungültig interpretiert und sollen daher vermieden werden.

Als nächstes wird untersucht, ob und wie sich die verschiedenen Werte von k sowie unterschiedliche Bevölkerungsdichten auf die Qualität der gefundenen Routen auswirken. In Abb. 4.9 ist der Anteil an optimalen Routenergebnissen für verschiedene Parameterbelegungen und VP-Heuristiken innerhalb der Region München dargestellt. Die verschiedenen Auswahlheuristiken zeigen für alle Werte von k dieselben Muster bzgl. der Anteile an optimalen Routenergebnissen, die zuvor schon für $k = 100$ beim Vergleich der verschiedenen Vervollständigungsstrategien beobachtet wurden. So schneiden z.B. die `*1`-basierten Verfahren durchwegs schlechter ab als jene Strategien, die zur Erzeugung mehrerer Dummyanfragen an den LBS führen.

Wie zu erwarten ist, lässt sich mit steigendem Wert von k ein negativer Trend erkennen, was das Auffinden der optimalen Routenantwort betrifft. Durch höhere Werte von k wächst sowohl die Anzahl an Straßensegmenten in der Verschleierungszone als auch die Anzahl an möglichen Ein- und Austrittspunkten, wodurch auch die Chance, dass ein VP ausgewählt wird, der die Zone durch einen – aus Sicht der tatsächlichen Routenendpunkte – suboptimalen Ausgang verlässt, steigt. Gleichzeitig erhöht sich dadurch aber der Grad an garantierter Standortanonymität, sodass ein hoher Wert von k den Privatsphärebedürfnissen eines Nutzers u.U. eher entspricht. Es lässt sich in diesem Zusammenhang

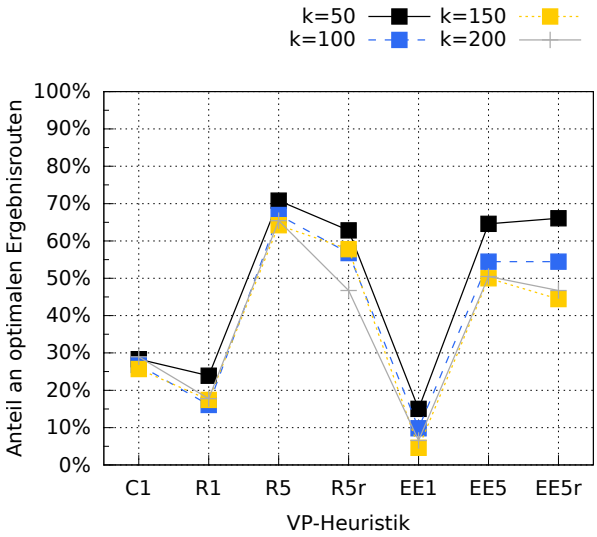


Abbildung 4.9: Anteil an optimalen Routenergebnissen in der Region München für verschiedene k unter Verwendung von `MB`.

jedoch die positive Beobachtung machen, dass der Anteil an optimalen Routenantworten nur verhältnismäßig langsam abnimmt. Für $k = 50$ findet die `R5r`-Strategie in 62,8 % der Fälle die optimale Route. Für $k = 200$ liegt dieser Wert immerhin noch bei 46,7 %. Der vierfache Wert für das Maß an Anonymität der Routenendpunkte führt somit nur in 16 % mehr der Fälle zu einem nicht optimalen Ergebnis.

Neben dem Anteil an optimalen Routenergebnissen ist auch die im Durchschnitt zu erwartende Dienstqualität über alle verschleierten Routenanfragen hinweg von Interesse. In vielen Fällen ist PrOSPR zwar dazu in der Lage, die Anfrage des Nutzers mit einer gültigen Ergebnisroute zu beantworten, findet dabei jedoch nicht den optimalen Routenverlauf. In diesem Fall entstehen Umwege, was sich aus Sicht des Nutzers in einer höheren Fahrzeit im Vergleich zu dem Ergebnis einer unverschleierten Routenanfrage ausdrücken lässt.

Abb. 4.10 zeigt den durchschnittlichen Qualitätsverlust für verschiedene Werte von k als das Mittel der zusätzlichen Fahrzeiten, die durch die von der verschleierten Routenanfrage verursachten Umwege im Vergleich zur optimalen Route hervorgerufen werden. Es fließen dabei jeweils nur die Routen in das Gesamtergebnis ein, die erfolgreich vervollständigt werden konnten. Der Anteil an nicht gefundenen Ergebnissen ist den vorangehenden Graphen zu entnehmen.

Auch diese Betrachtung liefert intuitiv nachvollziehbare Ergebnisse. Mit steigendem Wert von k ist tendentiell mit einer Verschlechterung der durchschnittlichen Dienstqualität zu rechnen. Je nach VP-Heuristik fallen diese Unterschiede jedoch verschieden stark aus. Während der durchschnittliche Qualitätsverlust erfolgreich gefundener Routen bei Verwendung von `R5` nur zwischen 9 Sekunden bei $k = 50$ und 14 Sekunden bei $k = 200$ variiert, gestaltet sich

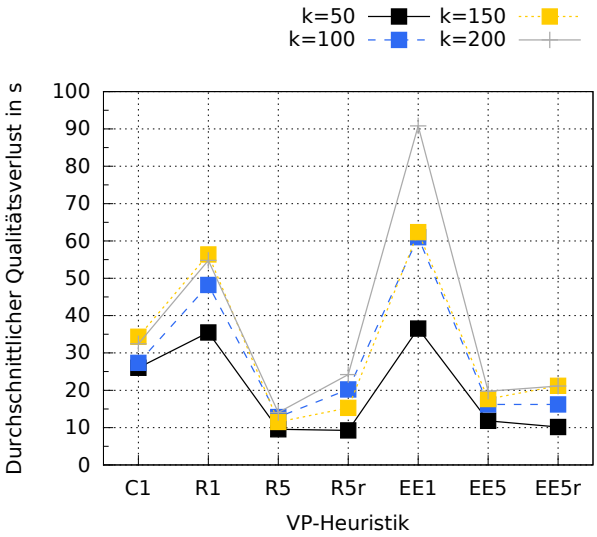


Abbildung 4.10: Durchschnittlicher Qualitätsverlust in der Region München für verschiedene k unter Verwendung von `MB`.

diese Differenz beim Einsatz von `EE1` für dieselben Belegungen von k mit 36 bzw. 90 Sekunden sehr deutlich.

Über alle Betrachtungen hinweg weisen die `EE*`-basierten Strategien zur Auswahl von Verschleierungspunkten somit schlechtere Eigenschaften auf als `C1` und die zufallsbasierten Verfahren. Dies liegt an der Vorgabe eines der Übergangspunkte am Rand der Zone, deren Anzahl mit wachsender Zonengröße ebenfalls zunimmt. Mit größer werdendem k wird daher immer wahrscheinlicher ein Punkt ausgewählt, der nicht der aus Nutzersicht optimale Ein- bzw. Austrittspunkt ist und zwangsläufig einen Umweg verursacht.

Vergleicht man vor dem Hintergrund des durchschnittlichen Qualitätsverlusts wieder die Heuristiken gegeneinander, die nur einen VP auswählen, ist `C1` den anderen Verfahren erneut überlegen. So tritt bei der Verwendung von `C1` im Durchschnitt ein Umweg von 26 bzw. 32 Sekunden für $k = 50$ und $k = 200$ auf, bei `R1` betragen diese Werte bereits 35 und 55 Sekunden.

Welchen Einfluss unterschiedliche Besiedlungsdichten und die Entfernung von Start- und Zieladresse auf die Qualität der privatsphäreschonenden Routenplanung haben, kann den Werten aus Tabelle 4.2 entnommen werden. In jedem untersuchten Szenario zeigt sich der negative Einfluss eines großen Wertes von k auf die Qualität der gefunden Routen. Der Anteil an optimalen Routenergebnissen nimmt für jedes Szenario stetig ab, während der durchschnittlich zu erwartende Umweg zusammen mit k ansteigt. Lediglich bei der Anzahl an nicht korrekt beantwortbaren Routenanfragen scheint keine derartige Korrelation zu bestehen. Unabhängig von k und den an der Routenanfrage beteiligten Regionen schwankt dieser Wert zwischen 5 und 6 %.

Unterschiede lassen sich dafür beim Vergleich der verschiedenen Routing-

Routing-Szenario	k	optimal	⌀Umweg	max. Umweg	∅
in München	50	70,8 %	9,5 s	129 s	6,2 %
	100	67,0 %	12,8 s	124 s	5,4 %
	200	65,4 %	14,0 s	219 s	5,6 %
München→Rosenheim	50	65,0 %	10,7 s	94 s	6,0 %
	100	60,7 %	14,2 s	109 s	6,0 %
	200	53,9 %	19,2 s	349 s	6,0 %
München→Erding	50	80,2 %	6,6 s	57 s	5,2 %
	100	72,4 %	8,9 s	109 s	5,2 %
	200	73,9 %	9,6 s	153 s	6,1 %

Tabelle 4.2: Qualitätseigenschaften der verschleierten Routenanfragen für verschiedene Regionen mit `R5` und `MB`.

szenarien untereinander ausmachen. Zwar zeigen alle dasselbe Verhalten hinsichtlich der Belegung von k, tun dies aber auf deutlich unterschiedlichem Niveau. So wird bei der Routenplanung innerhalb Münchens deutlich öfter (65 bis 70 %) das optimale Ergebnis gefunden als im Szenario *München→Rosenheim*. Auch hier liegt der Anteil der optimalen Routen jedoch stets bei über 50 %. In Übereinstimmung mit diesem Ergebnis fällt auch der durchschnittliche Umweg innerhalb Münchens für alle k niedriger als auf dem Weg nach Rosenheim aus.

Genau entgegengesetzt verhält es sich im Fall *München→Erding*. Hier wird mit Abstand am häufigsten die optimale Route gefunden, und es werden die kürzesten Umwege produziert. Eine Erklärung hierfür ist der verhältnismäßig ländliche Charakter der untersuchten Kleinstadt, zu deren Postleitzahlgebiet auch viele Ziele gehören, für deren Erreichen sich kaum Alternativen bieten – lässt sich die ermittelte Dummy-Route korrekt vervollständigen, handelt es sich daher in den meisten Fällen um das optimale Ergebnis.

Die Ziele innerhalb Münchens liegen stets näher beisammen als Routenendpunkte zwischen München und Erding (im Mittel 40 km). Die Distanz zu Zielen in Rosenheim ist noch einmal knapp um das doppelte größer. Ein direkter Zusammenhang zwischen dem Abstand der Routenendpunkte und der Qualität der verschleierten Ergebnisrouten besteht demnach nicht. Die Ursachen für die Niveauunterschiede müssen in weiteren Arbeiten genauer analysiert werden.

Der Maximalwert für die Dauer eines Umwegs, den eine verschleierte Routenanfrage erzeugt, tritt für $k = 200$ im Szenario *München→Rosenheim* auf. Bei einer Entfernung der beteiligten Routenendpunkte zueinander von über 77 km liegt dieser Wert jedoch immer noch bei unter sechs Minuten. Eine interessante Beobachtung ist, dass die gefunde Strecke in diesem Fall sogar 200 m kürzer ausfällt als die schnellste Strecke. Auch hieran zeigt sich, dass die in in [236] als Gütekriterium verwendete Länge der verschleierten Routenantwort keine geeignete Qualitätsmetrik darstellt.

In allen untersuchten Szenarien liegt der durchschnittlich in Kauf zu nehmende Umweg für $k = 50$ bei unter 11 Sekunden, für $k = 200$ bei weniger als 20 Sekunden. Insgesamt zeigt sich damit, dass sich die privatsphäresche-

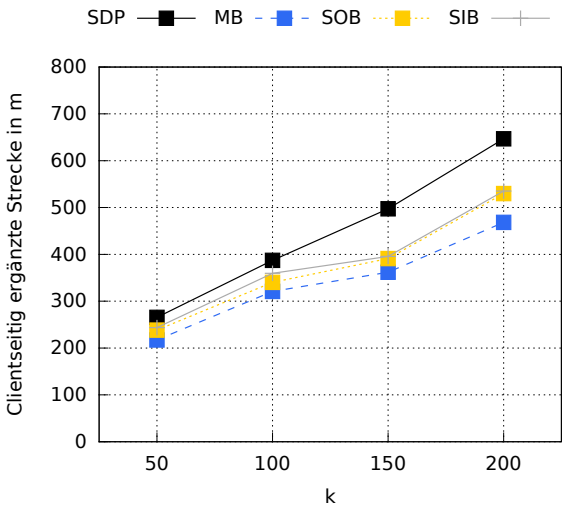


Abbildung 4.11: Länge der clientseitig berechneten Ergänzungsstrecken.

nende Routenplanung auch für größere Distanzen sinnvoll einsetzen lässt und die Umwege, die von ihr verursacht werden, in den meisten Fällen nur lokal auftreten und auf die Start- und Zielzone der Routenanfrage begrenzt sind.

Die letzte Betrachtung hinsichtlich der Dienstqualität der privatsphäreschonenden Routenplanung untersucht die Länge der lokal ergänzten Teilstrecken einer ermittelten Gesamtroute. Die die Vervollständigung von Dummyrouten clientseitig und ohne Wissen um Echtzeit-Verkehrsinformationen durchgeführt wird, sollen diese Strecken idealerweise möglichst kurz ausfallen.

Abb. 4.11 zeigt die durchschnittlichen Gesamtlängen der Strecken in Start- und Zielzone, die pro verschleierter Routenermittlung lokal berechnet werden müssen, in Abhängigkeit von k und für die verschiedenen Konkatenationsstrategien für das Szenario *München→Erding*. Datengrundlage sind alle erfolgreich vervollständigten Routen, die mit der `R5`-Heuristik gefunden werden konnten.

Wie zu erwarten ist, nimmt die Länge dieser Teilstrecken mit wachsendem k ebenfalls zu – je größer die Verschleierungszonen ausfallen, desto größer gestaltet sich auch der Teil des Straßennetzes in Start- und Zielzone, über den keine Echtzeitinformationen vorliegen. Ein aus Sicht der erreichbaren Dienstqualität positiver Aspekt ist erneut, dass auch die durchschnittliche Länge der „blind" berechneten Teilstrecken langsamer anwächst als der Wert von k.

Auch hierbei schneidet die Strategie `MB` – ihrer eigentlichen Zielsetzung entsprechend – am besten ab. Im untersuchten Szenario nimmt die durchschnittliche Länge der clientseitig ergänzten Streckenabschnitte hierbei von 217m bei $k = 50$ auf 468m für $k = 200$ zu. Unter Berücksichtigung der vorangehenden Ergebnisse lassen sich mit diesem Verfahren somit stets die besten Ergebnisse erzielen, da diese Strategie neben den kürzesten lokalen Ergänzungsrouten auch die meisten optimalen Routen erzeugt und am seltensten fehlschlägt.

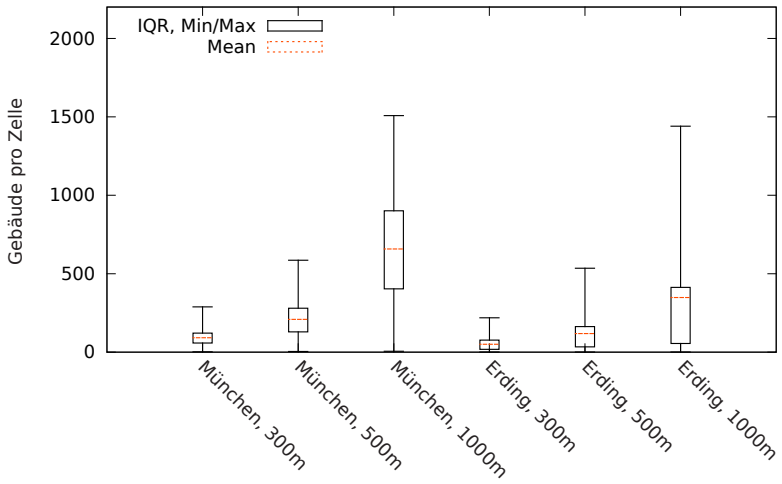


Abbildung 4.12: Anzahl der semantischen Orte pro Zone eines Grids mit variierender Seitenlänge in den Regionen München und Erding.

4.5.2.2 Erreichter Grad an Standortanonymität

Unter dem Einsatz verschiedener Formen der Standortverschleierung, verraten weder der hier vorgestellte Ansatz noch das Verfahren von Vicente et al. [236] die exakten Endpunkte einer Routenanfrage an den Dienstanbieter. Im Gegensatz zu dem Dummy-basierten Ansatz in [160] haben solche verschleierungsbasierten Vorgehensweisen den Vorteil, dass sich der Grad an Privatsphäre nicht allein durch die Anzahl an übertragenen Anfragen steigern lässt, sondern mit der Größe der Verschleierungszonen anwächst. Rein Dummy-basierte Verfahren besitzen darüber hinaus den Nachteil, dass sich wie beschrieben auch stets die tatsächliche Routenanfrage mit dem echten Start- und Zielpunkt in der Verschleierungsmenge befindet. Werden daher wie in [160] zufällige Koordinaten als Dummypositionen ausgewählt, ist es einem Angreifer mit Kartenwissen einfach möglich, echte Routenanfragen von unrealistischen – weil mit zufällig platzierten Endpunkten formulierten – Anfragen zu unterscheiden.

Vicente et al. [236] setzen zum Zweck der Endpunktverschleierung ein Gitternetz mit fester Kantenlänge ein, das über die gesamte Karte gelegt wird. Die Start- und Zieladresse der Routinganfrage werden dadurch in einer vergrößerten, zusammenhängenden Region verborgen, deren Lage aufgrund der statischen Partitionierung keine Hinweise auf den jeweiligen Endpunkt zulässt. Was hierbei nicht berücksichtigt wird, ist der effektiv erreichte Grad an Standortanonymität unter der Annahme, dass es sich bei Start- und Zielpunkt einer Routenanfrage i.d.R. um semantische Orte handelt und Freiflächen unwahrscheinliche Ziele darstellen. Die räumliche Ausbreitung einer Verschleierungszone allein sagt somit nichts über die Anzahl an plausiblen Routenendpunkten innerhalb der Zone aus.

Die privatsphärebezogene Betrachtung einer solchen gleichmäßigen Einteilung der Karte in ein Gitternetz ist in Abb. 4.12 zu sehen. Für die Regionen

München und Erding ist dort der Grad an Standortanonymität gezeigt, der unter der Verwendung verschiedener Kantenlängen $s \in \{300m, 500m, 1000m\}$ erreicht werden kann. Für jedes Szenario wurden 50 zufällig versetzte Partitionierungen der Region in Gitterzellen durchgeführt und die Anzahl an verschiedenen Gebäuden innerhalb jeder Zelle gezählt.

Im Durchschnitt enthalten die einzelnen Zellen eine ausreichende Anzahl an Gebäuden, um die Routenendpunkte des Nutzers darin zuverlässig zu verstecken. Für eine Kantenlänge von 300m ergibt sich z.B. für die Region Erding eine durchschnittliche Zahl von 82,6 Gebäuden pro Zelle.

Die Grafik zeigt jedoch auch, dass ein derartiges Vorgehen diesbezüglich keine Garantien bieten kann: In jedem der untersuchten Szenarien treten Fälle auf, in denen sich trotz großer räumlicher Ausdehnung kaum semantische Orte innerhalb einer Zelle befinden. Dies ist insbesondere an den Rändern von Siedlungsgebieten zu beobachten – so beinhalten einige Zellen in der Region München selbst bei einer Kantenlänge von 1.000m mitunter nur sechs Adressen. Bei 300m treten Fälle mit exakt einer Adresse auf. Derart niedrige Werte sind zudem in ländlichen Gebieten zu erwarten, sodass eine statische Einteilung der Karte gemäß einer solchen Vorgehensweise aus Sicht der Privatsphäre nicht zu empfehlen ist.

Region	**k**	min **c**	∅ **c**	max **c**	∅ **Seitenlänge**
München	50	50	69,7	135	238,8 m
	100	101	137,0	257	447,6 m
	200	200	254,7	556	667,9 m
Erding	50	50	66,1	158	256,0 m
	100	100	129,9	231	410,5 m
	200	200	257,4	465	638,8 m
Rosenheim	50	50	68,1	178	287,5 m
	100	100	135,2	236	447,1 m
	200	200	259,9	538	668,4 m

Tabelle 4.3: Anzahl an Gebäuden und durchschnittliche Seitenlänge pro SZ.

Zur Gewährleistung k-immuner Routenanfragen setzt PrOSPR daher auf die Erzeugung von Verschleierungszonen auf Basis der gebäudebasierten k-Anonymität mit Hilfe des Silent Zone-Verfahrens. Tabelle 4.3 zeigt den Grad an Standortanonymität, der sich damit minimal und im Durchschnitt erreichen lässt, sowie die durchschnittlich zu beobachtende Seitenlänge der erzeugten Zonen. Wie gewünscht, wird der notwendige Wert von k dabei stets zuverlässig erreicht. Die minimal beobachtete Anzahl an Adressen in einer Zone entspricht dabei dem angestrebten Wert für k, der im Mittel in den untersuchten Regionen sogar deutlich überboten wird. Die Seitenlänge der erzeugten Zonen passt sich intelligent der Bebauungsdichte der jeweiligen Nachbarschaft an.

Anhand dieser Ergebnisse kann festgehalten werden, dass der Einsatz eines kartenbasierten, adaptiven Verfahren zur Erzeugung von Verschleierungszonen

aus Sicht der Privatsphäre deutlich besser für die zuverlässige, privatsphärekonforme Umsetzung der online Routenplanung geeignet ist als die in bestehenden Arbeiten verwendeten Ansätze.

4.5.2.3 Qualität der Routenergebnisse mit PrOSPR+

Als nächstes wird untersucht, wie sich der Einsatz der in Kapitel 4.4.2 vorgestellten Erweiterungen von PrOSPR auf Dienstqualität und Kommunikationsaufwand auswirken. Die Verwendung der sog. Must-Have-Components (MHC) stellt sicher, dass die minimale Anzahl an VPs ermittelt wird, die nötig sind, um eine vollständige Route zu garantieren. Dadurch soll erreicht werden, dass mit einer intelligent abgeschätzen Anzahl an Dummyanfragen und sorgfältig ausgewählten Dummy-Endpunkten stets ein korrektes Ergebnis erzeugt werden kann und nur so viele Anfragen wie nötig an den Routenplaner übermittelt werden. Die Transitpunkt-Strategie ist zusätzlich dazu in der Lage, durch die Verwendung von Viarouten überflüssige LBS-Anfragen einzusparen.

Parameter	**Belegungen**
k	50, 100, 150, 200
`BORDER`	50 m
VP-Heuristik	`RMHC`, `MMHC`
Transitpunkt	ja, nein
Routenvervollständigung	`MB`
Routing-Szenario	in München

Tabelle 4.4: Verwendete Parameterbelegungen

In Tabelle 4.4 sind die hierfür verwendeten Parameterbelegungen dargestellt. Wie zuvor wurden pro Parameterbelegung 120 Routen für dieselben, zufällig ausgewählten Endpunkte ermittelt.

Abb. 4.13 zeigt die Fehlerverteilung bei der privatsphäreschonenden Routenplanung mit PrOSPR+ auf Basis der VP-Heuristiken *Random from MHC* (`RMHC`) und *Midmost from MHC* (`MMHC`). Wie zu sehen ist, wird durch den Einsatz der in Kap. 4.4.2.1 vorgestellten *Must-Have-Components* effektiv gewährleistet, dass sich tatsächlich jede Routenanfrage verkehrsregelkonform beantworten lässt. Darüber hinaus können hiermit auch Umwege von mehr als fünf Minuten Länge konsequent vermieden werden.

Hinsichtlich des Anteils an optimalen Routenergebnissen kann hierbei kein direkter Zusammenhang mit dem gewählten Wert von k festgestellt werden. Scheinbar unabhängig von k schwankt der Anteil an optimalen Ergebnissen bei der zufälligen `RMHC`-Heuristik zwischen 26 und 40 %. Die deterministische `MMHC`-Variante, die stets den am weitesten von allen Ein- bzw. Austrittspunkten der MHC entfernten Punkt auswählt, zeigt sich hinsichtlich dieser Eigenschaft über alle Werte von k stabiler als `RMHC` bei 32 bis 36 %.

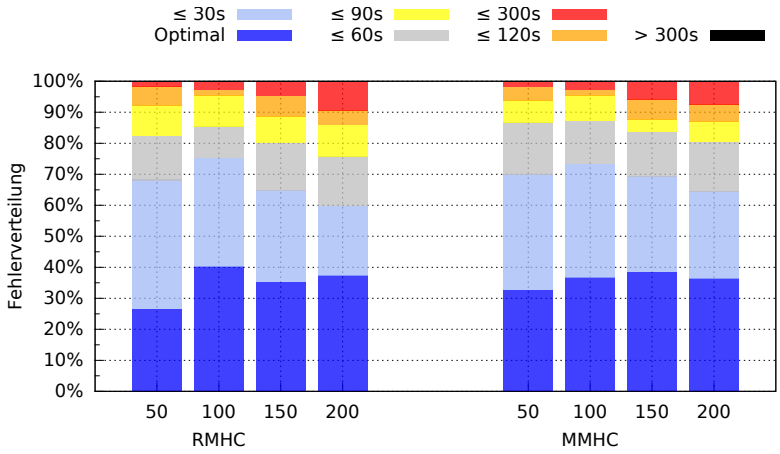


Abbildung 4.13: Fehlerverteilung bei der privatsphäreschonenden Routenplanung mit PrOSPR+ für verschiedene Werte von k.

Heuristik	k	optimal	∅Umweg	max. Umweg
RMHC	50	26,6 %	30,6 s	122 s
	100	40,4 %	27,3 s	252 s
	200	37,4 %	40,1 s	231 s
MMHC	50	32,7 %	27,1 s	123 s
	100	36,7 %	27,5 s	256 s
	200	36,5 %	35,1 s	234 s

Tabelle 4.5: Qualitätseigenschaften der verschleierten Routenanfragen mit PrOSPR+ unter Verwendung von `MB` im Szenario *in München*.

Was den durchschnittlichen Qualitätsverlust betrifft, lässt sich jedoch auch hier ein bereits mehrfach beobachteter Trend wiedererkennen. So nimmt der Anteil an langen Umwegen mit einer Dauer von mehr als zwei Minuten unabhängig von der verwendeten VP-Heuristik mit steigendem k kontinuierlich zu. Der Anteil an verschleierten Ergebnissen mit einem Umweg von unter einer Minute nimmt entsprechend ab. Die besten Ergebnisse ergeben sich interessanterweise nicht für $k = 50$, sondern für $k = 100$. Dies spiegelt sich auch in Tabelle 4.5 wieder, die neben dem Anteil an optimalen Routenergebnissen auch den durchschnittlichen und maximal auftretenden Qualitätsverlust beschreibt.

Insgesamt schneidet auch hier die Heuristik, die einen Verschleierungspunkt im Mittelpunkt der MHC auswählt, besser ab als die zufällige Variante. Die Unterschiede gestalten dabei jedoch bei weitem nicht mehr so deutlich aus wie zuvor: Während `MMHC` einen durchschnittlichen Umweg von 27,1 Sekunden für $k = 50$ und 35,1 Sekunden für $k = 200$ verursacht, liegt `RMHC` mit 30,6 und 40,1 Sekunden jeweils maximal fünf Sekunden darüber. Im Vergleich dazu fällt der Unterschied zwischen `C1` und `R1` für dasselbe Routingszenario mit 32 vs. 55 Sekunden für $k = 200$ weitaus größer aus (vgl. Abb. 4.10).

Die Ursache hierfür ist, dass durch den Algorithmus zur Erzeugung der Must-Have-Components das mögliche Set an Verschleierungspunkten sowohl bei `MMHC` als auch bei der `RMHC`-Variante stark auf dieselbe Menge eingeschränkt ist: Beide Strategien dürfen nur solche VPs wählen, die auf dem Straßennetzwerk der Zusammenhangskomponente liegen, welche die wenigsten gemeinsamen Ausgänge erreicht bzw. die von dem kleinsten Set an gemeinsamen Eingängen in die Zone erreichbar sind. Nur in dem Fall, dass die resultierende MHC mehr als einen möglichen Ein- bzw. Austrittspunkt aufweist, besteht wie zuvor die Möglichkeit, dass unter Verwendung von `MMHC` die Entscheidung, an welchem Punkt die Zonengrenze überschritten wird, stärker von der globalen Routenantwort des LBS-Anbieters beeinflusst ist.

Aus demselben Grund sinkt durch die Verwendung der MHCs auch die Chance, dass der externe Routenplaner die optimale Route zwischen Start- und Zieladresse finden kann. Führt beispielsweise eine Einbahnstraße aus der Startzone heraus, verlässt gemäß des vorgeschlagenen Algorithmus auch die angefragte Dummyroute dieser MHC die Zone zwangweise über diesen Ausgang, da nur dieser von allen Adressen der Komponente erreichbar ist. Handelt es sich dabei nicht zufällig um die vom eigentlich Startpunkt aus gesehen optimale Wahl, wird unausweichlich ein Umweg provoziert.

Insgesamt weisen die mit den `MHC`-basierten Heuristiken erzielten Ergebnisse somit im Durchschnitt über alle getesteten Routen eine geringere Dienstqualität auf als z.B. bei PrOSPR mit `R5`. Die Erklärung hierfür ist, dass bei `R5` mehrere Dummypunkte in der Zusammenhangskomponente des eigentlichen Routenendpunkts liegen können, wodurch die Chance, eine optimale Route zu finden, steigt. Liegt einer der Endpunkte dabei jedoch in einer Komponente, aus der kein VP ausgewählt wird, findet PrOSPR kein vollständiges Ergebnis. Im Gegensatz dazu garantieren die `MHC`-basierten Strategien stets eine gültige Routenantwort. Zudem lässt sich sich damit – wie im nächsten Kapitel gezeigt wird – die Anzahl an Dummyanfragen sinnvoll abschätzen, sodass meist deutlich weniger als die 25 Anfragen von `R5` an den LBS-Anbieter nötig sind.

4.5.2.4 Anzahl an Routenanfragen und Kommunikationsaufwand

Neben den in den vorangehenden Abschnitten untersuchten QoS-bezogenen Eigenschaften stellt auch der Kommunikationsaufwand, der durch die privatsphäreschonende Routenplanung im Vergleich zu einer regulären LBS-Nutzung entsteht, ein wichtiges Kriterium dar. Es wird daher als nächstes untersucht, wie viele Anfragen für die Beantwortung einer online Routenanfrage mit PrOSPR und PrOSPR+ im Schnitt versendet werden müssen. Darüber hinaus wird analysiert, welche Datenmenge im Rahmen der einzelnen Schritte anfällt.

Abb. 4.14 zeigt die durchschnittliche Anzahl an Dummyanfragen, die in der Region München unter Verwendung unterschiedlicher Werte von k und den verschiedenen VP-Heuristiken erzeugt werden. Die anderen untersuchten Regionen zeigen ein nahezu identisches Bild. Ebenfalls dargestellt ist der Einfluss

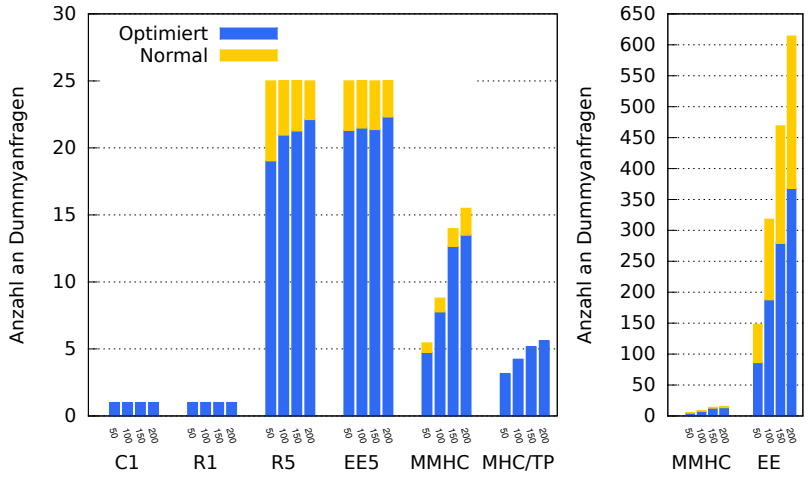


Abbildung 4.14: Anzahl an nötigen Dummyanfragen an den Routenplaner.

der in Kap. 4.4.1.6 vorgestellten Optimierung durch Einsparung implizit beantworteter Dummyanfragen durch vorangehende Routenantworten.

Im Gegensatz zu den trivialen Fällen `C1` und `R1`, die naturgemäß nur eine einzige Dummyanfrage an den online Routenplaner verursachen, erzeugen die `*N`-basierten Strategien bis zu N^2 Anfragen. Einen noch höheren Kommunikationsaufwand verursacht nur die `EE`-Strategie, die Routen für alle Kombinationen an Aus- und Eintrittspunkten von Start- und Zielzone vom Routenplaner ermittelt. Die `EE`-Heuristik, die als einzige dazu in der Lage ist, stets das optimale Routenergebnis zu liefern, bezahlt diese Optimalität mit einer sehr hohen Anzahl an dafür nötigen Dummy-Anfragen. Wie man im rechten Teil der Abbildung sieht, steigt dieser Wert für $k = 200$ auf über 600 Anfragen an. Da hierbei jeder Ein- bzw. Austrittspunkt der beiden Verschleierungszonen als Dummypunkt verwendet wird, wächst die Anzahl an nötigen Routen mit k. Je größer die Zonen ausfallen, desto mehr Straßensegmente schneiden die Zonengrenze und stellen einen Übertrittspunkt der Zone dar.

Während `R1` und `C1` aus Sicht der entstehenden Kommunikationsaufwands eindeutig vorzuziehen sind, sind diese Heuristiken wie zuvor gezeigt im Gegenzug häufig nicht dazu in der Lage, dem Nutzer ein gültiges Routenergebnis zu präsentieren. Einen sinnvollen Kompromiss zwischen Dienstverfügbarkeit und Kommunikationsoverhead stellen somit die `*N`-basierten Verfahren sowie die im Rahmen von PrOSPR+ vorgestellten Heuristiken dar.

Durch Überprüfung, ob eine noch ausstehende Dummyroute durch vorangehende Antworten des LBS bereits implizit beantwortet wurden, lässt sich bei `R5` und `EE5` ein gewisser Anteil an unnötigen Routenanfragen einsparen: So verringert sich die Zahl an angefragten Routen bei `R5` für $k = 50$ durchschnittlich von 25 auf 19,04. Je größer der Wert von k, desto geringer fällt jedoch die Wahrscheinlichkeit aus, dass zwei noch nicht untersuchte Dummy-Endpunkte unmittelbar auf einer extern ermittelten Ergebnisroute liegen. Für $k = 200$

lassen sich somit bei derselben Heuristik nur noch halb so viele Anfragen einsparen und es müssen im Schnitt 22,12 Routen angefragt werden.

Wie Abb. 4.14 zeigt, lassen sich mit Hilfe der vorgeschlagenen Erweiterungen von PrOSPR+ unnötige Anfragen deutlich effektiver einsparen. Die `MHC`-basierten Varianten sind damit nicht nur als einzige dazu in der Lage, stets eine gültige Ergebnisroute zu ermitteln, sondern benötigen hierfür darüber hinaus auch deutlich weniger Routenanfragen. Für $k = 50$ werden unter Verwendung von `MMHC` durchschnittlich nur 5,45 LBS-Anfragen verursacht, was gut einem Viertel der Anfragen entspricht, die von `R5` unter Verwendung der erwähnten Einsparungsmöglichkeit erzeugt werden. `RMHC` weist hierbei logischerweise exakt denselben Wert auf, da beide Verfahren auf demselben Algorithmus zur Erstellung der *Must-Have-Components* basieren und sich nur die Lage der gewählten VPs unterscheiden, nicht aber deren Anzahl.

Zudem fällt auf, dass sich die `MHC`-basierten Auswahlstrategien automatisch dem gewählten Wert für k anpassen. Je größer k, desto höher fällt die Anzahl an zur zuverlässigen Beantwortung einer verschleierten Routenanfrage nötigen Dummyanfragen aus, da Start- und Zielzone in den durchgeführten Experimenten mit zunehmender Größe mehr unterschiedliche MHCs beinhalten. Für die verschiedenen Werte von $k \in \{50, 100, 150, 200\}$ werden dabei im Durchschnitt 5, 8, 13 und 15 Anfragen an den Routenplaner formuliert. Auch der größte beobachtete Durchschnittswert liegt somit noch deutlich unter der von `R5` verursachten Anzahl und selbst für $k = 200$ treten Situationen ein, in denen nur eine einzige Anfrage erzeugt wird. Im Gegensatz zu den VP-Heuristiken von PrOSPR ermöglichen die neuen, `MHC`-basierten Strategien somit die adaptive Ermittlung der nötigen Anzahl und sinnvollen Platzierung von Verschleierungspunkten im Straßennetz von Start- und Zielzone.

Eine weitere Reduzierung der Routenanfragen lässt sich mit Hilfe der Transitpunkt-Strategie erreichen. Auf Basis der `MHC`-basierten Auswahlheuristiken nimmt die Anzahl an angeforderten Dummyrouten dabei auf Werte zwischen 3,13 für $k = 50$ und 5,59 für $k = 200$ ab – ohne dadurch die Garantie, eine gültige Ergebnisroute zu finden, aufzugeben.

Abschließend wird untersucht, welche Datenmenge bei der privatsphäreschonenden Routenplanung mit PrOSPR anfällt. Um eine ad-hoc Einsatzfähigkeit zu gewährleisten, muss das Endgerät des Nutzers im Rahmen einer verschleierten Routenanfrage auch solche Informationen herunterladen, die für das clientseitige Geocoding von Adressen benötigt werden sowie das Kartenmaterial für die ermittelten Start- und Zielzonen. Unter Angabe des gewünschten Postleitzahlgebiets liefert der Kartendienst eine vollständige Liste der Adressen der angefragten Region aus, die jeweils mit dem dazugehörigen WGS84-Koordinatenpaar versehen sind. Die Akquise von Kartenmaterial wird durch Angabe der Bounding-Box der jeweiligen Verschleierungszone ermöglicht.

Hinzu kommt eine variable Anzahl an Dummy-Anfragen, die zur Ermittlung einer gültigen Ergebnisroute an den LBS übermittelt werden müssen. Deren

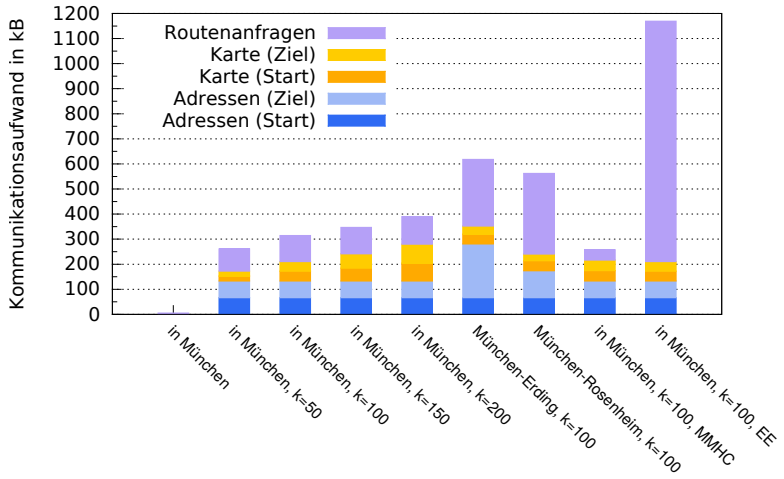


Abbildung 4.15: Durchschnittlicher Kommunikationsaufwand in verschiedenen Routingszenarien mit der `R5`-Heuristik.

Menge wiederum wird – wie soeben gezeigt – stark von der verwendeten VP-Heuristik beeinflusst sowie im Fall der `MHC`-Heuristiken von dem gewählten Wert von k.

In Abb. 4.15 ist der durchschnittliche Kommunikationsaufwand der privatsphäreschonenden Routenplanung dargestellt. Sofern nicht anders gekennzeichnet, beziehen sich die Werte auf die VP-Heuristik `R5` unter Verwendung der beschriebenen Einsparung redundanter Anfragen. Als Referenzwert ist links die durchschnittliche Größe der Routenantwort des LBS auf eine unverschleierte Routenanfrage angezeichnet. Im Rahmen der vorliegenden Arbeit wird davon ausgegangen, dass die Antworten des online Routenplaners neben dem HTTP-Header lediglich die Geometrie der angefragten Route in Form einer Liste von Koordinatenpaaren beinhalten. Für die mittels OSRM ermittelten Routen innerhalb Münchens beträgt die Größe diese Daten im Schnitt 5,1 kB pro Route. Werden dabei zusätzliche Informationen wie Fahrtanweisungen, Straßennamen, etc. übertragen, fällt dieser Anteil des Kommunikationsaufwands entsprechend höher aus und die soeben aufgezeigten Möglichkeiten zur Einsparung von Dummyanfragen gewinnen an Bedeutung.

Das Herunterladen von Geocoding-Informationen für Start- und Zielregion verursacht je nach Region unterschiedlich großen Overhead. Während dies in München im Schnitt 66,55 kB pro PLZ-Gebiet sind, fallen in der Region Erding hierfür 214,49 kB an. Der Grund dafür ist, dass es in der ländlichen Region deutlich mehr Einzelgebäude bzw. Einfamilienhäuser gibt als in der Metropole: Es befinden sich dadurch mehr unterschiedliche Adressen in einem PLZ-Gebiet als in München, wodurch die Menge an Adressinformationen pro PLZ entsprechend größer ausfällt. Rosenheim liegt mit 107,7 kB zwischen den beiden anderen Regionen. Diese Werte sind für alle Werte von k dieselben, da stets das umgebende Postleitzahlgebiet zur initialen Verschleierung dient.

Abhängig von k ist jedoch die Größe der erzeugten Verschleierungszonen und damit auch die Menge an Informationen, die über das jeweils darin enthaltene Straßennetz heruntergeladen werden müssen. So lässt sich beobachten, dass der hierfür anfallende Kommunikationsaufwand innerhalb der Region München von 86,4 kB für $k = 50$ auf 138,1 kB für $k = 200$ kontinuierlich ansteigt. Im Vergleich der Regionen untereinander treten hinsichtlich der Größe des herunterzuladenden Kartenmaterials kleine Unterschiede auf, wobei in München aufgrund des dichteren Straßennetzes erwartungsgemäß die größte Datenmenge anfällt. So beträgt die durchschnittliche Größe des Kartendownloads pro Verschleierungszone für $k = 100$ in München 38,2 kB, in Erding und Rosenheim hingegen nur 32,1 bzw. 26,8 kB.

Die Gesamtgröße der als Antwort auf die Dummyanfragen übermittelten Routenergebnisse bleibt für die verschiedenen Werte von k im Szenario *in München* nahezu unverändert. Es lässt sich jedoch ein Zusammenhang zwischen dem Abstand der Routenendpunkte und des verursachten Kommunikationsaufwands feststellen. So lassen sich im Vergleich der Szenarien *in München*, *München→Rosenheim* und *München→Erding* hinsichtlich der Anzahl an nötigen Dummyanfragen wie erwähnt keine erkennbaren Unterschiede beobachten. In Bezug auf die übertragene Datenmenge macht die Distanz zwischen dem Start- und Zielpunkt jedoch einen intuitiv nachvollziehbaren Unterschied, da bei größerem Abstand jede einzelne Rouenantwort eine längere Liste an Koordinatenpaaren beinhaltet. Für $k = 100$ und `R5` beträgt die akkumulierte Datenmenge der übermittelten Routenantworten 103,6 kB für das Szenario *in München*. *München→Erding* verursacht unter derselben Parametrisierung bereits 265,6 kB, die noch weiter auseinanderliegenden Endpunkte im Szenario *München→Rosenheim* kommen auf 321,1 kB.

Es zeigt sich auch hier der hohe Kommunikationsaufwand der `EE`-Heuristik, der durch die Vielzahl an dabei nötigen LBS-Anfragen verursacht wird. Zudem ist aber auch das Einsparungspotential ersichtlich, das die `MHC`-basierten Varianten von PrOSPR+ bieten. So reduziert sich die durch die Dummyanfragen anfallende Datenmenge für das Szenario *in München* und $k = 100$ von 103,6 kB bei `R5` auf nur 40,9 kB mit `MMHC`. Der gesamte Kommunikationsaufwand einer verschleierten Routenanfrage mit PrOSPR+ beläuft sich in dem untersuchten Szenario durchschnittlich somit auf 257,5 kB – eine Datenmenge, die mit der Geschwindigkeit heutiger mobiler Datenverbindungen gut vereinbar ist und für den Nutzer i.d.R. keine störende Verzögerung bewirken sollte.

4.5.3 Zusammenfassung

In diesem Kapitel wurde mit PrOSPR eine privatsphäreschonende Umsetzung der verkehrsadaptiven online Routenplanung vorgestellt, die nur clientseitige Modifikationen erfordert. Für den Nutzer gestaltet sich der Schutz seiner Privatsphäre dabei überaus einfach – er muss sich lediglich für einen Wert von k entscheiden, der fortan automatisch eingehalten wird.

Für die einzelnen Schritte des Verfahrens wurden verschiedene Umsetzungsvarianten vorgeschlagen und verglichen. Es stellt sich heraus, dass die Strategien `C1`, `R1` und `EE1`, die nur eine Dummyanfrage erzeugen, für den praktischen Einsatz ungeeignet sind, da sie oft kein gültiges Routenergebnis zwischen Start- und Zieladresse finden. Die `*N`- und `*Nr`-basierten Varianten zeigen deutlich bessere Ergebnisse, was die Optimalität der gefundenen Routen betrifft, und schlagen weniger häufig fehl. Diese erzeugen jedoch einen verhältnismäßig hohen Kommunikationsaufwand in der Größenordnung von N^2. Garantien hinsichtlich der Erzeugung einer vollständigen Route geben auch sie nicht.

Das Nicht-Auffinden einer Ergebnisroute wirkt sich sowohl aus Sicht der Dienstverfügbarkeit als auch hinsichtlich der Privatsphäre des Nutzers negativ aus. Eine praktische Umsetzung müsste in diesem Fall eine weitere verschleierte Routenanfrage an den LBS schicken. Dieser erfährt, dass die Route des Nutzers im ersten Durchlauf nicht gefunden wurde, woraus geschlossen werden kann, dass sich Start- oder Zieladresse in einer Graphkomponente der Verschleierungszonen befinden, in der kein VP lag. Der angestrebte Grad an Standortanonymität kann dadurch unter den Wert von k fallen. Derselbe Angriff ist möglich, selbst wenn eine gültige Route gefunden wurde. Gab es Zusammenhangskomponeten in Start- oder Zielzone, in denen kein VP lag, kann der Angreifer diese Bereiche trivial aus der Kandidatenliste an möglichen Endpunkten streichen und den Nutzer somit auch in diesem Fall genauer lokalisieren.

Diese Problematik umgehen die Heuristiken, die im Rahmen der Erweiterung PrOSPR+ vorgestellt wurden. Durch die intelligente Platzierung eines Verschleierungspunktes innerhalb jeder *Must-Have-Component* von Start- und Zielzone wird nicht nur garantiert, dass stets ein gültiges Routenergebnis ermittelt werden kann – es wird darüber hinaus auch sichergestellt, dass tatsächlich jede Adresse, die zu k beiträgt, als Routenendpunkt in Frage kommt.

Es konnte darüber hinaus gezeigt werden, dass unter Verwendung dieser VP-Heuristiken eine gute mittlere Dienstqualität erreicht wird, unnötige Dummyanfragen eingespart werden können und somit deutlich weniger zusätzlicher Kommunikationsaufwand entsteht als bei den `*N`-basierten Heuristiken.

Einen unvermeidlichen Teil des Kommunikationsaufwands stellt die Akquise der Geocoding-Informationen von Start- und Zielregion dar. Neben Abweichungen vom optimalen Routenergebnis in Form kurzer Umwege ist dies der Preis, den der Nutzer für die kartenbasierte Standortverschleierung zur effektiven Verhinderung ortsbasierter Inferenzangriffe in Kauf nehmen muss.

4.6 Topologiebasierte, reziproke Standortverschleierung in LBS

Aufbauend auf den Erkenntnissen, die während des Entwurfs und bei der Auswertung von PrOSPR gewonnen werden konnten, soll nun ein neues Verfahren für die kontinuierliche Verschleierung von Standortinformationen entwickelt

werden: Die vorausschauende, kartenbasierte Standortverschleierung *LAMA LocO* (engl. *Look-Ahead Map-Aware Location Obfuscation*) beinhaltet einen neuen, topologiebasierten Ansatz zur vorausschauenden Erstellung von Verschleierungszonen sowie in sich konsistente Strategien zur dienstübergreifenden Preisgabe von Standortinformationen an verschiedene LBS-Typen.

Sowohl PrOSPR als auch PrOSPR+ basieren wie in Kapitel 4.4.1.3 beschrieben auf der Verwendung von Verschleierungszonen, die gemäß des *Silent Zones*-Verfahrens [253] erzeugt werden. Letzteres garantiert, dass sich innerhalb einer solchen Zone, deren Grenzen im Rahmen der Routenplanung an externe Parteien übertragen werden, eine Mindestzahl an k verschiedenen semantischen Orten befindet, um den tatsächlichen Standort des Nutzers darin zu verbergen. Die Topologie des zugrundeliegenden Straßennetzes wird dabei jedoch nicht berücksichtigt.

Um die für eine sichere Beantwortung einer Routenanfrage nötige Anzahl an Verschleierungspunkten intelligent abzuschätzen, analysieren die MHC-basierten Heuristiken von PrOSPR+ das lokale Straßennetz der Zone hinsichtlich des Vorkommens von starken und schwachen Zusammenhangskomponenten im Graph. Dieser Ansatz soll im Folgenden weiterverfolgt und bereits für eine topologiebezogene Zonenerstellung berücksichtigt werden.

Zudem wird argumentiert, dass die ad-hoc generierten Silent Zones mitunter Schwächen hinsichtlich der von ihnen suggerierten Privatsphäre aufweisen, deren Ursache in der inhomogenen Verteilung von semantischen Orten liegt. Diese Schwachstelle soll im Folgenden durch ein Verfahren zur Erzeugung von reziproken Verschleierungszonen verhindert werden.

4.6.1 Nutzungsszenarien der Standortverschleierung

In diesem Abschnitt werden verschiedene Szenarien beschrieben, in denen Smartphone-Besitzerin Alice regelmäßig unterschiedliche Typen ortsbezogener Dienste nutzt und für die im Folgenden ein durchgängig einsetzbarer Schutzmechanismus entwickelt werden soll. Alice ist meist im Auto unterwegs und hält sich an geltende Verkehrsregeln und Fahrgebote.

Für die Standortermittlung nutzt Alice das GPS-Modul ihres Smartphones oder andere Formen der terminalbasierten Positionsermittlung. O.B.d.A. wird im Folgenden angenommen, dass die verwendete Form der Positionsermittlung keinem übermäßig großen Fehler unterliegt und ihr Endgerät ihren realen Standort somit genau kennt. Die Erkennung und Korrektur möglicher Fehlersituationen ist nicht Bestandteil dieser Arbeit.

Für die Vermeidung von Seitenkanal-Angriffen bei der Kommunikation mit den Dienstanbietern setzt Alice ggf. Werkzeuge wie das *Onion-Routing* [62] ein, so dass die für die Kommunikation verwendeten Netzwerk-Identifikatoren keine Rückschlüsse auf ihren aktuellen Standort zulassen.

- **Szenario 1** Um den schnellsten Weg zu ihrem nächsten Termin zu ermitteln, nutzt Alice, die auch oft in fremden Städten unterwegs ist, sowohl

von ihrer Privat- und Firmenadresse als auch von ihren Kunden aus regelmäßig einen online Routenplaner. Wenn sie merkt, dass eine Erkältung im Anflug ist, sucht sie auch schon einmal mittels eines POI-Finders nach der nächstgelegenen Apotheke. Und wenn sie einen Termin deutlich schneller abschließen konnte als gedacht, lässt sie sich bei Verlassen des Kundens von derselben App gut bewertete Restaurants empfehlen. Diese *sporadischen LBS-Anfragen* verwenden jeweils den aktuellen Standort von Alice, um relevante Ergebnisse zu erhalten.

- **Szenario 2** Alice arbeitet im Außendienst eines Unternehmens und ist beruflich viel unterwegs. Um sich nicht aus den Augen zu verlieren, möchte sie regelmäßigen Kontakt mit ihren Freunden pflegen. Aufgrund ihrer häufigen Dienstreisen eignen sich spontane Treffen daher optimal. Aus diesem Grund nutzt sie zusammen mit ihrem Freundeskreis einen Buddytracker-Dienst, der im Hintergrund läuft und Alice automatisch benachrichtigt, wenn sich einer ihrer Bekannten in unmittelbarer Nähe befindet. Damit dieser Dienst nutzbar ist, benötigt er *in periodischen Abständen* den aktuellen Standort von Alice, um ihn mit den Aufenthaltsorten ihrer Freunde vergleichen zu können.

- **Szenario 3** Da sie viel auf der Straße unterwegs ist, freut sie sich, wenn sie schnell von A nach B kommt. Aus diesem Grund ist sie auch gerne dazu bereit, die mittel- und langfristige Verkehrsplanung durch die Übermittlung ihrer gefahrenen Strecken zu unterstützen. Sie nimmt daher an einer großen Crowdsensing-Kampagne teil, die in regelmäßigen Abständen die einzelnen Fahrten von Alice an eine zentrale Sammelstelle übermittelt. Die dafür nötige Datenerhebung findet mit einer hohen Abtastrate statt, um die Benutzung einzelner Straßensegmente durch die Verkehrsteilnehmer exakt nachvollziehen zu können. Diese *nachträgliche Veröffentlichung vollständiger Trajektorien* findet nicht in Echtzeit statt, sondern z.B. dann, wenn sie gerade mit einem WLAN verbunden ist und ihr somit keine Kosten durch die Datenübertragung entstehen.

- **Szenario 4** Um auch während der Fahrt über aktuelle Verkehrsstörungen und geeignete Umfahrungen informiert zu bleiben, nutzt Alice nach der initialen Routenermittlung die online Navigation des Routenplaners. Auch hierbei wird die Ermittlung der Standortinformationen mit hoher Frequenz durchgeführt. Im Unterschied zu Szenario 3 findet hierbei die *Übertragung der Trajektorie in Echtzeit* statt. Da der Navigationsdienst zudem die Standortupdates seiner Teilnehmer für die Echtzeit-Ermittlung der Verkehrslage verwendet, dürfen hierbei keine gefälschten oder veralteten Ortsinformationen übermittelt werden.

Diese vier verschiedenen Szenarien der LBS-Nutzung beschreiben jeweils unterschiedliche Ausprägungen ortsbezogener Dienste. Je nach LBS-Typ gilt es, unterschiedliche Strategien für die Standortfreigabe zu verfolgen: Während die

sporadischen und periodischen Standortangaben für den Buddytracker oder die POI-Suche nicht zwangsläufig exakt sein müssen, basieren sowohl die online Navigation als auch die nachträgliche Veröffentlichung von Trajektorien auf der Übermittlung von unverfälschten Informationen, die sich eindeutig auf einen Punkt auf dem Strassennetz mappen lassen.

Bei der periodischen und kontinuierlichen Preisgabe von Standortinformationen ist – egal, ob in Echtzeit oder nachgelagert – im Gegensatz zum sporadischen Fall darüber hinaus zu beachten, dass aufeinanderfolgende Updates in Korrelation zueinander stehen. So können z.B. Teilbereiche einer Verschleierungszone aufgrund einer durch zeitliche Constraints bedingte Nicht-Erreichbarkeit seit dem letzten Update [89] oder der nicht vorhandenen Konnektivität im Straßennetz von einem Angreifer definitiv ausgeschlossen werden. Auch derartige Angriffe will LAMA LocO effektiv verhindern.

4.6.2 Zielsetzung und Angreiferbeschreibung

Grundsätzliche Zielsetzung und Angreiferbeschreibung bleiben bei der Realisierung der kontinuierlichen, dienstübergreifenden Standortverschleierung dieselben wie schon bei der Umsetzung der k-immunen Routenanfragen mit PrO-SPR: Es soll verhindert werden, dass ein Angreifer, der in Besitz einer beliebigen Teilmenge der preisgegebenen Standortdaten eines mobilen Nutzers gelangt, die semantischen Orte, an denen sich der Nutzer aufhält oder aufgehalten hat, mit einer Wahrscheinlichkeit größer $\frac{1}{k}$ identifizieren kann. Für jeden Aufenthalt eines Nutzers an einem bestimmten Ort muss demnach auch bei Kenntnis aller veröffentlichten Standortupdates eine Menge von mindestens k realistischer Kandidaten mit derselben Wahrscheinlichkeit in Frage kommen.

Der Parameter k kann dabei vom Nutzer individuell festgelegt werden. Sobald dies geschehen ist, soll sich das zu entwickelnde Verfahren selbständig und proaktiv um die angemessene Verschleierung aller Standorte kümmern, an denen sich ein Nutzer aufhält bzw. aufhalten wird.

Im Vergleich zu der in den vorangehenden Abschnitten angestrebten privatsphäreschonenden Umsetzung der online Routenplanung kommt nun jedoch die Erweiterung hinzu, dass je nach LBS-Typ auch kontinuierlich bzw. periodisch übermittelte LBS-Anfragen zuverlässig gegen ortsbasierte Inferenzangriffe geschützt werden sollen. Dies impliziert, dass neben der räumlichen Dimension der Standortverschleierung nun auch stets die zeitliche Korrelation sukzessiver Standortangaben berücksichtigt werden muss.

Zusätzlich wird im Folgenden angenommen, dass die Nutzung der einzelnen Dienstangebote nicht exklusiv, sondern i.d.R. parallel stattfindet. Alice verwendet z.B. dauerhaft den Buddytracker, fragt vereinzelt benachbarte POI-Informationen und schnellste Routen an und übermittelt im Nachhinein für denselben Zeitraum ihre Trajektorie für die Verkehrsplanung.

Außerdem wird davon ausgegangen, dass alle Dienste von demselben Anbieter betrieben werden oder die an die einzelnen Dienste übertragenen Informa-

tionen von einer Partei zusammengetragen werden können. Der angenommene Angreifer ist somit potentiell dazu in der Lage, alle Standortupdates, die Alice preisgibt, in Erfahrung zu bringen und ihr über ein Pseudonym eindeutig zuzuordnen. Es muss also ein dienstübergreifender Schutz der Aufenthaltsorte von Alice gewährleistet werden. Zudem besitzt der Angreifer wie zuvor umfangreiches Kartenwissen über die Verteilung semantischer Orte, vollständige Kenntnis des Straßennetzes und erlaubter Fahrgeschwindigkeiten und weiss, welche Verschleierungstechniken und Parameterbelegungen der Nutzer verwendet.

4.6.3 Nachteile bei der Verwendung von Silent Zones

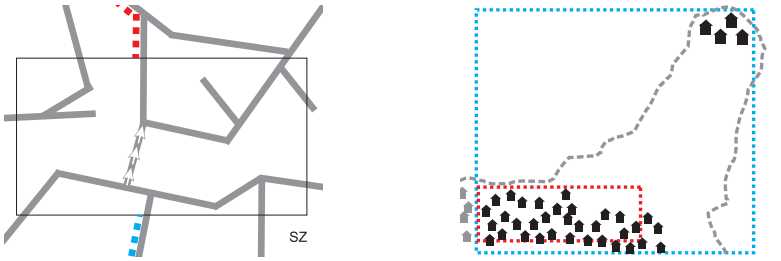
In diesem Abschnitt wird beschrieben, welche Schwächen der Einsatz des Silent Zones-Verfahrens für die Standortverschleierung im Rahmen der LBS-Nutzung und mitunter auch bei seinem ursprünglichen Einsatzgebiet – dem selektiven Ausblenden von Messwerten in Participatory Sensing-Anwendungen – zeigt.

4.6.3.1 Anzahl an nötigen Routenanfragen

Wie bei der Evaluation von PrOSPR beobachtet werden kann, entsteht bei der bislang verwendeten Erstellung der Verschleierungszonen auf Basis des Silent Zones-Ansatzes teilweise eine Vielzahl an nicht oder nur schwach verbundenen Zusammenhangskomponenten. Diese Situation ist in Abb. 4.16a dargestellt und ist auf die Tatsache zurückzuführen, dass die Algorithmen zur Erstellung der Zonen aus [253] zwar die Verteilung semantischer Orte berücksichtigen, nicht jedoch das Straßennetz, das diese Orte verbindet.

Verglichen mit der normalen Nutzung eines online Routenplaners führt dies in vielen Fällen zu einer verhältnismäßig großen Anzahl an Anfragen, die an den online Routenplaner geschickt werden müssen, um die Privatsphäre des Nutzers nicht zu gefährden und seine Routenanfrage dennoch garantiert beantworten zu können. So muss sichergestellt werden, dass aus jeder Zusammenhangskomponente der Startregion eine Route in jede Zusammenhangskomponente der Zielregion ermittelt wird. Andernfalls kann – je nach verwendeter VP-Heuristik – die Anfrage des Nutzers entweder nicht beantwortet werden oder es wird verraten, dass sich Start- und Zieladresse des Nutzers mit Sicherheit nicht in einer der nicht angefragten Zusammenhangskomponenten befinden, wodurch sich die Anzahl an Adresskandidaten reduziert und unter den geforderten Wert von k abrutschen kann.

Die MHC-basierten Verfahren zur VP-Auswahl machen sich diese Erkenntnis bereits zu nutze und sorgen durch Analyse des lokalen Straßennetzes dafür, dass aus jeder Graphkomponente ein Dummypunkt ausgewählt wird. Auch diese intelligenten Heuristiken verursachen abhängig von der Lage und Topologie der jeweiligen Zone jedoch ein Vielfaches des Kommunikationsaufwands einer regulären Routenanfrage.



(a) Mehrere Zusammenhangskomponenten

(b) Inhomogene Verteilung semantischer Orte

Abbildung 4.16: Nachteile des Silent Zone-Verfahrens.

4.6.3.2 Nicht-Erreichbarkeit von Teilbereichen

Für das Einsatzszenario der privatsphäreschonenden online Routenplanung kann die fehlende Konnektivität des lokalen Straßennetzes innerhalb einer Zone durch mehrfache Routenanfragen ausgeglichen werden.

Bei der kontinuierlichen LBS-Nutzung, zu der aufgrund der regelmäßigen Preisgabe von orts- und zeitannotierten Messwerten auch das Participatory Sensing zu zählen ist, kann diese ungünstige Eigenschaft der Verschleierungszonen jedoch bereits unmittelbar zu Problemen hinsichtlich des tatsächlich erreichten Grades an Standortanonymität führen. Sobald ein Nutzer eine Silent Zone betritt, werden weitere Standortangaben unterdrückt bzw. im Rahmen der LBS-Nutzung durch den Mittelpunkt der Zone ersetzt. Besteht das lokale Straßennetz der Zone jedoch aus mehr als einer starken Zusammenhangskomponente, sind gewisse Teilbereiche der Zone nicht erreichbar.

Auch diese Situation ist in Abb. 4.16a zu sehen: Bewegt sich der Nutzer in seinem Fahrzeug entlang der roten Trajektorie in die Verschleierungszone und verweilt dort, kann sein Ziel lediglich eines der oberen Gebäude gewesen sein. Der vom Nutzer festgelegte Anonymitätsgrad k kann dabei je nach Gestalt der Zone und Verteilung der Gebäude deutlich unterschritten werden.

Je nach Konnektivität und Komplexität des Graphen kann es in diesem Zusammenhang unterschiedliche Auswirkungen haben, über welche Kante der Nutzer eine Zone betritt oder verlässt. Betritt der Nutzer die Verschleierungszone entlang der blauen Trajektorie, ist auch die obere Zusammenhangskomponente erreichbar und der Grad an Privatsphäre sinkt weniger stark als zuvor. Für Fußgänger, die sich nicht an Einbahnstraßen und Abbiegeverbote halten müssen, tritt dieser Effekt ebenfalls auf – so scheidet auch in diesem Szenario der Teilgraph links oben in Abb. 4.16a als plausibles Ziel aus.

Sowohl für die Standortverschleierung bei der LBS-Nutzung als auch im Rahmen der Teilnahme an opportunistischen Messkampagnen sollten daher solche Verschleierungszonen um die Aufenthaltsorte der Nutzer gelegt werden, für die die folgende Bedingung zutrifft. Von jedem Eingang in eine Verschleierungszone ausgehend müssen alle plausiblen Ziele innerhalb der Zone erreich-

bar sein und von jedem Punkt in einer Zone muss jeder Ausgang innerhalb des Straßengraphen der Zone erreichbar sein.

4.6.3.3 Ungleiche Wahrscheinlichkeit der Zonenerzeugung

Aus statistischer Sicht lässt sich ein Nachteil des bislang eingesetzten Verfahrens ausmachen, dessen Ursache in der inhomogenen Verteilung der semantischen Orte auf der Karte liegt. Unterschiedliche Adressen in einer Zone weisen je nach der Bebauungsdichte ihrer unmittelbaren Umgebung unterschiedliche Wahrscheinlichkeiten auf, eine Zone bestimmten Ausmaßes zu erzeugen.

Ein Vorteil der Silent Zones gegenüber rein geometrischen Verschleierungstechniken wie z.B. in [11, 59, 89, 148, 236] ist, dass sich die Größe der erzeugten Zonen dynamisch an die Besiedlungsdichte einer Region anpasst. Anders als eine starre Einteilung in ein Gitternetz oder die Erzeugung eines zufällig versetzten Umkreises garantiert dieser Ansatz somit, dass sich tatsächlich immer mindestens k Gebäude in einer Zone befinden. Um eine hohe Dienstqualität und -verfügbarkeit zu ermöglichen, streben die Algorithmen in [253] danach, flächenmäßig möglichst kleine Zonen zu erzeugen, die diese Bedingung erfüllen.

Zuverlässigen Schutz gewährleistet eine solche Kandidatenmenge allerdings nur dann, wenn alle Kandidaten tatsächlich mit derselben oder zumindest sehr ähnlichen Wahrscheinlichkeit Ausgangspunkt einer bestimmt ausgeprägten Verschleierungszone sind. Die vorgeschlagenen Algorithmen zur ad-hoc Erstellung von Silent Zones, die sich stets auf ein bestimmtes Ursprungsgebäude beziehen, können diese Eigenschaft jedoch nicht durchgängig gewährleisten.

Analog zu der naiven Erzeugung von Verschleierungszonen um k verschiedene Nutzer [104, 177] verursachen Ausreißer, d.h. einzelne Gebäude, die wie in Abb. 4.16b zu einem Postleitzahlgebiet gehören, aber außerhalb der dicht besiedelten Kerngebiete liegen, flächenmäßig im Durchschnitt deutlich größere Zonen (in blau dargestellt) als die Orte in dicht bebauten Gebieten (rot). Allein die Größe einer Silent Zone lässt daher u.U. Rückschlüsse auf das Gebäude zu, für das diese Zone erzeugt wurde, bzw. schränkt die Kandidatenmenge ein. Sowohl in der Theorie als auch an den Rändern dichtbesiedelter Gebiete sowie in ländlichen Regionen lassen sich derartige Problemfälle beobachten.

Auf logischer Ebene kann dieses Problem durch die Bedingung der Reziprozität von Verschleierungszonen gelöst werden, ähnlich wie Kalnis et al. es für die Erzeugung von reziproken Anonymitätssets vorschlagen [136]. Dies wird dort über die Bildung von Buckets fester Länge erreicht, die je k Nutzer enthalten. Unabhängig davon, welcher der k Nutzer die Verschleierung verursacht, ist das Set immer gleich. Es werden dort aber weder verschiedene semantische Orte noch die Konnektivität des Straßennetzes beachtet.

Unter Berücksichtigung dieser Aspekte wird im Folgenden eine Lösung für die Erzeugung von Verschleierungszonen vorgestellt, die neben der vom Nutzer geforderten Mindestzahl an k semantischen Orten pro Zone auch das zugrundeliegende Straßennetz berücksichtigt und deren Reziprozität garantiert.

4.6.4 Topologiebezogene Erstellung reziproker Verschleierungszonen

Vor dem Hintergrund der soeben erläuterten Probleme des bisher verwendeten Verfahrens soll nun eine alternative Herangehensweise für die Erstellung sog. *starker Verschleierungszonen* (SVZ) entwickelt werden. Ziel ist es dabei, sowohl die lokale Nichterreichbarkeit von Teilbereichen einer Zone als auch unterschiedliche Wahrscheinlichkeiten der darin enthaltenen Ausgangsadressen, die Zone erzeugt zu haben, zu umgehen.

In Anlehnung an die Argumentation von Damiani et al. wird die Erzeugung der Verschleierungszonen hierbei nicht reaktiv für ein bestimmtes Ausgangsgebäude durchgeführt, da ein derartiges Vorgehen u.U. Rückschlüsse auf den tatsächlichen Aufenthaltsort zulässt [54]. Stattdessen wird vorausschauend in einer Offline-Phase die Partitionierung des gesamten Definitionsbereichs vorgenommen, um die Zonenerstellung vom tatsächlichen Ort des Nutzers zu entkoppeln und die Bedingung der Reziprozität für alle möglichen Ziele zu gewährleisten.

Die eigentliche Transformation des aktuellen Aufenthaltsortes zur Laufzeit besteht dann lediglich in der Ermittlung derjenigen Zone, die diesen Punkt enthält. Die entsprechenden Strategien, wie die erzeugten SVZs während der LBS-Nutzung einzusetzen sind, werden in Kapitel 4.6.5 beschrieben.

Um die Verschleierungszonen im weiteren Verlauf derart zu erzeugen, dass sie sich auch für die Umsetzung der privatsphäreschonenden online Routenplanung eignen, soll o.B.d.A. weiterhin das Postleitzahlsystem als Ausgangspunkt dienen. In Ländern, in denen ein solches System nicht zur Verfügung steht, kann hierfür auf andere vollständige, disjunkte Unterteilungen der Karte wie Gemeinde- oder Verwaltungsgrenzen, etc. ausgewichen werden.

4.6.4.1 Formale Anforderungen an starke Verschleierungszonen

Sei $\mathcal{R}$ eine zusammenhängende Region, deren Umrisse durch ein beliebiges Polygon beschreibbar sind und für die eine Partitionierung in möglichst viele starke Verschleierungszonen $\mathcal{SVZ}_i \in \mathcal{SVZ}(\mathcal{R})$ durchgeführt werden soll.

$B_\mathcal{R}$ ist die Menge aller semantischen Orte in $\mathcal{R}$. $B_\mathcal{R}(\mathcal{SVZ}_i)$ ist die Menge der semantischen Orte in $\mathcal{R}$, die zu Partition $\mathcal{SVZ}_i$ gehören.

$G_\mathcal{R}$ sei das befahrbare Straßennetz der Region. $G_\top$ sei das globale Straßennetz und beinhaltet somit auch Straßen außerhalb von $\mathcal{R}$. $G^-_\mathcal{R}(\mathcal{SVZ}_i)$ ist die *Kerntopologie* des Straßennetzes von $G_\mathcal{R}$, das zu $\mathcal{SVZ}_i$ gehört und das die physischen Zugangsmöglichkeiten zu allen Elementen aus $B_\mathcal{R}(\mathcal{SVZ}_i)$ darstellt. $G^+_\top(\mathcal{SVZ}_i)$ ist das erweiterte Straßennetz von $\mathcal{SVZ}_i$, auf dem jedoch keine Zugangspunkte zu semantischen Orten liegen, die zu $B_\mathcal{R}(\mathcal{SVZ}_i)$ gehören.

$EEP_{in}(\mathcal{SVZ}_i)$ und $EEP_{out}(\mathcal{SVZ}_i)$ sind alle Knoten von $G^-_\mathcal{R}(\mathcal{SVZ}_i)$, die in $\mathcal{SVZ}_i$ hinein oder aus der Zone heraus führen. Diese Knoten haben somit Nachbarn in angrenzenden SVZs und ermöglichen den Übergang zwischen Zonen.

$b \in B_{\mathcal{R}}$ ist ein einzelner semantischer Ort in $\mathcal{R}$, $\mathcal{SVZ}(b)$ die Verschleierungszone, zu der b gehört. $P_{b_p \to b_q}(G)$ ist eine verkehrsregelkonforme, vollständige Route von b_p nach b_q im Straßennetz G.

An eine Partitionierung einer Region $\mathcal{R}$ in starke Verschleierungszonen werden die folgenden formalen Anforderungen gestellt:

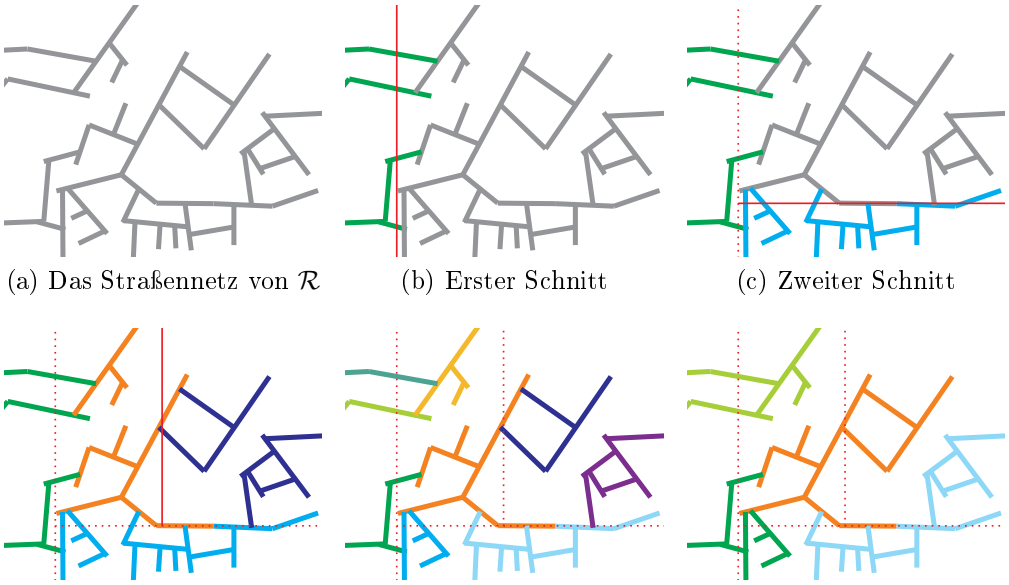
1. $\forall \mathcal{SVZ}_i \in \mathcal{SVZ}(\mathcal{R}) : |B_{\mathcal{R}}(\mathcal{SVZ}_i)| \geq k$
2. $\bigcup_{\mathcal{SVZ}_i \in \mathcal{SVZ}(\mathcal{R})} B_{\mathcal{R}}(\mathcal{SVZ}_i) \equiv B_{\mathcal{R}}$
3. $\forall \mathcal{SVZ}_i, \mathcal{SVZ}_j \in \mathcal{SVZ}, i \neq j : B_{\mathcal{R}}(\mathcal{SVZ}_i) \cap B_{\mathcal{R}}(\mathcal{SVZ}_j) \equiv \emptyset$
4. $\forall b_p, b_q \in B_{\mathcal{R}}(\mathcal{SVZ}_i) : \Pr(\mathcal{SVZ}_i | b_p) \equiv \Pr(\mathcal{SVZ}_i | b_q)$
5. $\forall b_p, b_q \in B_{\mathcal{R}} : b_p \in \mathcal{SVZ}(b_q) \iff b_q \in \mathcal{SVZ}(b_p)$
6. $G_{\mathcal{R}} \equiv \bigcup_{\mathcal{SVZ}_i \in \mathcal{SVZ}(\mathcal{R})} G^-_{\mathcal{R}}(\mathcal{SVZ}_i) \subseteq \bigcup_{\mathcal{SVZ}_i \in \mathcal{SVZ}(\mathcal{R})} G^+_{\top}(\mathcal{SVZ}_i)$
7. $\forall \mathcal{SVZ}_i, \mathcal{SVZ}_j \in \mathcal{SVZ}, i \neq j : G^-_{\mathcal{R}}(\mathcal{SVZ}_i) \cap G^-_{\mathcal{R}}(\mathcal{SVZ}_j) \equiv \emptyset$
8. $\forall b \in B_{\mathcal{R}}(\mathcal{SVZ}_i), \forall e_{in} \in EEP_{in}(\mathcal{SVZ}_i), \forall e_{out} \in EEP_{out}(\mathcal{SVZ}_i) :$
   $\exists P_{e_{in} \to b}(G^+_{\top}(\mathcal{SVZ}_i)) \wedge \exists P_{b \to e_{out}}(G^+_{\top}(\mathcal{SVZ}_i))$

Der in Anforderung 1 verwendete Parameter k ist erneut die minimale Anzahl an unterschiedlichen semantischen Orten, die zum Schutz des tatsächlichen Nutzerstandorts in einer Zone enthalten sein müssen. Anforderungen 2 und 3 besagen, dass durch eine gültige Partitionierung alle Gebäude der Ausgangsregion abgedeckt sein müssen und dass ein Gebäude genau zu einer Partition gehören darf.

Die vierte Bedingung legt fest, dass alle semantischen Orte, die zur selben Zone gehören, dieselbe Wahrscheinlichkeit besitzen müssen, zur Erzeugung der Zone in ihrer aktuellen Form zu führen. Die folgende Anforderung drückt die Reziprozität der Verschleierungszonen aus.

Anforderung 6 bezieht sich auf das Straßennetz der Ausgangsregion und verlangt, dass die Menge aller in $\mathcal{R}$ enthaltenen Straßensegmente gleich der Vereinigung der Kerntopologien aller Partitionen sein muss. Es gibt daher keine Straßensegmente, die nicht Teil einer Zone sind. Zum erweiterten Straßengraphen einer Zone dürfen ggf. jedoch auch solche Straßensegmente gehören, die nicht in $G_{\mathcal{R}}$ liegen. Anforderung 7 besagt, dass die Kerntopologien verschiedener SVZs paarweise disjunkt sein müssen, um eine eindeutige Zuordnung von Orten zu Verschleierungszonen zu ermöglichen. Dies gilt nicht für $G^+_{\top}$.

Die letzte Anforderung bezieht sich auf die gegenseitige Erreichbarkeit von Orten innerhalb derselben Zone durch das Straßennetz der Zone. Hierfür wird gefordert, dass alle semantischen Orte von allen Eingängen sowie alle Zonenausgänge von allen enthaltenen Orten aus erreichbar sind, ohne das erweiterte Straßennetz der Zone dabei verlassen zu müssen.



(a) Das Straßennetz von $\mathcal{R}$ (b) Erster Schnitt (c) Zweiter Schnitt

(d) Letzter Schnitt (e) ZK-Ermittlung (f) Nach Merge-Schritt

Abbildung 4.17: Schrittweise Erzeugung starker Verschleierungszonen.

Als weitere, nichtfunktionale Anforderung an die Zonen wird wie in [253] angestrebt, dass die Zonen möglichst klein ausfallen, um trotz Verschleierung eine hohe Dienstqualität zu erlauben.

Im nächsten Abschnitt wird ein Algorithmus vorgestellt, der auf die Einhaltung dieser Bedingungen achtet und somit die Partitionierung einer Region in starke Verschleierungszonen ermöglicht.

4.6.4.2 Algorithmus zur Erstellung starker Verschleierungszonen

Die optimale Partitionierung eines Graphen gemäß unterschiedlicher Optimierungsziele gehört zur Klasse der NP-vollständigen Probleme (vgl. [38]). Aus diesem Grund wird im Folgenden eine heuristische Herangehensweise zur Partitionierung des Straßennetzes einer Region in starke Verschleierungszonen vorgestellt, welche die Anforderungen aus dem letzten Abschnitt erfüllt.

Algorithmus 4 zeigt die einzelnen Schritte der vorausschauenden, topologiebasierten und reziproken Zonenerstellung in Pseudocode. Eine Visualisierung der wichtigsten Teilschritte des Algorithmus ist in Abb. 4.17 zu sehen.

Die nötigen Eingabeparameter für den Algorithmus sind der vom Nutzer gewünschte Wert von k sowie die geokodierten Adressinformationen B der zu partitionierenden Region $\mathcal{R}$. Darüber hinaus muss auch das lokale Straßennetz $G(V_G, E_G)$ von $\mathcal{R}$ sowie das globale Straßennetz $\top(V_\top, E_\top)$ angegeben werden, die jeweils als gerichteter Graph mit Abbiegeverboten, etc. modelliert sind.

Im ersten Schritt werden die Straßen von $G_\mathcal{R}$ in einzelne Segmente zerlegt und die in B enthaltenen semantischen Orte der nächsten Kante zugeteilt. Um eine möglichst korrekte Zuordnung von Adressen auf den jeweili-

Algorithmus 4 SVZ-Erzeugung

Require: $k > 0 \wedge B \wedge G \wedge \top \wedge maxtries \geq 0$
  mapPlacesToEdges(B, G)
  $partitions \Leftarrow$ createInitialSplit(G, k)

  $connected \Leftarrow \emptyset$
  for all $p \in partitions$ **do**
    $connected \Leftarrow connected \cup$ findCC(p)
  end for

  $connected$.sortByNumPlacesAsc()
  $smallest \Leftarrow connected$.pop()
  while $smallest.NumPlaces < k$ **do**
    $smallest.Neighbors \Leftarrow$ findNeighbors(smallest, connected)
    $mergee \Leftarrow$ selectMergee$(smallest.Neighbors)$
    $mergee \Leftarrow$ merge$(smallest, mergee)$
    $connected$.sortByNumPlacesAsc()
    $smallest \Leftarrow connected$.pop()
  end while

  $result \Leftarrow \emptyset$
  for all $p \in connected$ **do**
    if $p.NumPlaces \geq 2k$ **then**
      $result \Leftarrow result \cup$ split$(p, maxtries)$
    else
      $result \Leftarrow result \cup p$
    end if
  end for

  for all $p \in result$ **do**
    $critical \Leftarrow$ findInlineOnewaySegments$(p.G)$
    for all $c \in critical$ **do**
      $path \Leftarrow$ findShortestBypass$(c.S, \top)$
      $p.ExtraEdges \Leftarrow p.ExtraEdges \cup (path \setminus p.G.Edges)$
    end for
    $p.ExtraEdges \Leftarrow$ addPathsFromEntries$(p.ExtraEdges, critical, p.G, \top)$
    $bans \Leftarrow$ findTurnBans$(p.ExtraEdges \cup p.G)$
    $p.ExtraEdges \Leftarrow$ eliminateTurnBans$(p.ExtraEdges, bans, p.G, \top)$
  end for

  $\tau \Leftarrow$ extractSVZTopologies$(result)$
  $\pi \Leftarrow$ extractSVZPlaces$(result)$
  return τ, π

gen Zugangspunkt im Straßengraphen zu erreichen, verwendet die Methode `mapPlacesToEdges` die Mapping-Funktionalität von OSRM[5], um jedes Gebäude $b \in B$ anhand seiner OSM-Daten der korrekten Kante $e \in E_G$ zuzuweisen. Für jede Kante des lokalen Straßennetzes wird in diesem Schritt gespeichert, welche Gebäude sich über sie erreichen lassen.

Auf Basis dieser Zuordnung wird anschließend eine rein geometrische Partitionierung der Kanten von G parallel zu den Achsen des verwendeten WGS84-Koordinatensystems durchgeführt. Diese findet unter Berücksichtigung von k statt und teilt die von G abgedeckte Fläche rekursiv, abwechselnd horizontal und vertikal so lange auf, bis es keine Partition mehr gibt, deren Kanten zusammen mehr als $2k$ Adressen beinhalten (Abb. 4.17b – 4.17d). Die genaue Position der Schnittkante wird durch Abzählen jeweils so gewählt, dass die Anzahl an abgetrennten Gebäuden nah an einem ganzzahligen Vielfachen von k liegt, um im Idealfall möglichst kleine Zonen erzeugen zu können. Die Strategie *random* wählt dabei ein zufälliges ganzzahliges Vielfaches von k als Schnittkante aus. Bei *optimal* wird der Schnitt gesucht, der von allen den kleinsten Rest modulo k erzeugt. Eine Kante wird zu einer Partition gezählt, wenn ihre minimale x- oder y-Koordinate links bzw. unterhalb der Schnittkante liegt. Die so ermittelten Kanten einer Partition bilden deren Kerntopologie.

Nach diesem Schritt steht eine Partitionierung des Raumes fest, die bereits reziprok ist und bei der jede Zelle mindestens k semantische Orte beinhaltet. Die Topologie des Straßennetzwerks wurde bislang jedoch nicht berücksichtigt. Im nächsten Schritt wird in der Funktion `findCC` mit Hilfe eines Flood-Fill-Verfahrens für alle Partitionen ermittelt, aus welchen (schwachen) Zusammenhangskomponenten (ZKs) sie bestehen (Abb. 4.17e).

Diese ZKs werden nun so kombiniert, dass jede Partition mindestens k Adressen enthält: Dazu werden sie zunächst gemäß der Anzahl ihrer Adressen aufsteigend sortiert. Im Anschluss werden benachbarte ZKs so lange miteinander verbunden, bis es keine Partition mit weniger als k Adressen mehr gibt. Zwei Partitionen sind benachbart, wenn ihre Kerntopologien gemeinsame Knoten aus V_G besitzen. Stehen mehr Kandidaten für dieses Verschmelzen zur Verfügung, wird der Gewinner in der Funktion `selectMergee` gemäß der folgenden Strategie gewählt. Wähle den Nachbarn mit den wenigsten Adressen $a < k$, der nach dem Verschmelzen mehr als k Adressen beinhalten würde. Wenn es keinen solchen gibt, wähle den Nachbarn mit den wenigsten Adressen. Mit dieser Strategie soll die Erzeugung möglichst kleiner Partitionen erreicht werden.

Im `merge`-Schritt werden alle Kanten der bisher kleinsten ZK zu der Kantenmenge des Gewinnerkandidaten hinzugefügt und die Anzahl an enthaltenen Adressen dementsprechend neu berechnet. Nach diesem Prozess stehen wie in Abb. 4.17f Partitionen fest, die (schwache) ZKs darstellen und mindestens k verschiedene semantische Orte beinhalten. Sind durch das Verschmelzen Partitionen entstanden, die mehr als $2k$ Adressen enthalten, wird versucht, diese Zonen an unterschiedlichen x- und y-Positionen – erneut sowohl vertikal als

[5] `http://project-osrm.org/`

(a) Kern-Topologie einer Verschleierungszone (b) Vervollständigung einer Einbahnstraße (c) Vervollständigung eines Abbiegeverbots

Abbildung 4.18: Ermittlung des erweiterten Straßengraphen einer Partition.

auch horizontal – zu teilen. Gelingt es dabei innerhalb einer maximalen Anzahl *maxtries* von `split`-Versuchen nicht, die Zone gültig zu teilen, bleibt die große Partition bestehen und wird so für die Standortverschleierung verwendet.

Im letzten Schritt wird das erweiterte Straßennetz jeder Partition berechnet. Dieser Vorgang ist exemplarisch in Abb. 4.18 zu sehen. Da es sich bei den bisher erzeugten Zonen u.U. nur um schwache ZKs handelt, sind ggf. nicht alle Punkte der Zone über das Straßennetz der Kerntopologie wie gefordert gegenseitig erreichbar. Für die Herstellung der gegenseitigen Erreichbarkeit werden die Überlegungen aus Kapitel 4.4.2.1 aufgegriffen: Verantwortlich für die potentielle Nichterreichbarkeit von Teilbereichen im Straßengraphen einer schwachen ZK sind Einbahnstraßen und Abbiegeverbote.

Die Einbahnstraßen-Problematik wird vermieden, wenn im erweiterten Straßennetz der Zone für jede gerichtete Kante e wie in Abb. 4.18b dargestellt Wege von deren Mittelpunkt $\mathbf{S}$ zu allen Ausgängen der Zone sowie Wege von allen Eingängen zu $\mathbf{S}$ existieren. Abbiegeverboten kann begegnet werden, indem von jeder Kante r zu einer Kreuzung, an der von r kommend ein Abbiegeverbot herrscht, eine Route von einem Punkt $\mathbf{S} \in r$ in der jeweiligen Fahrtrichtung auf die verbotene Folgekante $\mathbf{T}$ innerhalb der Zone existiert (Abb. 4.18c). Die orange eingefärbten Kanten stellen dabei jeweils den erweiterten Straßengraphen $G^{+}_{\top}$ der Zone dar, die grünen Kanten bilden die Kerntopologie $G^{-}_{\mathcal{R}}$.

Bei den Einbahnstraßen müssen in diesem Zusammenhang jedoch nur solche Kanten „repariert", d.h. vervollständigt werden, die selbst den Zugangspunkt zu mindestens einer Adresse der Zone darstellen oder die im Inneren der Zone liegen (engl. *inline*). Führt eine Einbahnstraße ohne Adressen lediglich in die Zone hinein oder nur heraus, hat sie keinen Einfluss auf die gegenseitige Erreichbarkeit von Punkten der Kerntopologie und kann aus Sicht des vorliegenden Problems auf einen punktförmigen Ein- bzw. Ausgang komprimiert werden. Durch das Ausfiltern solcher „irrelevanter" kritischer Stellen wird verhindert, dass z.B. Autobahnauffahrten, die typischerweise solche „leeren" Einbahnstraßen am Zonenausgang darstellen, zur Erzeugung kilometerlanger Ergänzungsrouten führen, die aus Sicht der Privatsphäre überflüssig sind.

Um die Bedingung der gegenseitigen Erreichbarkeit gemäß dieser Überlegungen zu erfüllen, ermittelt der Algorithmus im nächsten Schritt in der Methode `findInlineOnewaySegments()` alle Einbahnstraßen in der Kerntopologie aller Partitionen. Für jede dieser Stellen wird im globalen Straßennetz $\top$ die schnellste Verbindung von **S** über das Ende der Einbahn zurück nach S ermittelt. Sind hieran Kanten beteiligt, die nicht bereits zur Kerntopologie der jeweiligen Zone gehören, werden diese in $G^+_\top$ der Partition gespeichert. Die so gefundenen Routen stellen einen Bypass für die Richtungseinschränkung der jeweiligen Einbahnstraßen dar und sorgen somit sukzessive für die Erzeugung einer großen starken Zusammenhangskomponente. Um Anforderung 8 zu gewährleisten, wird schließlich zunächst in $G^+_\top$, bei Nichterfolg dann erneut in $\top$ ein Weg von jedem Eingang zu allen Einbahnstraßen ermittelt und ggf. zu $G^+_\top$ hinzugefügt. Abschließend werden alle nun in $G^-_\mathcal{R}$ und $G^+_\top$ enthaltenen Abbiegeverbote durch Sicherstellung einer Route von S nach T in $G^+_\top$ in der jeweiligen Fahrtrichtung ebenfalls „aufgehoben".

Das Endergebnis dieses Algorithmus ist eine gültige Partitionierung von $\mathcal{R}$ in starke Verschleierungszonen. Die Ausgabe besteht aus einer Liste τ an Graphen, die für jede Partition die Kerntopologie und den erweiterten Straßengraphen beinhalten sowie einer zweiten Liste π, welche die zugehörigen geokodierten Adressinformationen pro SVZ speichert.

4.6.5 Strategien zur Standortfreigabe in verschiedenen LBS-Typen

Die soeben beschriebene Partitionierung der von einem Nutzer besuchten Regionen in starke Verschleierungszonen kann auf Basis des gesamten Kartenmaterials lokal im Voraus berechnet werden. Alternativ ist vorstellbar, dass die Zonenbeschreibungen unter Angabe der betreffenden Region $\mathcal{R}$ und dem Wert für k auch ad-hoc von einem Kartenanbieter heruntergeladen werden können. Insbesondere in Community-basierten Kartenprojekten wie OSM stellt dies eine realistische Annahme dar, da entsprechende Regionsinformationen einfach als zusätzlicher Relationstyp pro Postleitzahl gespeichert werden können.

O.B.d.A. wird im Rahmen der vorliegenden Arbeit angenommen, dass es einen solchen Kartenanbieter gibt, der neben normalen Karten- und Geocodinginformationen auch die privatsphärebezogene Partitionierung des Straßengraphen in SVZs für verschiedene k kennt und auf Anfrage ausliefert. Auch dieser stellt jedoch keine vertrauenswürdige Partei dar, da der Nutzer zu keiner Zeit seinen tatsächlichen Aufenthaltsort an diese Komponente übermittelt.

4.6.5.1 Durchführung der Standortverschleierung

Um lokal ermitteln zu können, in welcher SVZ man sich gerade befindet, wird im Rahmen von LAMA LocO davon ausgegangen, dass die Liste τ für das aktu-

ell vom Benutzer besuchte Gebiet auf dem Endgerät des Nutzers vorliegt. Für das gesamte Stadtgebiet von München mit 74 Postleitzahlgebieten beträgt die Größe dieser Daten im unkomprimierten, original XML-Format von OSM für $k = 50$ bei 38MB, was für ein modernes Smartphone mit mehreren Gigabytes an Speicherkapazität kein Problem darstellt. Für die Akquise der Partitionen neuer Regionen auf Basis von GPS-Koordinaten wird davon ausgegangen, dass wie bei PrOSPR die Datei L_{PLZ} vorliegt, die anhand der gespeicherten Umrisse aller PLZ-Gebiete ein rudimentäres, clientseitiges Geocoding ermöglicht.

Der Kartenanbieter liefert auf Anfrage sowohl die Liste τ aus, in der die Straßengraphen der einzelnen Partitionen gespeichert sind, als auch die Aufteilung aller Adressen auf diese Partitionen π. Egal, ob der aktuelle Standort des Benutzers oder der im Rahmen von Routenanfragen gesuchte Zielort als Adresse oder in Form von GPS-Koordinaten gegeben ist, ist somit die eindeutige Ermittlung der entsprechenden SVZ möglich.

LBS	**Frequenz**	**Auflösung**	**Echtzeit**
POI-Suche	sporadisch	mittel	✓
Buddytracker	periodisch	mittel	✓
Routenplanung	sporadisch	hoch	✓
Crowdsensing	kontinuierlich	hoch	
Online Navigation	kontinuierlich	hoch	✓

Tabelle 4.6: Klassifikation von LBS-Typen nach ihren Anforderungen

LAMA LocO zielt darauf ab, auch periodische und kontinuierliche Standortupdates derart zu verschleiern, dass für die von einem Angreifer beobachtbaren Aufenthaltsorte eines Nutzers stets mindestens k Gebäude in Frage kommen. Hierfür werden in den nächsten Abschnitten entsprechende Strategien für unterschiedliche LBS-Typen vorgestellt. Auf abstrakter Ebene wird dabei zwischen den Diensttypen und Anforderungen aus Tabelle 4.6 unterschieden.

Da pessimistisch davon ausgegangen wird, dass ein Angreifer dazu in der Lage ist, in Besitz aller von einem Nutzer übermittelten Informationen zu gelangen, müssen diese Strategien hinsichtlich der Preisgabe von Standortinformationen an verschiedene LBS auch dienstübergreifend konsistent sein.

4.6.5.2 Nutzung sporadischer und periodischer LBS

Sowohl die Nutzung einer POI-Suche als auch die im Hintergrund laufende Verwendung eines Buddytrackers benötigen keine exakten Standortangaben. Stattdessen kann der Nutzer hier zwischen QoS und Standortanonymität abwägen, ohne all zu große Einbußen hinsichtlich der Benutzbarkeit des Dienstes erwarten zu müssen. Bei der Verwendung von LAMA LocO werden die Standorte eines Benutzers zum Schutz vor ortsbasierten Inferenzangriffen bei der sporadischen LBS-Nutzung daher grundsätzlich nur auf Ebene der Verschleierungszone preisgegeben. Für sporadische LBS genügt es, die Zone $\mathcal{SVZ}_i$ zu ermitteln, in dessen Kerntopologie $G^-_{\mathcal{R}}(\mathcal{SVZ}_i)$ sich der Nutzer gerade aufhält.

Algorithmus 5 SporadicRelease

Require: $pos \wedge \tau$
$\tau' \Leftarrow \text{getCoreEdges}(\tau)$
$svz \Leftarrow \text{findSurroundingZone}(pos, \tau')$
return $svz.Center$

Algorithmus 5 zeigt den Pseudocode für diese simple Strategie. Zur effizienten Zonenermittlung kann die Geometrie dieser Graphen in einem R-Baum [110] gespeichert werden. Dies ermöglicht die performante Ermittlung der aktuellen Verschleierungszone $\mathcal{SVZ}_i$, wenn der Standort des Nutzers in Form von GPS-Koordinaten vorliegt. Es wird dabei die Zone ermittelt, auf deren Straßennetzwerk sich der Nutzer gerade befindet oder zu einer dessen Kanten er den geringsten Abstand besitzt.

Anstelle des exakten Nutzerstandorts werden dabei stets die Koordinaten des Gebäudes zurückgegeben, das am nächsten zum geographischen Mittelpunkt aller Gebäude der Zone liegt. Dieser Punkt kann einfach durch ein entsprechendes Tagging im Straßennetz τ der Zone hinterlegt werden. Da dieser unabhängig vom tatsächlichen Standort des Nutzers innerhalb der Zone gewählt wird, lässt dies keinen Rückschluss auf den tatsächlichen Standort innerhalb der Zone zu. Dank der Reziprozitätseigenschaft der Zonen ist somit kein Aufenthaltsort wahrscheinlicher als ein anderer.

Bei dieser einfachen Strategie werden jedoch noch keine wiederholten bzw. kontinuierlichen Standortupdates berücksichtigt, aus denen sich eine zeitliche Korrelation ergibt. Lässt sich aus den einzelnen Orten jedoch eine Bewegung des Nutzers nachvollziehen, muss auch dieser Aspekt berücksichtigt werden.

4.6.5.2.1 Kontinuierliche Standortverschleierung Aus diesem Grund wird für die Nutzung periodischer LBS, die keine exakten Standortangaben benötigen, die in Algorithmus 6 beschriebene Strategie vorgeschlagen. Dabei wird davon ausgegangen, dass parallel zur kontinuierlichen Standortverschleierung auch ein Verfahren aus dem Bereich der Kontexterkennung läuft (vgl. Kapitel 2.1.3.1), das auf Basis von Sensordaten erkennt, ob sich der Nutzer in Bewegung befindet oder ob gerade ein Aufenthalt stattfindet.

Die Ermittlung der aktuellen Standortverschleierung muss hierbei nicht nur die aktuelle Position des Nutzers berücksichtigen, sondern zudem auch die früheren bzw. bereits veröffentlichten Standortangaben. Da davon ausgegangen wird, dass der Nutzer u.U. nachträglich seine vollständige Trajektorie z.B. im Rahmen einer Crowdsensing-Kampagne in hoher Auflösung preisgibt, stellt die tatsächliche Trajektorie den Ausgangspunkt einer realistischen Verschleierung dar. Diese wird daher nun als kontinuierlicher Prozess aufgefasst, der selbst in periodischen Abständen die aktuelle Nutzerposition übergeben bekommt und daraus einen privatsphärekonformen *Report* erzeugt. Für einen solchen Report wird sowohl die konnektivitätsbasierte Erreichbarkeit aller Punkte innerhalb

Algorithmus 6 CreatePeriodicReleaseList

Require: $pos \wedge t \wedge \tau \wedge realC \wedge reportC \wedge activeCovers \wedge staydetected \wedge delay$

```
currCell ⇐ findSurroundingZone(pos, getCoreEdges(τ))
lastCell ⇐ realC.last()
reportCell ⇐ ∅

if currCell.id ≠ lastCell.id then
    currCell.entryTime ⇐ t
    if lastCell.hadStay then
        delay ⇐ calculateDelay(lastCell.entryTime, t, τ, delay)
    end if
    realC.append(currCell)
end if

for all cell ∈ activeCovers do
    if pos ∉ cell.extraEdges ∧ cell ≠ currCell then
        activeCovers.remove(cell)
    end if
end for
if pos ∈ lastCell.extraEdges then
    if not reportC.last().hadStay then
        activeCovers.append(lastCell)
    end if
end if

reportCell ⇐ currCell
for all cell ∈ activeCovers do
    if pos ∈ cell.extraEdges ∨ cell = currCell then
        reportCell ⇐ cell
        break
    end if
end for

/* (Algorithmus 8) */

if staydetected then
    realC.last().hadStay = true
    reportCell ⇐ currCell
    activeCovers ⇐ currCell
end if

if reportCell ≠ reportC.last() then
    reportCell.entryTime ⇐ t + delay
    reportC.append(reportCell)
end if
reportC.last().hadStay ⇐ reportC.last().hadStay ∨ staydetected
```

des erweiterten Straßengraphen der Zone als auch die zeitliche Erreichbarkeit aller Orte innerhalb einer Zone berücksichtigt.

In natürlich Sprache arbeitet der Algorithmus wie folgt: Beim Eintreffen eines neuen Standortupdates wird zunächst geprüft, ob seit dem letzten Update ein Zonenwechsel stattgefunden hat. Falls ja, wird die neue Zone samt dem echten Eintrittszeitpunkt `t` in der Liste `realC` abgelegt.

Wenn zudem ein Aufenthalt in der soeben verlassenen Zone stattgefunden hat, wird auf Basis des in τ gespeicherten Straßengraphen geprüft, ob der Aufenthalt angesichts der bekannten Ein- und Austrittszeitpunkte realistisch an jedem semantischen Ort in der Zone stattgefunden haben kann. Diese Bedingung ist wichtig, um keinen Ort aufgrund zeitlicher Constraints aus dem Anonymitätsset auszuschließen. Ein vergleichbare Argumentation verfolgt [260], jedoch unabhängig davon, ob überhaupt ein Aufenthalt stattgefunden hat.

Die Methode `calculateDelay()` berechnet auf Basis der tatsächlichen Aufenthaltszeit in der Zone, der maximalen Fahrzeit in deren Graph und des aktuell u.U. bereits gültigen Delays die Verzögerung, die für das Verbergen eines verräterischen Aufenthalts ab sofort auf die weiteren Standortangaben addiert werden muss. War die Gesamtaufenthaltszeit in der Zone so lang, dass alle Orte realistisch als Aufenthaltsort in Frage kommen, ist kein Delay nötig. `delay` ist eine globale Variable, die ihren Wert bei Verlassen der Methode behält.

Der übrige Teil der Algorithmus kümmert sich um die Behandlung des erweiterten Straßengraphen einer Zone. Damit ein Angreifer keine Teilbereiche der Zone ausschließen kann, die innerhalb der Kerntopologie nicht aus jeder Richtung erreichbar sind, enthält der zu veröffentlichende Report so lange die jeweilige Zone, bis der Nutzer entweder den erweiterten Straßengraphen verlässt oder ein Aufenthalt in einer anderen Zone erkannt wird. Findet ein Aufenthalt in einer Zone statt, kann der Empfänger der Standortdaten somit nicht entscheiden, ob dieser in einem direkt erreichbaren Teil der Zone liegt oder ob der Nutzer die Kanten des erweiterten Straßennetzes genutzt hat, um an durch Einbahnstraßen in der Kerntopologie unerreichbare Orte zu gelangen.

Algorithmus 7 RetrieveReportZone

Require: $t \wedge reportC$
  sortByEntryTimeAsc($reportC$)
  $result \Leftarrow \emptyset$
  for all $cell \in reportC$ **do**
    if $cell.entryTime \leq t$ **then**
      $result \Leftarrow cell$
    end if
  end for
  return $result.Center$

Algorithmus 7 beschreibt das Gegenstück der Reporterzeugung und ermittelt in Abhängigkeit des eigentlichen Anfragezeitpunkts, welche Zone im Rahmen

Abbildung 4.19: Inferenzangriffe auf Basis genommener „Umwege".

der sporadischen oder periodischen LBS-Nutzung derzeit als Standort anzugeben ist. War der letzte Aufenthalt in einer Zelle lang genug, dass aktuell keine Verzögerung nötig ist, wird stets die Zone zurückgegeben, auf dessen Kern- oder erweiterten Straßengraphen sich der Nutzer gerade befindet.

4.6.5.2.2 Standortverschleierung mit bekanntem Ziel Eine weitere Variante der kontinuierlichen Standortverschleierung ergibt sich, wenn der nächste Aufenthaltsort des Nutzers bekannt ist. Dies kann entweder durch manuelle Eingabe erfolgen oder durch die modellbasierte Vorhersage des nächsten Ziels wie in [210], was aufgrund des repetitiven Charakters menschlicher Mobilität gut funktioniert [98, 224].

Algorithmus 8 PeriodicReleaseWithLookAhead

Require: $pos \wedge \tau \wedge ... \wedge start \wedge target \wedge n$
  /* wie Algorithmus 6 bis Kommentar */
  if $\mathrm{d}(pos, start) < \mathrm{thShortcut}(start, n)$ **then**
    $reportCell \Leftarrow start$
  else
    if $\mathrm{d}(pos, target) < \mathrm{thShortcut}(target, n) \vee reportC.\mathrm{last}() = target$ **then**
      $reportCell \Leftarrow target$
    end if
  end if
  /* weiter wie Algorithmus 6 */

In Algorithmus 8 ist eine erweiterte Strategie beschrieben, die davon ausgeht, dass neben dem letzten Aufenthaltsort auch das nächste Ziel des Nutzers bekannt ist. Um gegen einen weiteren möglichen Angriffsvektor gewappnet zu sein, unterdrückt diese Strategie sowohl nach dem Verlassen der Startzone als auch vor dem Betreten der Zielzone die Preisgabe der tatsächlichen Aufenthaltszelle, um womöglich verräterische lokale Umwege zu verheimlichen. Um parallel dazu unvorhergesehene Aufenthalte zu verschleiern, verwendet dieser Algorithmus auch alle Konzepte von Algorithmus 6.

Diese Situation ist aus Angreifersicht exemplarisch in Abb. 4.19 dargestellt. In die Zielzone des Nutzers führen mehrere Eingänge. Der Nutzer bewegt sich entlang der blauen Trajektorie, lässt dabei die ersten beiden Eingänge in die Zone aus und nimmt stattdessen den obersten. Erst dann verschwindet er aus Sicht des Angreifers in der grünen Zone. Selbst auf Zonenebene (orange → grau → grün) ist dies für den Angreifer ersichtlich, spätestens aber, wenn der Nutzer nachträglich seine vollständige Trajektorie veröffentlicht sollte. Beim Verlassen einer Aufenthaltzone lässt sich dasselbe Phänomen beobachten.

Zwar kann ein Angreifer aufgrund der vollständigen Erreichbarkeit aller semantischen Orte durch das erweiterte Straßennetz der Zone hier keine definitiven Aussagen treffen. Insbesondere bei mehrfach besuchten Zielen steigt aus Angreifersicht jedoch schrittweise die Wahrscheinlichkeit, dass sich das Ziel des Nutzers nicht in dem Teil der Zone befindet, der über die unteren Eingänge deutlich schneller erreicht würde.

Um derartige Inferenzmöglichkeiten zu verhindern, gibt diese vorausschauende Strategie nach dem Verlassen einer Zone, in der ein Aufenthalt stattgefunden hat, noch so lange die Startzone als Standort aus, bis sich der Nutzer eine vordefinierte Distanz $th_{shortcut}$ von allen Ausgängen der Zone wegbewegt hat. Hierfür wird die minimale euklidische Distanz zu den Ein- bzw. Ausgängen der Start- und Zielzone verwendet. Analog wird vor dem Erreichen der Zielzone verfahren. Ab dem Moment, in dem sich der Nutzer bis auf $th_{shortcut}$ den Eingängen der Zielzone genähert hat, wird diese als Standort ausgegeben.

$th_{shortcut}$ wird im Rahmen dieser Arbeit als ein ganzzahliges Vielfaches n der maximalen Wegstrecke innerhalb der jeweiligen Zone definiert. Letzteres bewirkt, dass die lokale Wegewahl und „Umwege" in der Nachbarschaft der Aufenthaltszonen vor dem Angreifer verborgen werden. Je größer n gewählt wird, desto weiträumiger findet dieses Ausblenden somit statt.

4.6.5.3 Online Routenplanung

Auch bei der online Routenplanung handelt es sich um einen sporadisch genutzten LBS. Im Gegensatz zu einer POI-Suche benötigt diese jedoch exakte Positionsangaben, um eine vollständige Route vom Startpunkt zum Ziel des Nutzers erzeugen zu können. In diesem Abschnitt wird erklärt, welche Anpassung am Ablauf von PrOSPR vorgenommen werden müssen, um die privatsphäreschonende Routenplanung auf Basis der neuen SVZs zu realisieren.

Die Ermittlung der Verschleierungszone, die den Startpunkt der Routenanfrage enthält, geschieht analog zum vorigen Abschnitt. Für Routenendpunkte, die in Form von Adressen vorliegen, fragt der Client den Kartenanbieter über Angabe der jeweiligen Postleitzahl und dem Parameter k – falls noch nicht lokal vorhanden – nach dem Kartenmaterial τ sowie der Adressen-zu-Partitionen-Zuordnung π der Region. Der Kartenanbieter übernimmt damit auch im Rahmen einer SVZ-basierten Umsetzung nicht die Rolle eines vertrauenswürdigen Dritten, da er wie zuvor nur erfährt, in welchem Postleitzahlengebiet sich die Start- und Zieladresse des Nutzers befinden.

Auf Basis der in π und τ enthaltenen Informationen können die entsprechenden SVZs ermittelt und die exakt geokodierten Adressinformationen unter Angabe der Partitions-ID vom Kartenanbieter heruntergeladen werden.

Die weiteren Schritte gestalten sich analog zu PrOSPR. Ein maßgeblicher Unterschied ist jedoch, dass aufgrund der topologiebasierten Erstellung der Verschleierungszonen nun auch im Rahmen der privatsphäreschonenden Routenplanung nur eine einzige Anfrage an den LBS geschickt werden muss, um eine vollständige Route zwischen Start- und Zieladresse zu erhalten.

Als Verschleierungspunkte kommen demnach gemäß der in Kapitel 4.4.1.5 eingeführten Auswahlheuristiken entweder ein zufälliger Punkt der Kerntopologie (`R1`) oder der dem geografischen Mittelpunkt der Zone am nächsten liegende Punkt (`C1`) in Frage. Auch die lokale Routenervollständigung funktioniert wie zuvor. Wie die Verwendung der starken Verschleierungszonen dabei im Vergleich zu PrOSPR abschneidet, wird in Kapitel 4.7 untersucht.

4.6.5.4 Online Navigation

Für das nächste Szenario wird angenommen, dass der Nutzer im Anschluss an die Ermittlung der derzeit schnellsten Route auch die live Navigation des verkehrsadaptiven Routenplaners nutzen möchte. Dabei überträgt der Nutzer während seiner Fahrt in kurzen, periodischen Abständen seine aktuelle Position an den LBS. Dieser verwendet die kontinuierlichen Standortupdates des Nutzers u.a., um auf Basis dieser Informationen auch seine eigene Abschätzung der Verkehrslage am Ort des Nutzers zu aktualisieren.

Der Nutzer ist bereit, unterwegs diese hochauflösenden Daten zur Verfügung zu stellen, möchte jedoch nicht, dass der Start- und Zielort dadurch verraten werden. Grundsätzlich sind diese beiden Orte durch die Verwendung der im letzten Abschnitt beschriebenen privatsphäreschonenden Routenplanung mit starken Verschleierungszonen vor derartigen Angriffen geschützt.

Eine einfache Herangehensweise wäre es, mit der Übermittlung der exakten Positionsangaben erst zu beginnen, nachdem der Nutzer die Startzone verlassen hat und aufzuhören, wenn die Zielzone betreten wird. Das vorhin beschriebene Problem verrätischer lokaler Umwege stellt sich hier nicht, da zwischen den Zonen – unabhängig davon, welche der Orte in Start- und Zielzone die Routenendpunkte darstellen – den Routenangaben des LBS gefolgt wird.

Ein Problem stellt hierbei aber die zeitliche Korrelation der initialen Routenanfrage mit dem Losfahren dar. Je nachdem, an welchem Ort innerhalb der Startzone sich der Startpunkt der Route befindet, dauert es unterschiedlich lange, bis der vom Routenplaner angegebene Ausgang der Zone erreicht ist und ermöglicht es dem Angreifer dadurch, gewisse Teilbereiche der Startzone auszuschließen. Hierfür wird die folgende Lösung vorgeschlagen.

In Algorithmus 9 werden zunächst anhand des lokalen Straßennetzes der Startzone die minmal und maximal nötige Fahrzeit innerhalb der Zone zu dem vom Routenplaner ermittelten Ausgang berechnet. Als nächstes wird die Zeit berechnet, die man von der eigenen Startposition bis zum Erreichen dieses

Algorithmus 9 LiveNavigationConfig

Require: $startpos \wedge exit \wedge \tau$
  $t_{max} \Leftarrow \text{calcMaxReachTime}(\tau, exit)$
  $t_{min} \Leftarrow \text{calcMinReachTime}(\tau, exit)$
  $t_{self} \Leftarrow \text{calcReachTime}(startpos, \tau, exit)$
  $relativePosition \Leftarrow (t_{self} - t_{min})/(t_{max} - t_{min})$
  $\delta t_{max} \Leftarrow t_{max} - t_{self}$
  $\delta t_{min} \Leftarrow t_{self} - t_{min}$
  return $relativePosition, \delta t_{max}, \delta t_{min}$

Ausgangs benötigt. Aus diesen Werten lässt sich abschätzen, an welcher relativen Position der Nutzer die Zone über diesen Ausgang verlassen würde, wenn in diesem Moment an jedem Ort in der Zone ein Fahrzeug losfahren würde. Zudem kann abgeschätzt werden, wie viel Zeit δt_{min} vor dem Nutzer der nächstgelegene Ort den Ausgang erreicht haben kann und wie viel später δt_{max} der hinterste auch dort angekommen sein kann.

Algorithmus 10 LiveNavigationRelease

Require: $pos \wedge t \wedge t_0 \wedge lastpos \wedge dist \wedge \delta t_{max} \wedge \delta t_{min} \wedge th_v \wedge startz \wedge targetz$
  if $pos \in startz$ **then**
    return $startz.Center$
  end if
  if $t_0 = \emptyset$ **then**
    $t_0 \Leftarrow t$
  end if
  $t \Leftarrow t - t_0$
  $dist \Leftarrow dist + \text{d}(lastpos, pos)$
  if $t > \delta t_{max} \wedge dist > 0$ **then**
    $v_{min} \Leftarrow dist/(t + \delta t_{min})$
    $v_{max} \Leftarrow dist/(t - \delta t_{max})$
    if $v_{max} - v_{min} \leq th_v$ **then**
      if $pos \in targetz$ **then**
        return $targetz.Center$
      end if
      return pos
    end if
  end if
  return $startz.Center$

Die berechneten Zeiten dienen als Eingabeparameter für Algorithmus 10, der jedes neue Positionsupdate dahingehend prüft, ob es herausgegeben werden darf. Die Idee dieses Verfahrens ist es, den genauen Standort des Nutzers so lange zu unterdrücken, bis man von allen potentiellen Punkten der Startzone die aktuelle Position mit hoher Wahrscheinlichkeit erreicht haben kann.

Im Rahmen der vorliegenden Arbeit wird diese Wahrscheinlichkeit als die Abweichung der Durchschnittsgeschwindigkeiten interpretiert, die der vorderste und der hinterste Ort bis zum aktuellen Standpunkt seit Verlassen der Startzone gefahren sein müssten. Um sich der Verkehrslage der Umgebung anzupassen, sollen diese zudem nicht stark von der tatsächlich gefahrenen Geschwindigkeit abweichen. Es wird daher gefordert, dass diese Geschwindigkeitsdifferenzen innerhalb eines bestimmten Bereichs th_v liegen.

Algorithmus 10 wird bei jeder neuen gemessenen Position nach der Formulierung einer Routenanfrage ausgeführt. Solange die Startzone nicht verlassen wurde, wird weiterhin deren Mittelpunkt als Standort verwendet.

Während sich der Nutzer zwischen den beiden Zonen befindet, wird für jedes Positionsupdate die seit Verlassen der Zone zurückgelegte Strecke und die dazugehörige Fahrzeit ermittelt. Durch die in Algorithmus 9 ermittelten minimalen und maximalen Fahrzeiten von anderen Orten der Zone aus existiert eine Abschätzung, wie viel weiter vorne oder hinten der Benutzer von einem anderen Startpunkt aus momentan wäre. Auf Basis dieser Daten lassen sich die jeweiligen Durchschnittsgeschwindigkeiten berechnen, welche die fiktiven Nutzer zum aktuellen Standort des Nutzers benötigt hätten. Sobald diese innerhalb eines Korridors der Breite th_v liegen, wird die aktuelle Position *pos* an den LBS übermittelt. Befindet sich der Nutzer bereits in der Zielzone, wird diese gesendet. Sonst gibt der Algorithmus die Startzone aus.

Problematisch gestaltet sich dieses Verfahren u.U., wenn der Startpunkt des Nutzers nah am Ausgang der Zone war und dieser mit einer hohen Durchschnittsgeschwindigkeit fährt. In diesem Fall würde das gesamte erlaubte th_v auf seine tatsächliche Geschwindigkeit addiert werden. Für $th_v = 10km/h$ und einer gefahrenen Durchschnittsgeschwindigkeit $v_{self} = 50km/h$ würden sich für den hintersten Ort somit $v_{max} = 60km/h$ ergeben, was über der erlaubten Geschwindigkeit liegt und einem Angreifer auffallen könnte.

Während auch dies keine eindeutigen Rückschlüsse durch den Angreifer zulässt, wird hierfür folgende Lösungsidee skizziert. Ähnlich anderen Assistenzsystemen können privatsphärebezogene Geschwindigkeitsempfehlungen an den Fahrer ausgesprochen werden. Fahrzeuge empfehlen schon heute ein optimales Geschwindigkeitsband, um die nächste Ampel bei grün zu erreichen [197]. Auf Basis der in Algorithmus 9 berechneten relativen Position des Nutzers in der Zone kann dieselbe Idee verwendet werden, um den Nutzer zu empfehlen, seine Geschwindigkeit – je nachdem, wie weit „vorne" er sich in der Zone befindet – geeignet zu drosseln, um derartige Auffälligkeiten zu vermeiden.

Der entgegengesetzte Fall, dass der Nutzer weit hinten und besonders langsam fährt, ist weniger problematisch, da es für langsameres Vorankommen auch für die anderen potentiellen Startpunkte viele Erklärungen geben kann: So fährt ein Nutzer nach Ermittlung der initialen Route u.U. nicht unmittelbar los oder wird durch ein spontanes Mikro-Verkehrshindernis aufgehalten, z.B. ein Fußgängerüberweg oder ein in zweiter Reihe parkendes Fahrzeug.

Startet der Benutzer die live Navigation erst, wenn er sich bereits aus seiner

Startzone entfernt hat, kann unmittelbar mit dem Senden der eigenen Position begonnen werden, da diese nicht mit einem Aufenthaltsort korreliert. In diesem Fall muss lediglich das Ziel der Routenanfrage anonymisiert werden.

4.6.5.5 Nachträgliche Freigabe von Trajektorien

Entscheidet sich ein Nutzer nachträglich für die Freigabe seiner Trajektorie, sollen auch hierdurch keine exakten Hinweise auf seine Aufenthaltsorte verraten werden. Im Folgenden wird davon ausgegangen, dass eine Trajektorie aus einer beliebig langen Liste an Tupeln aus Standortangaben und den dazugehörigen Zeitstempeln besteht. Um die Qualität der Daten nicht zu verschlechtern, stellt das selektive Ausblenden von verräterischen Teilstücken der Trajektorie dabei das einzig erlaubte Mittel für den Schutz der Privatsphäre dar.

Aus der gesamten zu veröffentlichenden Trajektorie *realP* sollen daher all jene Bereiche herausgeschnitten werden, die Hinweise auf die Aufenthaltsorte eines Nutzers geben können, die über die durch k definierte Größe der Verschleierungsmenge hinausgehen. Dies lässt sich auf Basis der Verschleierungszonen mit folgenden Schritten erreichen:

1. Entferne alle Punkte aus der Trajektorie, die in einer Zone liegen, in der bei diesem Betreten der Zone ein Aufenthalt stattgefunden hat.

2. Entferne so lange alle Punkte, die zeitlich vor oder nach dem Besuch der Zone liegen, bis die euklidische Distanz eines Punktes der Trajektorie zu allen Ein- bzw. Ausgängen der Zone größer als $th_{shortcut}$ ist.

3. Entferne alle Punkte aus der Trajektorie, die zeitlich nach einem Aufenthalt liegen, der – laut Cover – kürzer als th_{visit} gedauert hat.

4. Entferne alle Punkte aus der Trajektorie, die während der Nutzung der online Navigation unterdrückt wurden.

5. Alle übrigen Punkte bilden die privatsphärekonforme Trajektorie, die keine Hinweise auf Aufenthaltsorte enthält und veröffentlicht werden kann.

Die ersten beiden Anweisungen beziehen sich auf den grundsätzlichen Schutz vor ortsbasierten Inferenzangriffen durch die Verwendung der Verschleierungszonen sowie auf das Ausblenden potentiell verräterischer Umwege, wie es für bekannte Ziele auch schon in Abschnitt 4.6.5.2 eingesetzt wurde. Der dritte Punkt bedient sich des Prinzips der *plausible Deniability*, um eventuelle Covers nach verräterisch kurzen Aufenthalten in einer Zone effektiv zu verbergen. th_{visit} ist dabei ein Parameter, der pro Zone größer als die maximale Fahrzeit $t_{max}(\mathcal{SVZ}_i)$ gewählt werden sollte, die benötigt wird, um darin alle Ziele sowie von dort wieder den jeweiligen Ausgang erreichen zu können.

Hat in einer Zone ein verräterisch kurzer Aufenthalt stattgefunden, der realistisch nicht an allen Orten der Zone stattgefunden haben kann, lässt sich dies

durch Analyse der in die Zone und aus der Zone führenden Trajektorie schliessen. Durch die grundsätzliche Unterdrückung von Updates nach derart kurzen Aufenthalten ist für einen Angreifer nicht zu unterscheiden, ob nach dem Verlassen der Zone ein Cover durchgeführt wurde oder nicht. Somit lassen sich keinerlei Rückschlüsse auf die grobe Lage des besuchten Ortes innerhalb der Zone ziehen.

4.6.5.6 Zusammenfassung

In diesem Abschnitt wurden Strategien für die privatsphärekonforme Veröffentlichung von Standortdaten in verschiedenen Ausprägungen ortsbezogener Dienste vorgeschlagen. Dabei wird entsprechend den Anforderungen der jeweiligen LBS-Typen, was die Präzision der übermittelten Ortsinformationen angeht, jeweils eine passende Herangehensweise gewählt. Zudem wird darauf geachtet, dass die veröffentlichten Daten dienstübergtreifend konsistent zueinander sind und sich somit auch bei der Kombination der Daten, die ein Nutzer an unterschiedliche LBS übermittelt hat, keine zusätzlichen Hinweise auf dessen Aufenthaltsorte gewinnen lassen.

Die Eigenschaften der erzeugten Zonen selbst sowie die Auswirkungen der soeben vorgestellten Verschleierungsstrategien auf die zu erwartende Dienstqualität aus Nutzersicht werden im folgenden Kapitel untersucht.

4.7 Evaluation der kontinuierlichen Standortverschleierung

In den folgenden Abschnitten werden die soeben vorgestellten SVZs sowie die verschiedenen Strategien zur dienstübergreifend konsistenten Verschleierung von Aufenthaltsorten hinsichtlich ihrer QoS-Eigenschaften untersucht und mit bestehenden Verfahren verglichen.

Anhand der experimentell ermittelten Ergebnisse wird schließlich argumentiert, dass das vorgeschlagene Verfahren praktisch einsetzbar ist und effektiven Schutz vor ortsbasierten Inferenzangriffen bietet.

4.7.1 Datengrundlage

Die Datengrundlage für die nachfolgend durchgeführten Experimente stellt wie zuvor das OSM-Kartenmaterial für Oberbayern vom 15.2.16 dar. Für die Ermittlung kürzester Routen und der Abschätzung von Fahrzeiten wird erneut OSRM verwendet, einzelne Kartenbereiche werden mit Overpass abgefragt.

Die folgenden Experimente beziehen sich wie schon bei der Evaluation von PrOSPR auf die Städte München, Rosenheim und Erding, um die Eigenschaften von LAMA LocO in Regionen mit unterschiedlich hoher Bevölkerungsdichte und verschieden komplexen Straßennetzen untersuchen zu können. Tabelle 4.7 listet die relevanten Eigenschaften der jeweiligen Regionen auf.

Region	PLZ	Adressen	Einw./km^2	Straßenknoten	Einbahnen
München	74	153.154	4.601	130.749	7.983
Rosenheim	3	10.861	1.636	7.312	183
Erding	1	7.113	655	10.137	214

Tabelle 4.7: Statistik über das verwendete Kartenmaterial

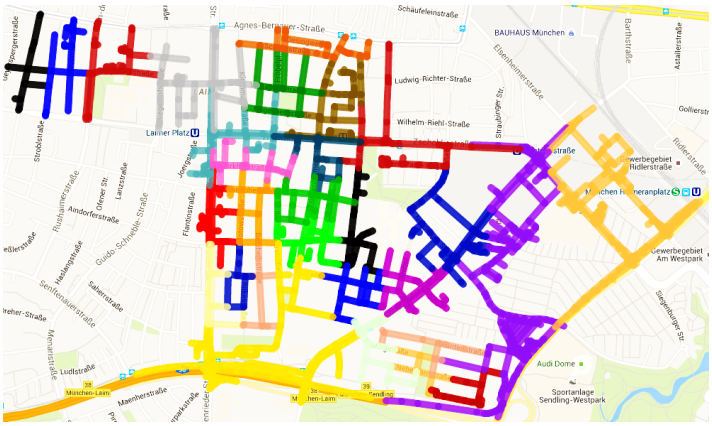


Abbildung 4.20: Partitionierung des PLZ-Gebiets 80686 in starke Verschleierungszonen für $k = 50$.

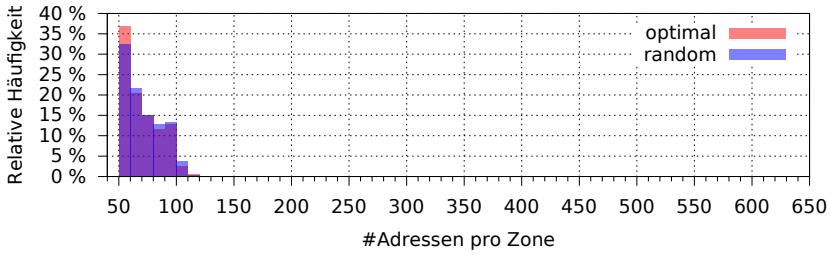
In einem Preprocessing-Schritt werden zunächst Adressduplikate aus den OSM-Daten entfernt. Solche liegen vor, wenn z.B. sowohl ein Gebäude als auch ein eigener Knoten, der den Zugangspunkt zu diesem Gebäude vom Straßennetz beschreibt, mit der entsprechenden Adresse versehen sind. Für die Experimente wird somit darauf geachtet, jede Adresse nur ein Mal zu verwenden. Den größten zusammenhängenden Bereich stellt die Stadt München mit über 150.000 verschiedenen Adressen in 74 Postleitzahlgebieten dar.

In den folgenden Abschnitten werden zunächst einige grundlegende Statistiken über die mit dem vorgestellten Algorithmus gefundene Partitionierung in starke Verschleierungszonen erhoben. Zudem wird die Relevanz der Einbahnstraßenvervollständigung gezeigt und es werden QoS-bezogene Betrachtungen der einzelnen Verschleierungsstrategien durchgeführt. Abschließend für dieses Kapitel erfolgt eine qualitative Analyse des vorgestellten Systems.

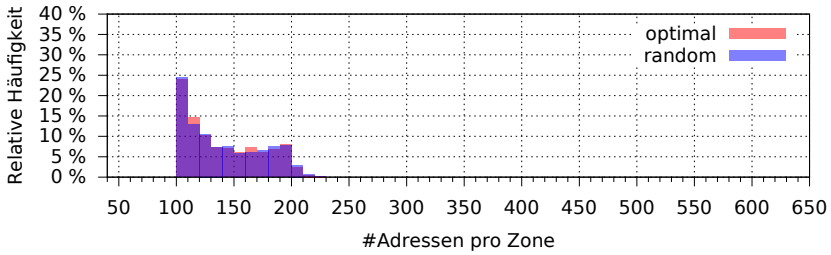
4.7.2 Auswertung der erzeugten Verschleierungszonen

In diesem Abschnitt wird die Größe der erzeugten starken Verschleierungszonen in Abhängigkeit der verwendeten Split-Strategie, der Bevölkerungsdichte sowie der gewählten Parameterbelegung für $k \in \{50, 100, 150, 200\}$ untersucht. Als Metrik wird hierfür zunächst wie in [253] die durchschnittliche Anzahl der in einer SVZ enthaltenen Adressen verwendet.

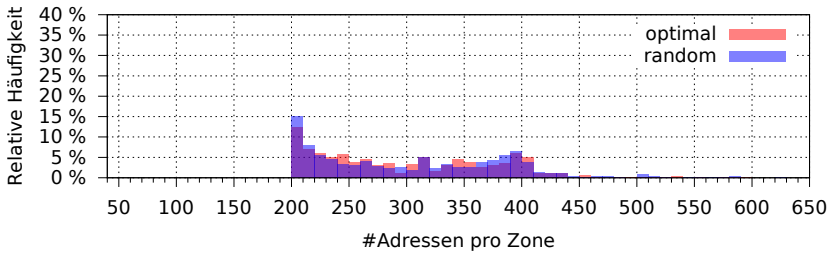
Ein Beispiel für die topologiebezogene Partitionierung eines Postleitzahlgebiets in starke Verschleierungszonen ist in Abb. 4.20 zu sehen. Aus Sicht der



(a) $k = 50$



(b) $k = 100$



(c) $k = 200$

Abbildung 4.21: Relative Häufigkeit der Zonengrößen für die Region München und verschiedene Werte von k.

Privatsphäre darf die Anzahl an semantischen Orten in einer Zone den Wert von k nie unterbieten. Dass diese Bedingung eingehalten wird, gewährleistet der vorgeschlagene Algorithmus zur reziproken Zonenerzeugung. Aus Sicht der erreichbaren Dienstqualität ist zudem jedoch die Erzeugung möglichst vieler SVZs pro Region erstrebenswert, die den Wert von k von oben her annähern.

In Abb. 4.21 sind die Histogramme der Zonengrößen für verschiedene Werte von k für die Region München abgebildet. Den in Kapitel 4.6.4.1 formulierten Anforderungen an starke Verschleierungszonen entsprechend, wird der geforderte Wert von k dabei in keinem Fall unterschritten.

Für alle abgebildeten Parameterbelegungen lässt sich eine Häufung an Zonen ausmachen, die k exakt treffen oder nur knapp darüber liegen. Ein weiterer leichter Anstieg lässt sich jeweils kurz vor $2k$ erkennen, danach ein abruptes Abfallen. Dieses Verhalten erklärt sich dadurch, dass für Zusammenhangskomponenten, die mehr als $2k$ Adressen beinhalten, ein nachträglicher Split-Vorgang ausgeführt wird. Ist dieser erfolgreich, entstehen aus einer großen Zone

zwei kleinere, deren Größen erneut nah an k liegen. Zonen, deren Gebäudezahl knapp unter $2k$ liegt, können zur Einhaltung der Privatsphäre-Anforderungen hingegen nicht geteilt werden und sorgen somit für diese zweite Häufung.

Mit zunehmender Größe von k verbreitern sich die jeweiligen Histrogramme, wobei die überwiegende Mehrheit aller erzeugten Zonen jeweils auf den Bereich zwischen k und $2k$ entfällt. Für größere Werte von k steigt jedoch die Wahrscheinlichkeit, Zonen mit mehr als $2k$ Adressen zu erzeugen. So besitzen jeweils nur $3-5\%$ der erzeugten Verschleierungszonen für $k \in \{50, 100, 150\}$ mehr als $2k$ Gebäude, während es für $k = 200$ unabhängig von der verwendeten Split-Strategie mehr als 11% sind. Für $k = 200$ treten vereinzelt sogar Fälle auf, in denen etwa $3k$ Adressen in einer Zone enthalten sind und für die dennoch keine gültige Partitionierung gefunden werden konnte.

Zwischen den beiden vorgeschlagenen Strategien zur Ermittlung der Schnittkanten für die geometrische Partitionierung *random* und *optimal* lassen sich in der Region München kaum Unterschiede ausmachen. Die jeweils erreichte Anzahl an Partitionen ist in Tab. 4.8 abgedruckt. So beträgt die durchschnittliche Zonengröße je nach verwendeter Split-Strategie 69 bzw. 71 Gebäude für $k = 50$ und 300 bzw. 305 Gebäude für $k = 200$. Die folgenden Untersuchungen beziehen sich daher, wenn nicht explizit anderweitig gekennzeichnet, stets auf die Strategie *optimal*. Die Gebäudezahlen in den erzeugten Partitionen übersteigen den geforderten Wert unabhängig von der verwendeten Strategie im Durchschnitt um ca. 40% für $k \in \{50, 100, 150\}$ sowie um 50% für $k = 200$. Je kleiner k, desto eher lassen sich gültige Partitionierungen finden.

Strategie	**k=50**	**k=100**	**k=150**	**k=200**
optimal	2.200	1.095	728	509
random	2.156	1.089	714	502

Tabelle 4.8: Anzahl der gefundenen Partitionen für die Region München

Als nächstes wird untersucht, ob die Bevölkerungsdichte einer Region Einfluss auf die durchschnittlich erreichbare Anzahl an Adressen pro Partition hat. Abb. 4.22 zeigt die durchschnittliche Gebäudezahl pro Verschleierungszone für die Regionen München, Rosenheim und Erding. Alle drei Regionen verhalten sich nahezu identisch, was diesen Vergleich angeht. Eine Erklärung hierfür ist, dass sich die Strukturen des Straßennetzes in den für die Partitionierung relevanten Eigenschaften trotz deutlicher Unterschiede in der Bevölkerungsdichte nur wenig voneinander unterscheiden bzw. sich im Durchschnitt nivellieren.

Menschliche Mobilität findet heutzutage i.d.R. nicht auf freien Flächen statt, sondern entlang der durch das Straßennetz beschränkten Wege. Angesichts dieser Tatsache wird argumentiert, dass es geringen Mehrwert hat, die von einer Zone abgedeckte Fläche als Vergleichseigenschaft zu verwenden. Eine sinnvollere Möglichkeit, die Größe einer Verschleierungszone zu beschreiben, ist es daher, die Länge des von deren Kerntopologie abgedeckten Straßennetzes aufzusummieren. Abb. 4.23 zeigt die entsprechenden Verteilungen für die Region

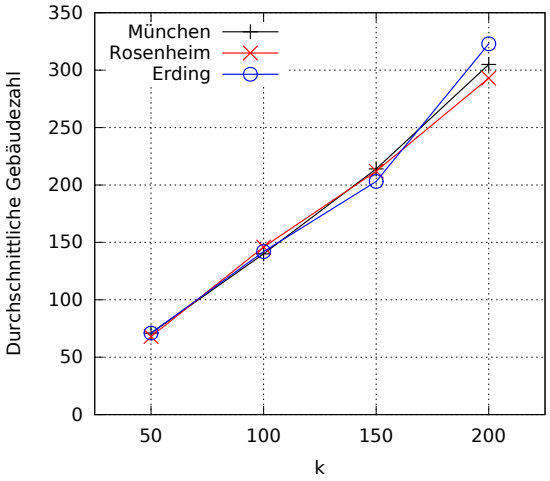
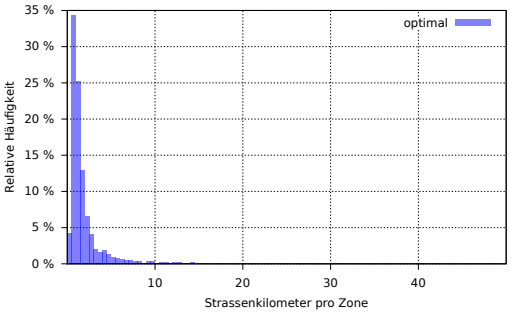
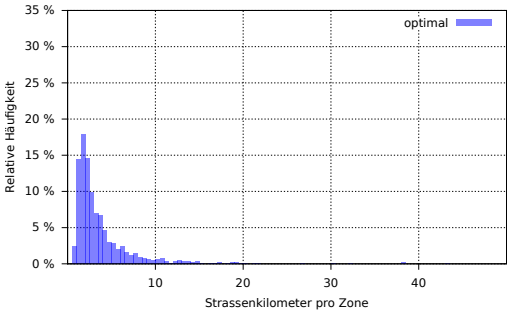


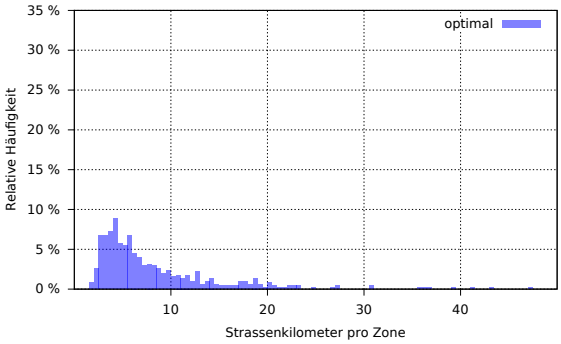
Abbildung 4.22: Durchschnittliche Anzahl an Gebäuden pro Zone nach Region in Abhängigkeit von k.



(a) $k = 50$



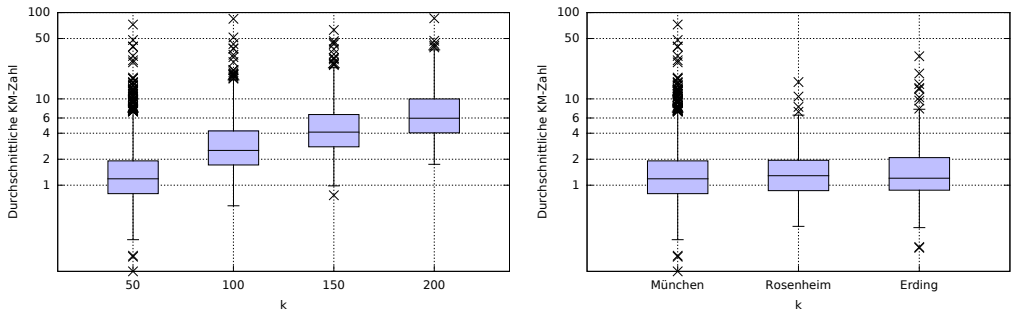
(b) $k = 100$



(c) $k = 200$

Abbildung 4.23: Relative Häufigkeit der Länge an Straßenkilometern pro Zone für die Region München, verschiedene Werte von k.

München für verschiedene Werte von k. Es werden dabei die Längen aller einer Zone zugehörigen Straßensegmente aufsummiert, die gemäß des OSM-Datenmodells gespeichert sind. Beidseitig befahrbare Straßen sowie parallele



(a) Anzahl an Straßenkilometern pro Zone in der Region München, verschiedene k.

(b) Pro Zone enthaltene Anzahl an Straßenkilometern, regionsübergreifend für $k = 50$.

Abbildung 4.24: Straßenkilometer pro Zone für unterschiedliche Werte von k und verschiedene Regionen.

Spuren derselben Fahrtrichtung werden daher nur einfach gezählt. Grundlage bildet bei dieser Statistik wie auch bei der Partitionierung selbst das gesamte befahrbare Straßennetz einer Region.

Es ergibt sich dabei das intuitiv erwartbare Bild. Je kleiner der Wert von k, desto kleiner fallen die erzeugten Verschleierungszonen auch hinsichtlich der in ihnen enthaltenen Straßenkilometer aus. Für $k = 50$ weisen 63% weniger als 1,5 km an Straßen auf. Für $k \in \{50, 100\}$ fallen jeweils deutlich mehr als 90% der Zonen in den Bereich unter 10 km. Für $k = 200$ sind dies noch 75% und weitere 19%, die weniger als 20km an Straßensegmenten beinhalten.

Zur Erhöhung der Lesbarkeit ist die x-Achse in allen drei Abbildungen jeweils bei 50 km abgeschnitten. Es sei daher darauf hingewiesen, dass es selbst für $k = 50$ deutliche Ausreißer nach oben gibt, sodass sich hierbei ein Mittelwert von 1,923 km pro Zone ergibt. Die vollständigen Verteilungen samt aller Ausreißer sind der Kastengrafik in Abb. 4.24a zu entnehmen. Man beachte, dass die y-Achse dort logarithmisch angegeben ist. Wie man sieht, wird schon für $k = 50$ eine Zone erzeugt, die ganze 72km an Straßensegmenten beinhaltet.

Die Kerntopologie des größten Ausreißers für $k = 50$ ist in Abb. 4.25 gezeigt. Für alle weiteren Werte von k befindet sich dieser an derselben Stelle und variiert in Abhängigkeit von k lediglich in seiner genauen Größe. Die Grenze des PLZ-Gebiets ist in schwarz eingezeichnet, die Topologie der Zone in blau.

Zunächst liegt die Vermutung nahe, der vorgeschlagene Algorithmus hätte keine gültigen Partitionierungen dieses großen Gebiets gefunden. Entlang des gesamten Straßennetzes dieser Zone finden sich jedoch gerade einmal 55 unterschiedliche Adressen – die Partitionierung hat ihr Ziel also sehr genau erreicht. Bei der Untersuchung aller Ausreißer für den Bereich München und $k = 50$ befinden sich insgesamt nur zwei Mal mehr als $2k$ Adressen in der entsprechenden Zone. Bei näherer Analyse der Ausreißer ergeben sich zwei Gründe für die große Zahl an Straßenkilometern. Insbesondere in der in Abb. 4.25 gezeigten Situation ist das Straßennetz an einigen Stellen extrem feinmaschig,

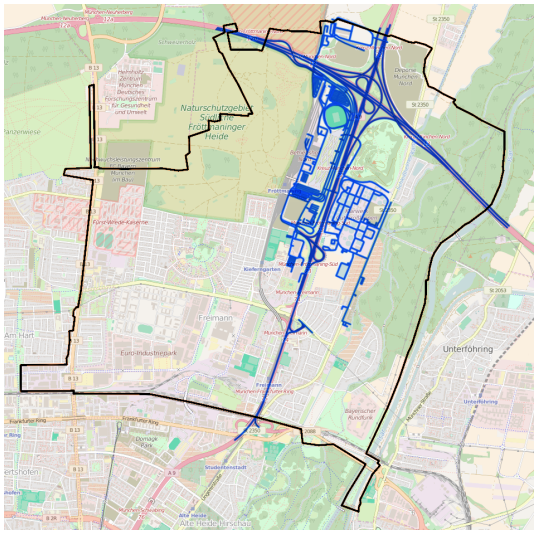
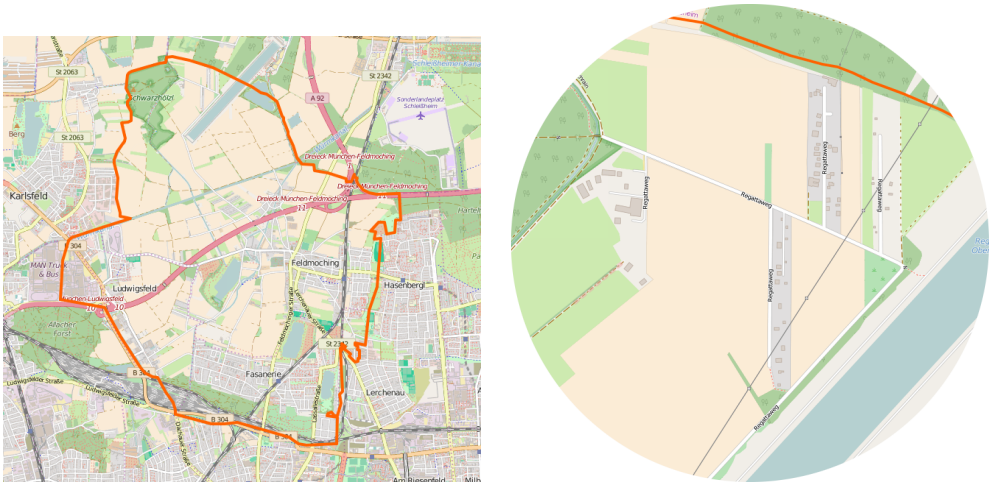


Abbildung 4.25: Kerntopologie der größten einzelnen Verschleierungszone in der Region München (PLZ 80939) für $k = 50$.



(a) Kartenausschnitt für das PLZ-Gebiet 80995.

(b) Vergrößerter Ausschnitt aus dem Norden des PLZ-Gebiets.

Abbildung 4.26: Straßennetz und Besiedlung eines Postleitzahlengebiets im Außenbereich.

da in Nähe dieses Autobahnendes mehrere Großparkplätze enthalten sind, die konsequenterweise – z.B. um von jedem Ort ein Routing zu ermöglichen – zum Straßennetz hinzugezählt werden müssen. Zieht man diese ab, verbleiben jedoch noch immer ca. 50km an Straßensegmenten in dieser Zone.

Neben derart hohen Anteilen an asphaltierten Flächen ist die zweite, auch privatsphäretechnisch relevante Erklärung für die Entstehung großer SVZs genau im Gegenteil zu sehen. Fast alle derartigen Ausreißer befinden sich im äußeren Gürtel der Münchner PLZ-Gebiete und beinhalten große Freiflächen

zwischen den besiedelten Bereichen. Die einzigen Ausnahmen hiervon bilden 81677 (S-Bahn-Stellwerk), 80809 (Olympiapark) und zwei PLZs am Englischen Garten, für die dieselben Beobachtungen gelten. Dadurch existieren lange Teilstrecken auf Landstraßen und Autobahnen ohne Kreuzungen, die meist vollständig einer Zone zugeordnet werden. Abb. 4.26a zeigt dies exemplarisch für die Postleitzahl 80995 im Münchner Norden.

Aus Sicht der Privatsphäre ist hierbei die folgende Beobachtung wichtig: Befinden sich in einer dieser Teilsiedlungen weniger als k Adressen, muss ein Verschmelzen zweier geometrisch ermittelter Teilpartitionen über relativ lange Strecken zu anderen bebauten Flächen der Zone stattfinden. Im Beispiel trifft das sowohl auf die Siedlung im Westen als auch auf das abseits liegende Wohngebiet an der Nordgrenze des PLZ-Gebietes (vgl. Abb. 4.26b) zu. V.a. letzteres benötigt schon für einen relativ kleinen Wert für k unbedingt eine Verbindung in andere, stärker bebaute Bereiche.

Abb. 4.24b zeigt dementsprechend auch im Vergleich der Größen der Kerntopologien der erzeugten SVZs für unterschiedliche Bevölkerungsdichten ein einheitliches Bild. Die durchschnittlichen Zonengrößen ähneln sich stark, wobei Rosenheim und Erding keine derart großen Ausreißer erzeugen wie München. Das kleinere Erding erzeugt dabei größere Ausreißer als Rosenheim, was der Existenz vieler kleiner Ansiedlungen im PLZ-Bereich zuzuschreiben ist.

München	**k=50**	**k=100**	**k=150**	**k=200**
Partitionen	2200	1.095	728	509
davon mit $G^+ \neq \emptyset$	36,2 %	46,7 %	52,6 %	56,6 %
∅ Weglänge	366,4 m	294,5 m	254,3 m	232, 2m
max. Weglänge	14,3 km	14,3 km	14,3 km	14,3 km

Tabelle 4.9: Erweiterte Straßennetze der SVZs in München.

Tabelle 4.9 zeigt die Statistik über die aufgrund der Vervollständigung von Einbahnstraßen entstehenden, erweiterten Straßentopologien G^+ der erzeugten Zonen in der Region München. Diese zusätzlichen Kanten sorgen bei LAMA LocO dafür, dass schwache Zusammenhangskomponenten einer Zone gegenseitig erreichbar sind ohne die Zone zu verlassen (vgl. Kapitel 4.6.4). Mit größerem k steigt der Anteil an Zonen, die zur Vervollständigung ihrer Einbahnstraßen eine Erweiterung ihrer Kerntopologie nötig machen. Gleichzeitig nimmt die durchschnittliche Länge der einzelnen Ergänzungsrouten ab.

Für die hohen Maximalwerte einzelner Routen gibt es zwei Gründe: In seltenen Fällen schlägt das OSRM-Mapping von Adressen auf das nächstgelegene Straßensegment dahingehend fehl, dass diese fälschlicherweise auf eine Autobahn gelegt werden. Hier kann nur die Verwendung einer hochwertigeren, manuell gepflegten Geocoding-Datenbank Abhilfe schaffen. Zum anderen sind im OSM-Kartenmaterial tatsächlich mit Adressen versehene Gebäude entlang von Autobahnsegmenten enthalten, die zu einem PLZ-Gebiet gehören. Beide Situationen verursachen zwangsläufig eine lange Ergänzungsroute über die

nächste Autobahnausfahrt zur davorliegenden Auffahrt.

Rosenheim	**k=50**	**k=100**	**k=150**	**k=200**
Partitionen	167	76	52	39
davon mit $G^+ \neq \emptyset$	18,6 %	19,7 %	23,1 %	20,5 %
∅ Weglänge	184,8 m	127,5 m	122,6 m	98,9 m
max. Weglänge	934,0 m	671,7 m	934,0 m	934,0 m

Tabelle 4.10: Erweiterte Straßennetze der SVZs in Rosenheim.

In Tabelle 4.10 ist dieselbe Statistik für Rosenheim abgedruckt. Die kleinere Stadt weist hier durchwegs kürzere durchschnittliche Weglängen sowie Maximalwerte von unter einem Kilometer auf. Die kürzesten beobachteten Ergänzungsrouten liegen unabhängig von k und Region bei unter 10m. Konsistent zu der deutlich niedrigeren Zahl an Einbahnstraßen verfügen hier auch relativ gesehen weniger Zonen überhaupt über einen erweiterten Straßengraphen. Dass eine Zone keine solchen besitzt, bedeutet jedoch nicht, dass sie keine Einbahnstraßen o.ä. beinhaltet. In solchen Fällen sind aber bereits innerhalb der Kerntopologie alle Orte bzw. Ein- und Ausgänge gegenseitig erreichbar, sodass G^+ keine Kanten enthält.

Am Auftreten der Ausreißer sowie an den präsentierten Beispielen zeigt sich, wie wichtig die Forderung nach der Reziprozität der Verschleierungszonen ist. Verfahren, die zwar auf die Einhaltung des Parameters k achten, ansonsten aber aus QoS-Gedanken möglichst kleine Zonen erzeugen, verraten somit bei dem Auftreten solch großer Zonen, welche Adressen diese Zone verursacht haben müssen. Durch die reziproke Zonenerstellung, die LAMA LocO durch die vollständige Partitionierung eines Gebietes unabhängig vom Nutzerstandort im Voraus durchführt, lassen sich solche Rückschlüsse verhindern.

Die Situation in Abb. 4.25 zeigt zudem, dass Gebäudezahl und Größe der reziproken Zonen nicht automatisch korrelieren. Darüber hinaus sagt die Größe einer Zone nicht unmittelbar etwas über die erreichbare Dienstqualität aus, da ein Nutzer auch im Worst Case nicht das gesamte Straßennetz der Zone abfahren muss, um von seinem aktuellen Standort den Mittelpunkt der Zone zu erreichen. Im nächsten Abschnitt wird daher mit Hilfe simulierter LBS-Anfragen die durchschnittlich zu erwartende Dienstqualität gem. der verschiedenen Strategien zur Standortverschleierung untersucht.

4.7.3 Einfluss der Standortverschleierung auf die Dienstqualität

Im Folgenden wird untersucht, wie sich der Einsatz der kontinuierlichen Standortverschleierung mit LAMA LocO aus Sicht des Nutzers auf die erreichbare Dienstqualität auswirkt. Als Metrik für den zu erwartenden Verlust an Servicequalität wird $Q_{loss}(\psi, f, d_q)$ von Shokri et al. [219] verwendet. $\psi(p)$ stellt

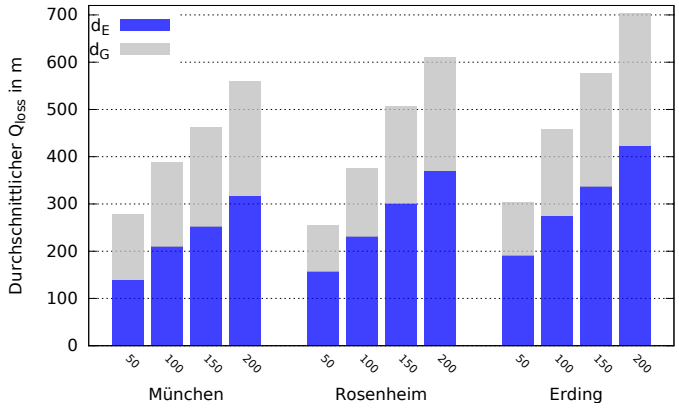


Abbildung 4.27: Durchschnittlicher Qualitätsverlust bei der Verwendung der Standortverschleierung mit unterschiedlichen Werten für k.

dabei die Aufenthaltswahrscheinlichkeit eines Nutzers an einem bestimmten Ort p dar, f ist die verwendete Verschleierungsmethodik, die den tatsächlichen Standort p auf einen verschleierten Punkt p' abbildet und d_q ist ein Maß für die Distanz zwischen p und p'.

Q_{loss} ist definiert als die durchschnittliche Abweichung der tatsächlichen Standorte von den verschleierten Positionsangaben gem. des gewählten Distanzmaßes – je mehr sich diese Punkte unterscheiden, desto größer ist der zu erwartende Qualitätsverlust. Im Rahmen der vorliegenden Arbeit werden hierfür die euklidische Distanz d und die Distanz im Straßennetz d_G verwendet. Letztere wird anhand der Länge des kürzesten, verkehrsregelkonformen Pfades von der tatsächlichen Position p zu p' gemessen.

Für alle der nachfolgenden Experimente werden für jede Parameterbelegung jeweils 10.000 (München) bzw. 5.000 (Rosenheim, Erding) Standorte auf Basis der soeben analysierten SVZs verschleiert. p wird dabei zufällig innerhalb der jeweiligen Region gewählt, mit $\psi(p) = 1$ und $\psi(q \neq p) = 0$. Die Datengrundlage bilden die SVZs aus dem vorigen Kapitel, die mittels der *optimal* Strategie erzeugt wurden.

Im ersten Schritt wird der Qualitätsverlust im Rahmen der sporadischen LBS-Nutzung evaluiert. Die Standortverschleierung findet dabei gem. Algorithmus 5 statt, d.h., dass die aktuelle Position unabhängig von den vorherigen Bewegungen des Nutzers auf den geografischen Mittelpunkt der aktuellen Aufenthaltszone verschoben wird. Das erweiterte Straßennetz dieser Zone spielt hierbei keine Rolle, da sich der Nutzer nicht in Bewegung befindet, sondern sich statisch an einem Ort innerhalb einer Zone aufhält.

Abb. 4.27 zeigt den Qualitätsverlust für verschiedenen Werte von k in den unterschiedlichen Regionen. Logischerweise ist Q_{loss} bei der Verwendung von d_G als Distanzmaß stets höher als bei der Berechnung über den euklidischen Abstand. Da d_G den interessanteren Wert darstellt, wird der Qualitätsverlust im Folgenden unter Berücksichtigung dieses Distanzmaßes untersucht.

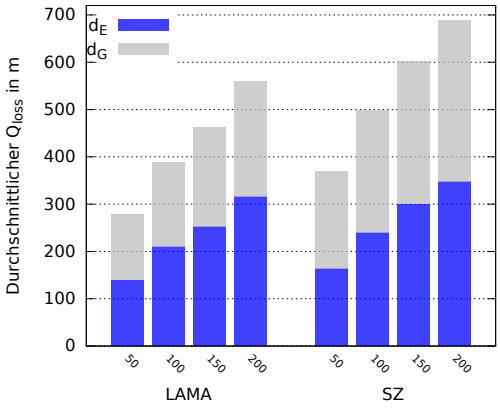


Abbildung 4.28: Qualitätsverlust im Vergleich zur Verwendung von Silent Zones ($\alpha = 1.0$, $\rho = 12.0$) in der Region München.

Mit steigendem Wert von k nimmt auch der durchschnittliche Qualitätsverlust zu, jedoch mit einer langsameren Wachstumsrate als k. So muss der Nutzer für $k = 50$ in der Region München davon ausgehen, dass der an den LBS übermittelte Standort im Durchschnitt 277,6 m Wegstrecke von seinem tatsächlichen Aufenthaltsort entfernt liegt. Für den vierfachen Wert von $k = 200$ verdoppelt sich Q_{loss} im Durchschnitt auf 551,5 m. Aus diesem Ergebnis lässt sich somit die aus QoS-Sicht positive Schlussfolgerung ableiten, dass ein doppelter Grad an Privatsphäre keinen doppelten Qualitätsverlust bedeutet.

Anhand von Abb. 4.27 lassen sich zudem Unterschiede hinsichtlich der erreichbaren Dienstqualität zwischen den einzelnen Regionen erkennen. Während sich die verschiedenen Bevölkerungsdichten wie gezeigt kaum unterscheiden, was die mittlere Anzahl an Gebäuden oder Straßenkilometern pro Zone betrifft, ist der zu erwartende Qualitätsverlust ortsbezogener Dienste in den weniger dicht besiedelten Gebieten Rosenheim und Erding v.a. für große Werte von k deutlich größer als in München. Für $k = 200$ sind in München durchschnittlich 551,5 m Umweg zu erwarten, in der Kleinstadt bereits knapp über 700 m. Eine Erklärung hierfür ist, dass das Straßennetz in der Großstadt enger vermascht ist als in den anderen beiden Gebieten. So ähneln sich die erzeugten SVZs zwar in der Gesamtlänge der enthaltenen Straßensegmente, unterscheiden sich – wie man auch am Vergleich der euklidischen Abstände sieht – jedoch in ihrer räumlichen Ausdehnung und der Länge einzelner Straßenzüge.

In Abb. 4.28 ist der durchschnittliche Qualitätsverlust bei der Verwendung von LAMA LocO im Vergleich zum Silent Zone-Verfahren in der Region München gezeigt. Letzteres ist das einzige bisher bekannte Verfahren, das bei der Erstellung von Verschleierungszonen eine Mindestanzahl an semantischen Orten innerhalb der Zone gewährleistet. Die Silent Zone-Erstellung erfolgt wie zuvor mit Hilfe der in Algorithmus 1 beschriebenen Implementierung des RR-KBI-Verfahrens. Sowohl für die Verschleierung mittels der topologiebasierten SVZs als auch mit einer spontan erzeugten SZ dient das zentral gelegenste

Gebäude der Zone als verschleierte Standortangabe an den LBS.

Wie man sieht, schneidet die Standortverschleierung mit LAMA LocO bei der verwendeten Parameterbelegung unabhängig vom verwendeten Distanzmaß für jeden Wert von k besser ab. Der durchschnittliche Qualitätsverlust für d_G fällt bei der Verwendung der im Rahmen dieser Arbeit vorgestellten SVZs mit 80m bis 130m geringerem Abstand zum Verschleierungspunkt jeweils kleiner aus als bei dem Einsatz der rein gebäudebasierten Silent Zones, obwohl im Schnitt jeweils gleich viele Adressen darin enthaltenen sind.

Die Erklärung hierfür ist, dass die topologiebasierte Zonenerstellung sicherstellt, dass die Kerntopologie einer Verschleierungszone eine schwache Zusammenhangskomponente darstellt. Dies sorgt mit hoher Wahrscheinlichkeit dafür, dass ein kurzer Weg zwischen zwei Punkten einer Zone im Straßennetz existiert. Die Silent Zone-Erstellung findet hingegen rein anhand der Verteilung von Gebäuden statt. Je nach Lage und Beschaffenheit der Umgebung können hierbei lange Umwege entstehen, wenn das von der Zone abgedeckte Straßennetz aus mehreren Zusammenhangskomponenten besteht, was sich in dem entsprechend höheren Wert für Q_{loss} widerspiegelt. Insbesondere wenn der RR-Algorithmus ein langgezogenes Rechteck mit stark ungleichen Seitenlängen aufspannt, kommt es häufig zu der Situation, dass d_G verhältnismäßig hohe Werte annimmt. Je kleiner der in Alg. 1 eingeführte Parameter ρ gewählt wird, umso kleiner fällt hier daher der durchschnittliche Q_{loss} aus: Für $\rho \leq 2.0$ verursachen die Silent Zones erwartungsgemäß sogar weniger Qualitätsverlust als LAMA LocO. Die in Kapitel 4.6.3 genannten Nachteile der Erzeugung nicht-reziproker, flächenmäßig möglichst kleiner Zonen ohne Einbezug des Straßengraphen bleiben dabei jedoch bestehen.

Im Gegensatz zu sporadischen LBS-Anfragen muss bei der periodischen oder kontinuierlichen Standortfreigabe auch die Korrelation aufeinanderfolgender Positionsupdates berücksichtigt werden. Dies gewährleisten die Algorithmen 6 und 7 durch Hinzunahme der erweiterten Straßennetze der SVZs sowie der Erzeugung von Cover-Stories für verräterisch kurze Aufenthalte.

Zunächst werden kurze Aufenthalte ignoriert, um untersuchen zu können, wie sich die kontinuierliche Standortverschleierung im Regelfall im Vergleich zur sporadischen LBS-Nutzung verhält. Hierfür muss nun auch das erweiterte Straßennetz der Verschleierungszonen berücksichtigt werden. Befindet sich der Nutzer aktuell z.B. auf einem Pfad, der zur Vervollständigung einer Einbahnstraße einer zuvor besuchten Zone z dient, wird nicht der Mittelpunkt der aktuellen Aufenthaltszone zurückgegeben, sondern weiterhin der von z.

Auch die kontinuierliche Standortverschleierung wird für 10.000 zufällig gewählte Punkte ausgeführt. Abb. 4.29 zeigt Q_{loss} auf Basis von d_G für die Region München. Wie zu erwarten ist, vergrößert sich der durchschnittliche Qualitätsverlust, wenn sich der Nutzer in Bewegung befindet und kontinuierliche Standortupdates an einen LBS übermittelt. Für $k = 200$ wächst Q_{loss} im Durchschnitt über alle Versuche von ca. 550 m bei der sporadischen

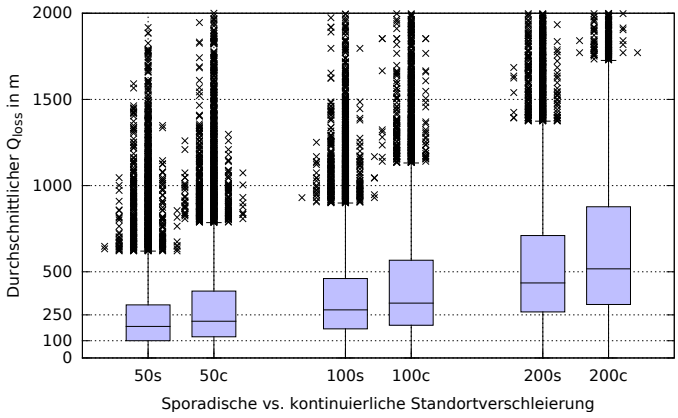


Abbildung 4.29: Qualitätsverlust mit d_G bei der sporadischen (s) und kontinuierlichen (c) Standortverschleierung in der Region München für verschiedene k.

Verschleierung auf knapp über 800 m an. Gleichzeitig erkennt man anhand der Balkendiagramme, dass sich die überwiegende Mehrheit aller Distanzen unterhalb dieser Marke bewegt. Die Ausreißer in Abb. 4.29 sind wie schon in Kapitel 4.7.2 beschrieben auf das fehlerhafte Adress-Mapping von OSRM und die tatsächliche Lage von Adressen an Autobahnen, etc. zurückzuführen.

Als nächstes wird untersucht, wie groß das Delay ist, das für das Verbergen womöglich verräterisch kurzer Aufenthalte an einem Ort zusätzlich auf die Standortverschleierung angewendet werden muss, bzw. was als „zu kurzer" Aufenthalt anzusehen ist. Für den Fall, dass in einer Zone ein Aufenthalt stattgefunden hat und die Zeit, die der Nutzer insgesamt in der Zone verbracht hat so kurz war, dass dieser Aufenthalt nicht an jedem Ort der Zone gewesen sein kann, erzeugt Algorithmus 6 eine entsprechende Cover-Story.

Um abschätzen zu können, um welche Zeitspannen es sich hierbei in Abhängigkeit von k handelt, werden zufällige Paare aus Ein- und Ausgängen einer Zone ausgewählt und die minimale sowie die maximale Fahrzeit vom Eingang zu allen Orten der Zone und von dort zum Ausgang ermittelt. Die Differenz $\Delta t = t_{max} - t_{min}$ zwischen der maximalen und minimalen Fahrzeit stellt das schlechtestenfalls notwendige Delay für das jeweilige Ein- und Ausgangspaar in dieser Zone dar. Für die Abschätzung der benötigten Fahrzeiten zwischen den Adressen einer Zone und dem gewählten Ein- und Ausgang wird eine lokale OSRM-Instanz mit dem Default-Geschwindigkeitsprofil (d.h. 80 % der zugelassenen Maximalgeschwindigkeit einer Kante) verwendet.

Die Ergebnisse dieses Experiments sind in Abb. 4.30 für verschiedene Werte von k und unterschiedliche Regionen dargestellt. Erwartungsgemäß nimmt die Differenz zwischen der minimalen und maximalen Fahrzeit innerhalb einer Zone mit größer werdendem k ebenfalls zu. Für die Region München und $k = 50$ beträgt dieser Wert für den schlechtesten Fall im Durchschnitt über alle Zonen

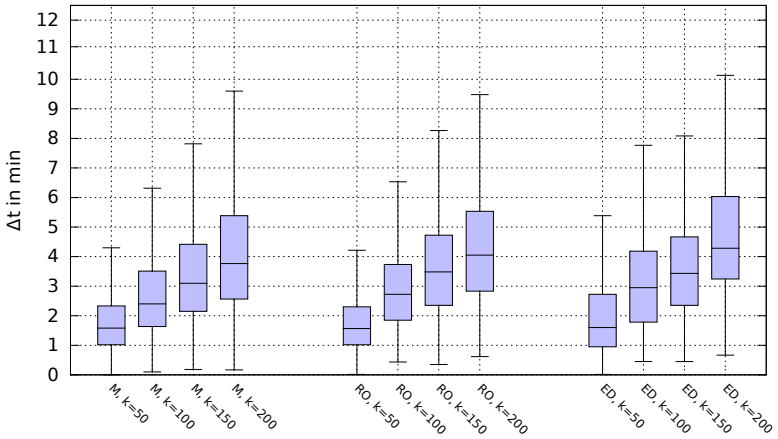


Abbildung 4.30: Notwendiges Delay für das Verbergen kurzer Aufenthalte in einer Zone für verschiedene Werte von k.

117 Sekunden und 262 Sekunden für $k = 200$. In den beiden anderen Gebieten fallen diese Werte ähnlich aus, wobei insbesondere für große Werte von k hier im Durchschnitt leicht größere Werte auftreten.

Deutliche Unterschiede zwischen den einzelnen Regionen zeigen sich lediglich bei den beobachtbaren Maximalwerten. Diese liegen in München bei 1.721 Sekunden (28 Minuten) für $k = 200$, in Rosenheim und Erding hingegen nur 879 bzw. 911 Sekunden. Zur besseren Lesbarkeit wurde in Abb. 4.30 auf eine Darstellung dieser Ausreißer verzichtet. Auslöser dieser hohen Maximalwerte sind erneut die auf Autobahnen gemappten Adressen einer Zone, die unweigerlich lange Rückfahrten verursachen und durch die Verwendung einer qualitativ hochwertigeren Geocoding-Datenbank vermieden werden könnten.

Je nachdem, wo und wie lange sich ein Nutzer in der Zone aufgehalten hat, beträgt die anzuwendende Verzögerung *delay* gem. Algorithmus 6 zwischen 0 und Δt Sekunden. Aus QoS-Sicht bedeutet dies, dass in der Folge die Zone als Standort ausgegeben wird, in der sich der Nutzer vor *delay* Sekunden aufgehalten hat. Der dabei auftretende zusätzliche Qualitätsverlust kann anhand der Durchschnittsgeschwindigkeit des Nutzers seit Verlassen der Zone berechnet werden. Bewegt sich der Nutzer z.B. mit 40 km/h fort, würde der durchschnittliche zusätzliche Worst-Case-Q_{loss} in München für $k = 50$ 1,3 km betragen und 2,9 km für $k = 200$. Der nächste „reguläre" Aufenthalt, der länger als das lokale $\Delta t + delay$ dauert, setzt die Verzögerung automatisch auf 0.

Es sei an dieser Stelle daran erinnert, dass diese Cover-Stories jedoch nur dann eingesetzt werden müssen, wenn der Nutzer einen Aufenthalt in einer Zone hatte, in der er insgesamt kürzer als das lokale Δt der Zone war. Zu kurze Aufenthalte, die an anderen Orten der Zone als das Ziel mit der minimalen Fahrzeit stattfinden, verursachen ein entsprechend kleineres Delay. Für Aufenthalte, die länger als Δt angedauert haben, wird gar kein Delay benötigt – ebenso wie beim bloßen Durchqueren einer Zone.

Bei der Nutzung eines interaktiven LBS während eine solche Cover-Story aktiv ist, könnte der Benutzer gefragt werden, ob die Eingrenzung seines letzten Aufenthaltsortes auf eine Verschleierungsmenge mit weniger als k Orten zugunsten einer besseren Dienstqualität für ihn akzeptabel ist. Bestätigt der Nutzer dies, kann die aktuelle Zone ausgegeben werden. Alternativ kann dem Nutzer angezeigt werden, wie hoch der zu erwartende Qualitätsverlust aktuell ausfällt bzw. wie lange er an seinem aktuellen Ort warten müsste, um das Delay durch einen solchen privatsphärebedingten Zwischenstop auszugleichen.

Abschließend soll untersucht werden, welche Ergebnisse sich bei der privatsphäreschonenden Routenplanung unter dem Einsatz der starken Verschleierungszonen erreichen lassen. In Tabelle 4.11 sind die hierfür verwendeten Parameterbelegungen dargestellt. Da die gegenseitige Konnektivität von Punkten der Zone bei der Verwendung von starken Verschleierungszonen bereits gewährleistet ist, werden nur die VP-Heuristiken aus Kapitel 4.4.1.5 eingesetzt, die nur einen VP für Start- und Zielzone auswählen. Für jeden Wert von k werden erneut 120 zufällig ausgewählte Routen erzeugt und gegen die unverschleierte Originalroute verglichen. Weicht die vom Routenplaner geschätzte Fahrzeit einer verschleierten Route weniger als 5 Sekunden von der Originalfahrzeit ab, wird das Ergebnis als optimal angesehen.

Parameter	**Belegungen**
k	50, 100, 150, 200
VP-Heuristik	`R1`, `C1`
Routenvervollständigung	`SIB`
Routing Szenario	München→Rosenheim

Tabelle 4.11: Verwendete Parameterbelegungen

In Abb. 4.31 sind die Ergebnisse dieses Experiments abgebildet. Wie man sieht, kann die Routenanfrage tatsächlich in jedem Fall beantwortet werden. Die `C1`-Heuristik, die den geographischen Mittelpunkt der Zone als Dummypunkt auswählt, schneidet dabei erwartungsgemäß noch etwas besser ab als die Random-Variante. So erzeugt erstere z.B. nie Umwege, die länger als 5 Minuten ausfallen und findet für die meisten Werte von k auch mehr optimale Antworten. Die Erklärung für dieses Verhalten ist, dass vom Mittelpunkt ausgehende Routen eine höhere Wahrscheinlichkeit haben, den „idealen" Ausgangspunkt aus der Zone zu wählen. Bei der Random-Implementierung hingegen kann der zufällig gewählte VP nah am Rand der Zone liegen, sodass die verschleierte Route mit größerer Wahrscheinlichkeit über einen anderen als den optimalen Ausgang führt, was zwangsläufig einen Umweg verursacht.

Für größere Werte von k werden bei `C1` immer weniger optimale Ergebnisse gefunden: Sind dies bei $k = 50$ immerhin 51 %, sinkt dieser Wert für $k \in \{150, 200\}$ auf 33 % ab. Interessanterweise nimmt die Anzahl an optimalen Routen bei `R1` für $k = 200$ deutlich zu, nachdem dieser Wert zuvor auch

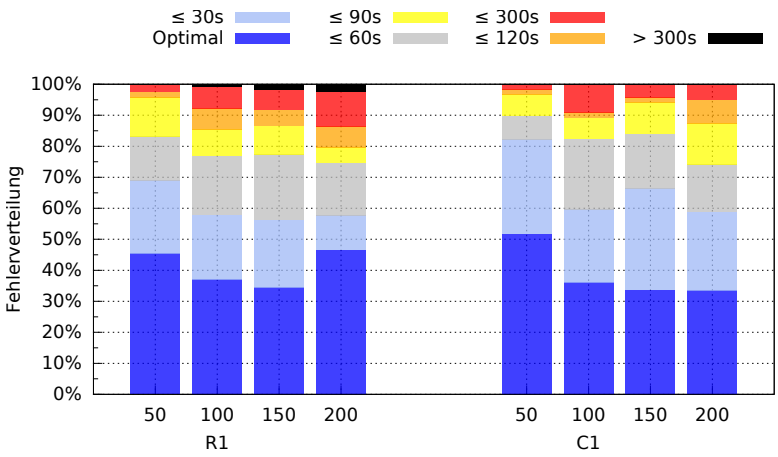


Abbildung 4.31: Fehlerverteilung bei der privatsphäreschonenden online Routenplanung mit LAMA LocO für verschiedene Werte von k.

mit wachsender Zonengröße abfällt. Gleichzeitig steigt jedoch die Wahrscheinlichkeit, Umwege einer Länge von mehr als 2 Minuten zu erzeugen, auf 13 % an, verglichen mit nur 5 % bei `C1`, was fast einer Verdreifachung entspricht.

Im Durchschnitt über alle ausgewerteten Routen entsteht bei der Verwendung von `C1` ein Umweg von 17,4 bzw. 38,3 Sekunden Länge für $k = 50$ und $k = 200$. Für `R1` liegen diese Werte jeweils um etwa 10 Sekunden höher bei 26,3 und 49,8 Sekunden. Auch in diesem Zusammenhang lässt sich somit festhalten, dass eine Vervierfachung des Grades an Standortanonymität im Mittel zu einer Verdopplung der Einbußen hinsichtlich der Dienstqualität führt.

Die Kombination aus den starken Verschleierungszonen von LAMA LocO und der Strategie `C1` ist somit zwar nicht dazu in der Lage, ähnlich gute Werte wie PrOSPR unter Verwendung der `R5`-Heuristik (vgl. Tabelle 4.2) zu erzeugen. Es wird damit jedoch garantiert ein Ergebnis gefunden, für das durchgängig nur eine einzige Routenanfrage an den LBS benötigt wird, während `R5` – ohne derartige Garantien – 25 verursacht, um diese Antwort zu ermitteln.

4.7.4 Zusammenfassung der Ergebnisse von LAMA LocO

In diesem Kapitel wurden die Eigenschaften der erzeugten starken Verschleierungszonen sowie die Auswirkungen der kontinuierlichen Standortverschleierung auf die Servicequalität ortsbezogener Dienste untersucht.

Weder die Anzahl an „überschüssigen" Gebäuden in einer Zone noch die aufsummierte Länge aller Straßensegmente ermöglichen einen unmittelbaren Rückschluss auf die erreichbare Dienstqualität. Auch eine Zone, die nur wenige Adressen beinhaltet, kann große Distanzen zwischen den einzelnen Gebäuden und damit eine deutliche Verschlechterung der Dienstqualität verursachen. Im Gegensatz kann eine Zone mit vielen Gebäuden inmitten eines dichtbesiedelten Stadtkerns liegen und dadurch nur wenig räumlichen Versatz hervorrufen.

Gleichzeitig kann, wenn das Straßennetz komplex und eng vermascht ist, auch ein hoher Wert für die Gesamtlänge der enthaltenen Straßensegmente einen nur kaum merklichen Qualitätsverlust bedeuten.

Im Vergleich zur rein gebäudebasierten Erzeugung von Verschleierungszonen wurde gezeigt, dass sich das topologiebezogene Vorgehen bei LAMA LocO positiv auf den durchschnittlichen Qualitätsverlust auswirkt. Darüber hinaus garantieren die neu vorgestellten SVZs, dass alle Orte einer Zone auch tatsächlich von allen Eingängen aus erreichbar sind, sodass ein potentieller Angreifer keine Teilbereiche effektiv ausschließen kann.

Zudem wurde argumentativ und anhand von Beispielen gezeigt, dass aufgrund der inhomogenen Verteilung von Gebäuden in einer Region selbst bei der Erzeugung flächenmäßig zusammenhängender Verschleierungszonen die Anzahl an in einer Zone enthaltenen Gebäude kein ausreichendes Maß für einen gleichbleibenden Grad an Standortanonymität darstellt. Auch dieses Problem löst LAMA LocO durch die vorausschauende, vollständige Partitionierung der Karte und die dadurch erreichte Reziprozität der Verschleierungszonen.

Darüber hinaus wurden erstmals Strategien zur konsequenten Einhaltung der k-Anonymität aller Aufenthaltsorte eines Nutzers bei der kontinuierlichen Freigabe von Standortinformationen vorgeschlagen und der dabei zu erwartende durchschnittliche Qualitätsverlust analysiert. Durch die vorausschauende Partitionierung der Karte und dem vorsichtigen Umgang mit Standortinformationen werden dabei sogar unvorhergesehene, spontane Aufenthaltsorte ohne Interaktion des Nutzers automatisch anonymisiert.

Abschließend wurde gezeigt, dass sich auf Basis der starken Verschleierungszonen auch die online Routenplanung privatsphäreschonend und im Großteil der Fälle mit hoher Dienstqualität durchführen lässt. Im Gegensatz zu PrOSPR genügt nun stets eine Anfrage, die zu diesem Zweck an die standardmäßigen LBS-Schnittstellen übermittelt wird.

4.8 Zusammenfassung

In diesem Kapitel wurden verschiedene Verfahren zum Schutz der Privatsphäre in ortsbezogenen Diensten vorgestellt. PrOSPR und die Erweiterung PrOSPR+ ermöglichen die privatsphäreschonende Ermittlung von schnellsten Routen auf Basis der Echtzeit-Verkehrsdaten eines beliebigen online Routenplaners. Weder der Dienstanbieter selbst noch irgendein Angreifer, der die Kommunikation des Nutzers belauscht, ist dabei in der Lage, Start- und Zieladresse einer Routenanfrage auf weniger als k Kandidaten einzugrenzen. Hierfür wurde das von den etablierten Schutzmechanismen der k-Anonymität sowie der l-Diversität abgeleitete Konzept der k-immunen Routenanfragen eingeführt, das von PrOSPR konsequent durchgesetzt wird.

Auch die Zielsetzung, dass es keinen per Definition vertrauenswürdigen Dritten geben soll, sondern nur das Endgerät des Nutzers selbst die exakten Routenendpunkte kennen darf, wird erfüllt. Um praktisch einsetzbar zu sein, wurde

das Verfahren so konzipiert, dass es ausschließlich Punkt-zu-Punkt-Anfragen verwendet, die von den verfügbaren online Routenplanern tatsächlich angeboten werden. Im Gegensatz zu vielen bestehenden Arbeiten kommt das vorgestellte System ohne die Existenz einer TTP und ohne eine über die übliche Diensterbringung hinausgehende Kooperation seitens des Dienstanbieters aus.

Anhand einer umfangreichen Evaluation konnte gezeigt werden, dass sich die privatsphäreschonende Routenplanung mit PrOSPR unter Inkaufnahme kurzer Umwege im Bereich von durchschnittlich etwa 30 Sekunden Dauer und eines akzeptablen Kommunikationsoverheads im meist niedrigen dreistelligen Kilobyte-Bereich praktisch einsetzen lässt.

Entsprechend der in Kapitel 2.1.2 vorgenommenen Kategorisierung kontextbezogener Dienste handelt es sich bei der Routenplanung um eine sporadische, unkorrelierte LBS-Nutzung. Anders verhält sich dies bei Diensten, die eine periodische oder kontinuierliche Übermittlung der aktuellen Position an den Dienstanbieter benötigen. Auch für diese LBS-Klasse wurde mit LAMA LocO eine umfassende Lösung präsentiert, die auf Basis einer topologiebezogenen Erstellung von Verschleierungszonen und verschiedenen Strategien zur Freigabe von Standortinformationen in unterschiedlichen Nutzungsszenarien einen kontinuierlichen, dienstübergreifenden Schutz der Standortanonymität eines Nutzers gewährleistet.

LAMA LocO stellt das erste Verfahren zur Verschleierung der Standortinformationen eines Nutzers dar, das den kontinuierlichen Schutz der Privatsphäre auf Basis der ortsbasierten k-Anonymität gewährleistet. Die vorgeschlagene Lösung folgt dabei dem Verständnis, dass sich die ortsbasierte Inferenz von Nutzeridentität und -interessen durch einen realistischen Angreifer nur durch die Berücksichtigung von Kartenwissen effektiv verhindern lassen. Wie gezeigt, wirkt sich der Einbezug der Straßentopologie in die Erzeugung von Verschleierungszonen im Vergleich zu bestehenden Verfahren zudem positiv auf die durchschnittlich erreichbare Dienstqualität aus.

Anhand der vorgeschlagenen Verschleierungsstrategien zeigt sich zudem, wie wichtig „Kontext" für den effektiven Schutz der Privatsphäre ist. Denn das Hintergrundwissen, auf dem die vorgestellten Freigabestrategien aufbauen, berücksichtigt Kontextinformationen wie: Wo kommt der Nutzer her? Wo möchte er hin? Befindet er sich derzeit statisch an einem Aufenthaltsort oder ist er in Bewegung? Wie lange hat der letzte Aufenthalt gedauert?

Ähnlich wie es ALPACA bei der Ermöglichung kontextabhängiger Freigaberegeln berücksichtigt, zeigt sich somit, dass auch bei algorithmischen Verfahren zum Schutz der Privatsphäre in ortsbezogenen Diensten der aktuelle Kontext des Nutzers eine entscheidende Rolle spielt. Im Gegensatz zur manuellen Erstellung von Freigaberegeln kümmert sich LAMA LocO jedoch automatisch um die korrekte Berücksichtigung des aktuellen Kontexts. Der Nutzer muss sich lediglich einmalig für einen Grad an Standortanonymität k entscheiden, der ab diesem Moment konsequent eingehalten wird.

5 Zusammenfassung und Ausblick

„Tell a little truth with many lies
It's the only way I've found."
— Dio, *Straight through the heart*

Bei der Nutzung kontextabhängiger und ortsbezogener Dienste fällt eine große Anzahl an persönlichen Daten an. Derartige Informationen stellen „das Öl des Internets und neue Währung der digitalen Welt" [153] dar. Sie müssen daher effektiv geschützt werden. In der vorliegenden Arbeit wurden hierfür unterschiedliche Lösungsansätze vorgestellt und bewertet.

Während das erste System aufgrund seines hohen, modellierungsbedingten Abstraktionsgrades allgemeingültig für die privatsphärezentrische Verwaltung verschiedenster Typen von Kontextinformationen angewendet werden kann, stellen die anderen beiden Ansätze spezialisierte Mechanismen für die Herstellung von Standortanonymität in ortsbezogenen Diensten dar. Der letzte Ansatz lässt sich wieder generisch für eine Vielzahl unterschiedlicher LBS-Szenarien anwenden. Durch die Spezialisierung auf Standort- und Bewegungsdaten kommen die beiden letzten Verfahren im Vergleich zum ersten Ansatz mit deutlich weniger Benutzerinteraktion aus und arbeiten nach minimalem Konfigurationsaufwand vollkommen selbständig.

Abschließend werden die wichtigsten Erkenntnisse noch einmal zusammengefasst und konkrete Richtungen für weitere Forschungsfragen gegeben.

5.1 Ergebnisse der vorliegenden Arbeit

Ausgehend von einer Definition und grundlegenden Charakterisierung kontextbezogener Anwendungen wurden zunächst aktuelle Möglichkeiten zur Kontexterkennung auf mobilen Endgeräten beschrieben. Es wurde gezeigt, dass die Ermittlung von Kontextinformationen und die Umsetzung komplexer, kontextabhängiger Anwendungen längst keine Visionen mehr sind, sondern Realität.

Gleichzeitig wurde verdeutlicht, wie elementar der Schutz der Privatsphäre bei der Benutzung kontext- und ortsbezogener Dienste ist. Dies gilt sowohl für im Berufsalltag eingesetzte Tracking-Systeme als auch für die Vielzahl an persönlich genutzten „Consumer-Apps", die sich auf unseren Smartphones befinden und laufend mit persönlichen Kontextinformationen hantieren.

Im Idealfall hat der Nutzer stets die Möglichkeit, diesen Diensten nur so viele Informationen zukommen zu lassen, wie es zur adäquaten Diensterbringung unbedingt nötig ist und es seinen individuellen Privatsphärebedürfnissen nicht entgegensteht. Zur Erreichung dieses Ziels leistet die vorliegende Arbeit verschiedene Beiträge.

Mit ALPACA wurde ein umfassendes System für die privatsphärezentrische Verwaltung von Kontextinformationen auf einem mobilen Endgerät entwickelt. Das enthaltene Kontextmodell ALPACA-CoRe ist so ausgelegt, dass es im Zusammenspiel mit dem vorgestellten kontextabhängigen Triggermechanismus die situationsabhängige und feingranulare Freigabe von Kontextinformationen an einen bestimmten Empfänger erlaubt. Mögliche Inkonsistenzen, welche die Privatsphäre oder Integrität des Nutzers verletzen könnten, werden auf Basis der formalen Modellzusammenhänge automatisch erkannt.

Neben einer Lösung für die transparente Beantwortung von Kontextanfragen durch bestehende Anwendungen wurde ein Peer-to-Peer-basiertes Kommunikationsprotokoll für den Austausch von Kontextdaten entwickelt, das eine Ausgewogenheit im Informationsfluss zwischen den Parteien gewährleistet. Zudem wurde mit dem Privacy Manager eine zentrale Komponente vorgestellt, die sich um die Kontextakquise und die Durchsetzung der Freigabeentscheidung des Nutzers kümmert. Sie lässt sich nahtlos in ein aktuelles mobiles Betriebssystem integrieren. Im Vergleich zu den heute verfügbaren Optionen mobiler Betriebssysteme ermöglicht ALPACA eine feingranulare, situationsabhängige und verschleierte Freigabe von Informationen – nicht nur das An- oder Abschalten der GPS-Ortung für eine bestimmte App.

PrOSPR sowie die Erweiterung PrOSPR+ bieten eine Lösung für die privatsphäreschonende Nutzung von standardmäßig verfügbaren, kommerziellen online Routenplanern. Diese spezielle LBS-Ausprägung zeichnet sich dadurch aus, dass sie stets höchst präzise und aktuelle Standortangaben benötigt, um qualitativ hochwertige Ergebnisse zu liefern.

Insbesondere im Falle von kostenlosen, werbefinanzierten Angeboten kann dem Anbieter ortsbezogener Dienste eine gewisse Neugierde hinsichtlich der Vorlieben seiner Benutzer unterstellt werden. Daher wird bei PrOSPR durchgängig darauf geachtet, dass die Aufenthaltsorte eines Benutzers bzw. die von ihm angefragten Routenendpunkte von etwaigen Lauschern oder dem Dienstanbieter selbst durch die in den Anfragen enthaltenen Informationen nicht auf weniger als jeweils k Kandidaten eingegrenzt werden können.

Es konnte gezeigt werden, dass sich derartige Dienste auch unter einem objektiv nachvollziehbaren Grad an Privatsphäre wie der k-Anonymität von Standortinformationen qualitativ hochwertig nutzen lassen. Mit verschiedenen Optimierungen lässt sich der nötige zusätzliche Kommunikationsaufwand auf ein praxistaugliches Maß reduzieren.

Aufbauend auf den Erkenntnissen, die bei der Umsetzung der privatsphäreschonenden Routenplanung gewonnen werden konnten, wurde schließlich ein dritter Ansatz präsentiert. Auch LAMA LocO befasst sich mit dem Schutz der Aufenthaltsorte eines mobilen Nutzers in ortsbezogenen Diensten, geht jedoch einen wichtigen Schritt weiter, indem es auch die konsistente Verschleierung von Standortinformationen in kontinuierlichen LBS ermöglicht.

Bereits bei der Erstellung von Verschleierungszonen wird hierfür neben der Bedingung, dass sich stets mindestens k Orte in einem Verschleierungsset befinden müssen, auch die zugrundeliegende Topologie des Straßennetzes mit einbezogen. Den Anforderungen unterschiedlicher LBS-Ausprägungen an die Aktualität und Genauigkeit von Ortsangaben entsprechend, wurden verschiedene Strategien zur Verschleierung der Standortinformationen eines Nutzers entwickelt. Als Ergebnis lässt sich LAMA LocO generisch für beliebige Typen ortsbezogener Dienste einsetzen, um die kontinuierliche Standortanonymität des Nutzers dienstübergreifend zu gewährleisten.

Für verschiedene Szenarien wurde untersucht, mit welchem Qualitätsverlust bei der Benutzung ortsbezogener Dienste zu rechnen ist. Es konnte gezeigt werden, dass die topologiebezogene Zonenerstellung zu vergleichbaren und oft sogar geringeren QoS-Abweichungen führt als bestehende Verfahren, dabei aber stärkere Garantien hinsichtlich der tatsächlichen Standortanonymität gibt.

5.2 Anknüpfungspunkte für weitere Arbeiten

In der vorliegenden Arbeit wurden offene Probleme hinsichtlich des Privatsphäreschutzes in kontext- und ortsabhängigen Diensten identifiziert und verschiedene Lösungsansätze entwickelt. Die Fähigkeiten der vorgestellten Ansätze übertreffen die bestehender Systeme, lassen aufgrund der großen Komplexität des Themenbereichs jedoch noch einigen Spielraum für weitere Entwicklungen. Abschließend werden nun Schwierigkeiten beschrieben, die bei der Arbeit an den vorgestellten Verfahren augenscheinlich wurden. Damit wird ein möglicher Ausblick auf künftige Forschungsarbeiten in diesem Themenfeld gegeben.

Wie bereits in der Diskussion des ALPACA-Systems erwähnt, ist davon auszugehen, dass der durchschnittliche Nutzer von der gebotenen Vielzahl an Freiheitsgraden bei der Freigabeentscheidung bzgl. einzelner Kontextinformationen in verschiedenen Situationen überfordert oder abgeschreckt sein könnte.

Ein Grund hierfür ist die Komplexität der Regelerstellung selbst. Mögliche Lösungsansätze sind in der Bereitstellung von Default-Profilen [35] sowie in der Machine-Learning- [30] oder Crowdsourcing-basierten Prädiktion und Empfehlung von Privacy-Präferenzen zu sehen [256, 265].

Diese Verfahren sind jedoch noch weitgehend experimentell, benötigen eine manuelle Feinjustierung, verlangen eine lange Trainingsphase und rufen selbst privatsphärebezogene Bedenken hervor, die aus der Veröffentlichung solcher

persönlichen Regeln entstehen. Solche Techniken gilt es daher weiter zu erforschen und zu verfeinern – genau wie intuitive, graphische Nutzerschnittstellen, die die Regeldefinition erleichtern. Stehen derartige Hilfsmittel zuverlässig zur Verfügung, bietet ALPACA ein praktisch einsetzbares, umfassendes Werkzeug zur kontextabhängigen Verwaltung von persönlichen Informationen, dessen Flexibilität und Ausdrucksstärke bestehenden Ansätzen überlegen ist.

Des Weiteren ist es wichtig, den Benutzer sowohl über die möglichen Gefahren der Datensammlung eines bestimmten Typs von Kontextinformationen aufzuklären als auch bei der Auswahl von Verschleierungstechniken zu unterstützen [37]. Hierfür muss eine verständliche Darstellung möglicher Angriffe, Schutzmechanismen und des ggf. zu erwartenden Qualitätsverlusts erfolgen, um dem Nutzer eine effektive und sinnvolle Wahl zu ermöglichen.

Speziell in Bezug auf den Schutz von Standortinformationen ist ein weiterer Angriffspunkt in den unterschiedlichen Semantiken von Gebäuden zu sehen. In dieser Arbeit wurde davon ausgegangen, dass jedes Gebäude einer Zone ein gleich wahrscheinlicher Aufenthaltsort eines Nutzers ist. Ein Angreifer, der weiteres Hintergrundwissen über den Kontext eines Aufenthalts sowie die Semantik der einzelnen Orte besitzt, ist dazu in der Lage, einige Orte anhand dieses Wissens als unplausibel auszuschließen. So gibt es Gebäude, die je nach Tageszeit nicht öffentlich begehbar oder unwahrscheinliche Ziele sind. Bestimmte Orte wie eine Kirche haben regelmäßige Besuchszeiten, die sich von den Gebäuden in der Nachbarschaft i.d.R. deutlich unterscheiden.

Zudem ist davon auszugehen, dass semantische Orte in Abhängigkeit von ihrer Häufigkeit und ihrer Semantik unterschiedliche Muster aufweisen, was die typischen Anreisewege, Aufenthaltsdauern und Besuchsregelmäßigkeiten betrifft. Für eine Bäckerei wird ein Nutzer kaum dutzende oder gar hunderte Kilometer weit fahren – für eine Spezialklinik oder ein Hotel hingegen schon. Dort hält er sich im Gegenzug dafür üblicherweise deutlich länger auf.

Im Rahmen der vorliegenden Arbeit wurde diesem Problem implizit durch die Wahl eines hohen Wertes für k zwischen 50 und 200 begegnet. Selbst wenn ein Angreifer einige Gebäude ausschließen kann, ist daher davon auszugehen, dass keine eindeutige Identizifierung des tatsächlichen Aufenthaltsortes gelingt. Dieser Aspekt sollte in Zukunft jedoch genauer betrachtet werden. Die topologiebezogene Erzeugung von Verschleierungszonen von LAMA LocO kann diesbezüglich z.B. erweitert werden, indem anstelle des Parameters k auf die Erfüllung anderweitig formulierter Bedingungen geachtet wird, die derartige Eigenschaften der einzelnen semantischen Orte miteinbeziehen.

Literaturverzeichnis

[1] *Persönlichkeitssphäre*. In: Wirtz, M. A. (Hrsg.): *Dorsch Lexikon der Psychologie*, S. 364. Verlag Hans Huber, 2013.

[2] Abowd, G. D., A. K. Dey, P. J. Brown, N. Davies, M. Smith und P. Steggles: *Towards a better understanding of context and context-awareness*. In: *Handheld and ubiquitous computing*, S. 304–307. Springer, 1999.

[3] Ackerman, M., T. Darrell und D. Weitzner: *Privacy in Context*. Hum.-Comput. Interact., 16(2):167–176, Dez. 2001.

[4] Amini, S., J. Lindqvist, J. Hong, J. Lin, E. Toch und N. Sadeh: *Caché: caching location-enhanced content to improve user privacy*. In: *Proceedings of the 9th international conference on Mobile systems, applications, and services*, S. 197–210. ACM, 2011.

[5] Amir, A., A. Efrat, J. Myllymaki, L. Palaniappan und K. Wampler: *Buddy tracking—efficient proximity detection among mobile friends*. Pervasive and Mobile Computing, 3(5):489–511, 2007.

[6] Andrés, M. E., N. E. Bordenabe, K. Chatzikokolakis und C. Palamidessi: *Geo-indistinguishability: Differential Privacy for Location-based Systems*. In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, CCS '13, S. 901–914, New York, NY, USA, 2013. ACM.

[7] Anjum, A. und M. U. Ilyas: *Activity recognition using smartphone sensors*. In: *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, S. 914–919, Jan 2013.

[8] Apolinarski, W., M. Handte, D. Phuoc und P. J. Marrón: *Modeling and Using Context: 7th International and Interdisciplinary Conference, CONTEXT 2011, Karlsruhe, Germany, September 26-30, 2011. Proceedings*, Kap. A Peer-Based Approach to Privacy-Preserving Context Management, S. 18–25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[9] Ardagna, C. A., M. Cremonini, E. Damiani, S. D. C. di Vimercati und P. Samarati: *Location Privacy Protection Through Obfuscation-based Techniques*. In: *Proceedings of the 21st Annual IFIP WG 11.3*

Working Conference on Data and Applications Security, S. 47–60, Berlin, Heidelberg, 2007. Springer-Verlag.

[10] Ardagna, C. A., M. Cremonini und G. Gianini: *Landscape-aware Location-privacy Protection in Location-based Services*. J. Syst. Archit., 55(4):243–254, Apr. 2009.

[11] Ardagna, C. A., M. Cremonini, S. D. C. di Vimercati und P. Samarati: *An Obfuscation-Based Approach for Protecting Location Privacy*. IEEE Transactions on Dependable and Secure Computing, 8(1):13–27, 2011.

[12] Ardagna, C. A., G. Livraga und P. Samarati: *Protecting Privacy of User Information in Continuous Location-Based Services*. In: *Computational Science and Engineering (CSE), 2012 IEEE 15th International Conference on*, S. 162–169, Dec 2012.

[13] Assange, J., J. Appelbaum, A. Müller-Maguhn und J. Zimmermann: *Cypherpunks: Unsere Freiheit und die Zukunft des Internets*. Campus Verlag, Frankfurt/New York, 2012. S. 74.

[14] Bagüés, S. A., A. Zeidler, C. F. Valdivielso und I. R. Matias: *Disappearing for a While - Using White Lies in Pervasive Computing*. In: *Proceedings of the 2007 ACM Workshop on Privacy in Electronic Society*, WPES '07, S. 80–83, New York, NY, USA, 2007. ACM.

[15] Balakrishnan, D., M. El Barachi, A. Karmouch und R. Glitho: *Challenges in Modeling and Disseminating Context Information in Ambient Networks*. In: *Proceedings of the Second International Conference on Mobility Aware Technologies and Applications*, MATA'05, S. 32–42, Berlin, Heidelberg, 2005. Springer-Verlag.

[16] Baldauf, M., S. Dustdar und F. Rosenberg: *A Survey on Context-Aware Systems*. Int. J. Ad Hoc Ubiquitous Comput., 2(4):263–277, Juni 2007.

[17] Balebako, R., J. Jung, W. Lu, L. F. Cranor und C. Nguyen: *"Little Brothers Watching You": Raising Awareness of Data Leaks on Smartphones*. In: *Proceedings of the Ninth Symposium on Usable Privacy and Security*, SOUPS '13, S. 12:1–12:11, New York, NY, USA, 2013. ACM.

[18] Bamba, B., L. Liu, P. Pesti und T. Wang: *Supporting Anonymous Location Queries in Mobile Environments with Privacygrid*. In: *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, S. 237–246, New York, NY, USA, 2008. ACM.

[19] Barkhuus, L.: *Privacy in location-based services, concern vs. coolness*. Workshop on Location System Privacy and Control at MobileHCI, 4, 2004.

[20] Bast, H., D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner und R. Werneck: *Route Planning in Transportation Networks*. Techn. Ber. MSR-TR-2014-4, January 2014.

[21] Bäumker, M.: *Hybride Navigationssysteme für Navigation, Regelung und direkte Georeferenzierung*. Zeitschrift für Vermessungswesen, 5:303–312, 2013.

[22] Behrooz, A. und A. Devlic: *Security and Privacy in Mobile Information and Communication Systems: Third International ICST Conference, MobiSec 2011, Aalborg, Denmark, May 17-19, 2011, Revised Selected Papers*, Kap. A Context-Aware Privacy Policy Language for Controlling Access to Context Information of Mobile Users, S. 25–39. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[23] Bellavista, P., A. Corradi, M. Fanelli und L. Foschini: *A Survey of Context Data Distribution for Mobile Ubiquitous Systems*. ACM Comput. Surv., 44(4):24:1–24:45, Sep. 2012.

[24] Benisch, M., P. G. Kelley, N. Sadeh, T. Sandholm, J. Tsai, L. F. Cranor und P. H. Drielsma: *The Impact of Expressiveness on the Effectiveness of Privacy Mechanisms for Location-sharing*. In: *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, S. 22:1–22:1, New York, NY, USA, 2009. ACM.

[25] Beresford, A. R., A. Rice, N. Skehin und R. Sohan: *MockDroid: Trading Privacy for Application Functionality on Smartphones*. In: *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, HotMobile '11, S. 49–54, New York, NY, USA, 2011. ACM.

[26] Beresford, A. R. und F. Stajano: *Location Privacy in Pervasive Computing*. IEEE Pervasive Computing, 2(1):46–55, Jan. 2003.

[27] Berners-Lee, T., J. Hendler, O. Lassila et al.: *The semantic web*. Scientific american, 284(5):28–37, 2001.

[28] Bettini, C., O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan und D. Riboni: *A Survey of Context Modelling and Reasoning Techniques*. Pervasive Mob. Comput., 6(2):161–180, Apr. 2010.

[29] Bettini, C., X. S. Wang und S. Jajodia: *Protecting Privacy Against Location-based Personal Identification*. In: *Proceedings of the Second*

VDLB International Conference on Secure Data Management, SDM'05, S. 185–199, Berlin, Heidelberg, 2005. Springer-Verlag.

[30] Bigwood, G., F. B. Abdesslem und T. Henderson: *Predicting location-sharing privacy preferences in social network applications*. Proc. of AwareCast, 12:1–12, 2012.

[31] Bilogrevic, I., M. Jadliwala, I. Lám, I. Aad, P. Ginzboorg, V. Niemi, L. Bindschaedler und J.-P. Hubaux: *Pervasive Computing: 10th International Conference, Pervasive 2012, Newcastle, UK, June 18-22, 2012. Proceedings*, Kap. Big Brother Knows Your Friends: On Privacy of Social Communities in Pervasive Networks, S. 370–387. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[32] Bindschaedler, V. und R. Shokri: *Privacy through Fake yet Semantically Real Traces*. CoRR, abs/1505.07499, 2015.

[33] Bissmeyer, N., S. Mauthofer, K. M. Bayarou und F. Kargl: *Assessment of node trustworthiness in VANETs using data plausibility checks with particle filters*. In: *Vehicular Networking Conference (VNC), 2012 IEEE*, S. 78–85, Nov 2012.

[34] Blount, M., J. Davis, M. Ebling, W. Jerome, B. Leiba, X. Liu und A. Misra: *Privacy Engine for Context-Aware Enterprise Application Services*. In: *Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on*, Bd. 2, S. 94–100, Dec 2008.

[35] Bokhove, W., B. Hulsebosch, B. Van Schoonhoven, M. Sappelli und K. Wouters: *User Privacy in Applications for Well-being and Well-working*. Proc. AMBIENT 2012, S. 53–59, 2012.

[36] Bolchini, C., C. A. Curino, E. Quintarelli, F. A. Schreiber und L. Tanca: *A Data-oriented Survey of Context Models*. SIGMOD Rec., 36(4):19–26, Dez. 2007.

[37] Brush, A. B., J. Krumm und J. Scott: *Exploring End User Preferences for Location Obfuscation, Location-based Services, and the Value of Location*. In: *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, UbiComp '10, S. 95–104, New York, NY, USA, 2010. ACM.

[38] Buluç, A., H. Meyerhenke, I. Safro, P. Sanders und C. Schulz: *Recent Advances in Graph Partitioning*. In: *Algorithm Engineering: Selected Results and Surveys, LNCS 9220*. Springer-Verlag, 2015 (in press).

[39] Burghardt, T., E. Buchmann, J. Müller und K. Böhm: *On the Move to Meaningful Internet Systems: OTM 2009: Confederated International Conferences, CoopIS, DOA, IS, and ODBASE 2009, Vilamoura,*

Portugal, November 1-6, 2009, Proceedings, Part I, Kap. Understanding User Preferences and Awareness: Privacy Mechanisms in Location-Based Services, S. 304–321. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[40] Cavoukian, A.: *Privacy by design: the definitive workshop. A foreword by Ann Cavoukian, Ph.D.* Identity in the Information Society, 3(2):247–251, 2010.

[41] CBC: *Smart watch alerts user to impending heart attack.* CBC/Radio-Canada, 17. März 2016. `http://www.cbc.ca/player/play/2685473091/`, letzter Abruf: 17.7.2016.

[42] Chapman University: *The Chapman University Survey on American Fears, Wave 2. Orange, CA: Earl Babbie Research Center [producer]*, 2015. `http://www.chapman.edu/wilkinson/research-centers/babbie-center/survey-american-fears.aspx`, letzter Abruf: 17.7.2016.

[43] Chatzikokolakis, K., C. Palamidessi und M. Stronati: *Distributed Computing and Internet Technology: 11th International Conference, ICDCIT 2015, Bhubaneswar, India, February 5-8, 2015. Proceedings*, Kap. Geo-indistinguishability: A Principled Approach to Location Privacy, S. 49–72. Springer International Publishing, Cham, 2015.

[44] Chen, H., T. Finin und A. Joshi: *An ontology for context-aware pervasive computing environments.* The Knowledge Engineering Review, 18(03):197–207, 2003.

[45] Chen, H., T. Finin und A. Joshi: *A Pervasive Computing Ontology for User Privacy Protection in the Context Broker Architecture.* Techn. Ber. TR-CS-04-08, University of Maryland, Baltimore County, July 2004.

[46] Chen, H., T. Finin und A. Joshi: *Ontologies for Agents: Theory and Experiences*, Kap. The SOUPA Ontology for Pervasive Computing, S. 233–258. Birkhäuser Basel, Basel, 2005.

[47] Chen, H., F. Perich, T. Finin und A. Joshi: *SOUPA: standard ontology for ubiquitous and pervasive applications.* In: *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, S. 258–267, Aug 2004.

[48] Chow, C.-Y. und M. F. Mokbel: *Enabling Private Continuous Queries for Revealed User Locations.* In: *Proceedings of the 10th International Conference on Advances in Spatial and Temporal Databases*, SSTD'07, S. 258–273, Berlin, Heidelberg, 2007. Springer-Verlag.

[49] Chow, C.-Y., M. F. Mokbel und X. Liu: *A Peer-to-peer Spatial Cloaking Algorithm for Anonymous Location-based Service*. In: *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems*, GIS '06, S. 171–178, New York, NY, USA, 2006. ACM.

[50] Christin, D., A. Reinhardt, S. S. Kanhere und M. Hollick: *A Survey on Privacy in Mobile Participatory Sensing Applications*. J. Syst. Softw., 84(11):1928–1946, Nov. 2011.

[51] Christl, W.: *Kommerzielle digitale Überwachung im Alltag*. Studie im Auftrag der Österreichischen Bundesarbeitskammer, Wien, AT, November 2014.

[52] Constandache, I., R. R. Choudhury und I. Rhee: *Towards Mobile Phone Localization without War-Driving*. In: *INFOCOM, 2010 Proceedings IEEE*, S. 1–9, March 2010.

[53] Damiani, M. L.: *Location privacy models in mobile applications: conceptual view and research directions*. GeoInformatica, 18(4):819–842, 2014.

[54] Damiani, M. L., E. Bertino und C. Silvestri: *The PROBE Framework for the Personalized Cloaking of Private Locations*. Trans. Data Privacy, 3(2):123–148, Aug. 2010.

[55] Das, A. S., M. Datar, A. Garg und S. Rajaram: *Google News Personalization: Scalable Online Collaborative Filtering*. In: *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, S. 271–280, New York, NY, USA, 2007. ACM.

[56] Delling, D., A. V. Goldberg, A. Nowatzyk und R. F. Werneck: *PHAST: Hardware-Accelerated Shortest Path Trees*. Techn. Ber. MSR-TR-2010-125, September 2010.

[57] Delling, D., A. V. Goldberg, T. Pajor und R. F. Werneck: *Customizable route planning in road networks*. Transportation Science, 2015.

[58] Delling, D. und D. Wagner: *Landmark-based Routing in Dynamic Graphs*. In: *Proceedings of the 6th International Conference on Experimental Algorithms*, WEA'07, S. 52–65, Berlin, Heidelberg, 2007. Springer-Verlag.

[59] Dewri, R.: *Local Differential Perturbations: Location Privacy Under Approximate Knowledge Attackers*. IEEE Transactions on Mobile Computing, 12(12):2360–2372, Dez. 2013.

[60] Dey, A. K.: *Providing architectural support for building context-aware applications*. Doktorarbeit, Georgia Institute of Technology, 2000.

[61] Dijkstra, E. W.: *A note on two problems in connexion with graphs*. Numerische Mathematik, 1(1):269–271.

[62] Dingledine, R., N. Mathewson und P. Syverson: *Tor: The Second-generation Onion Router*. In: *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, SSYM'04, S. 21–21, Berkeley, CA, USA, 2004. USENIX Association.

[63] Do, H. J., Y. S. Jeong, H.-J. Choi und K. Kim: *Another dummy generation technique in location-based services*. In: *2016 International Conference on Big Data and Smart Computing (BigComp)*, S. 532–538, Jan 2016.

[64] Domingo-Ferrer, J., A. Solanas und J. Castellà-Roca: *h(k)-private information retrieval from privacy-uncooperative queryable databases*. Online Information Review, 33(4):720–744, 2009.

[65] Dorfmeister, F., S. Feld und C. Linnhoff-Popien: *ALPACA: A Decentralized, Privacy-Centric and Context-Aware Framework for the Dissemination of Context Information*. In: *Band 7 Nummer 1 und 2 von International Journal On Advances in Intelligent Systems*, S. 223–236, 2014.

[66] Dorfmeister, F., S. Feld, C. Linnhoff-Popien und S. Verclas: *Privacy-Centric Modeling and Management of Context Information*. In: *CENTRIC 2013, The Sixth International Conference on Advances in Human oriented and Personalized Mechanisms, Technologies, and Services*, S. 92–97, 2013.

[67] Dorfmeister, F., M. Maier, M. Schönfeld und S. A. W. Verclas: *SmartBEEs: Enabling Smart Business Environments Based on Location Information and Sensor Networks*. In: *9. GI/ITG KuVS Fachgespräch Ortsbezogene Anwendungen und Dienste*, S. 43–56. Universitätsverlag Chemnitz, sep 2012. Chemnitz, Germany.

[68] Dorfmeister, F., K. Wiesner, M. Schuster und M. Maier: *Preventing Restricted Space Inference in Online Route Planning Services*. In: *Proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MOBIQUITOUS '15, S. 209–218, ICST, Brussels, Belgium, Belgium, 2015. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[69] Duckham, M. und L. Kulik: *A Formal Model of Obfuscation and Negotiation for Location Privacy*. In: *Proceedings of the Third International Conference on Pervasive Computing*, PERVASIVE'05, S. 152–170, Berlin, Heidelberg, 2005. Springer-Verlag.

[70] Duckham, M. und L. Kulik: *Location privacy and location-aware computing*. Dynamic & mobile GIS: investigating change in space and time, 3:35–51, 2006.

[71] Dwork, C.: *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, Kap. Differential Privacy, S. 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[72] Dürr, M., M. Maier und F. Dorfmeister: *Vegas–A Secure and Privacy-Preserving Peer-to-Peer Online Social Network*. In: *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, S. 868–874. IEEE, 2012.

[73] Ebert, A., F. Dorfmeister, M. Maier und C. Linnhoff-Popien: *EMMA: A Context-Aware Middleware for Energy Management on Mobile Devices*. In: *CENTRIC 2014, The Seventh International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services*, S. 48–53, 2014.

[74] Eckert, W.: *Entwicklung und Vergleich von Verfahren zum Schutz der Privatsphäre in kontinuierlichen Location-Based Services*. Diplomarbeit, LMU München, München, Deutschland, 2015.

[75] Ekstrand, M. D., J. T. Riedl und J. A. Konstan: *Collaborative filtering recommender systems*. Foundations and Trends in Human-Computer Interaction, 4(2):81–173, 2011.

[76] Enck, W., P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel und A. N. Sheth: *TaintDroid: An Information-flow Tracking System for Realtime Privacy Monitoring on Smartphones*. In: *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, OSDI'10, S. 1–6, Berkeley, CA, USA, 2010. USENIX Association.

[77] Eriksson, J., L. Girod, B. Hull, R. Newton, S. Madden und H. Balakrishnan: *The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring*. In: *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, S. 29–39, New York, NY, USA, 2008. ACM.

[78] Fawaz, K., H. Feng und K. G. Shin: *Anatomization and Protection of Mobile Apps' Location Privacy Threats*. In: *24th USENIX Security Symposium (USENIX Security 15)*, S. 753–768, 2015.

[79] Fawaz, K. und K. G. Shin: *Location privacy protection for smartphone users*. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, S. 239–250. ACM, 2014.

[80] Freedman, M. J., K. Nissim und B. Pinkas: *Advances in Cryptology - EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings*, Kap. Efficient Private Matching and Set Intersection, S. 1–19. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[81] Fuest, B.: *Wie man Staus schon vor Entstehung umfahren kann*. Die Welt, 1. Apr. 2014. `http://www.welt.de/wirtschaft/article126410320/Wie-man-Staus-schon-vor-Entstehung-umfahren-kann.html`, letzter Abruf: 17.7.2016.

[82] Gambs, S., M. O. Killijian und M. N. d. P. Cortez: *De-anonymization Attack on Geolocated Data*. In: *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, S. 789–797, July 2013.

[83] Gambs, S., M.-O. Killijian und M. N. n. del Prado Cortez: *Show Me How You Move and I Will Tell You Who You Are*. In: *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, SPRINGL '10, S. 34–41, New York, NY, USA, 2010. ACM.

[84] Ganti, R. K., N. Pham, Y.-E. Tsai und T. F. Abdelzaher: *PoolView: Stream Privacy for Grassroots Participatory Sensing*. In: *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, S. 281–294, New York, NY, USA, 2008. ACM.

[85] Gao, S., J. Ma, W. Shi, G. Zhan und C. Sun: *TrPF: A Trajectory Privacy-Preserving Framework for Participatory Sensing*. IEEE Transactions on Information Forensics and Security, 8(6):874–887, June 2013.

[86] Gao, X., B. Firner, S. Sugrim, V. Kaiser-Pendergrast, Y. Yang und J. Lindqvist: *Elastic Pathing: Your Speed is Enough to Track You*. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, S. 975–986, New York, NY, USA, 2014. ACM.

[87] Gedik, B. und L. Liu: *Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms*. IEEE Transactions on Mobile Computing, 7(1):1–18, Jan 2008.

[88] Geisberger, R., P. Sanders, D. Schultes und C. Vetter: *Exact Routing in Large Road Networks Using Contraction Hierarchies*. Transportation Science, 46(3):388–404, Aug. 2012.

[89] Ghinita, G., M. L. Damiani, C. Silvestri und E. Bertino: *Preventing Velocity-based Linkage Attacks in Location-aware Applications*. In: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, S. 246–255, New York, NY, USA, 2009. ACM.

[90] Ghinita, G., P. Kalnis, A. Khoshgozaran, C. Shahabi und K.-L. Tan: *Private Queries in Location Based Services: Anonymizers Are Not Necessary*. In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, S. 121–132, New York, NY, USA, 2008. ACM.

[91] Ghinita, G., P. Kalnis und S. Skiadopoulos: *Advances in Spatial and Temporal Databases: 10th International Symposium, SSTD 2007, Boston, MA, USA, July 16-18, 2007. Proceedings*, Kap. MobiHide: A Mobilea Peer-to-Peer System for Anonymous Location-Based Queries, S. 221–238. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[92] Ghinita, G., K. Zhao, D. Papadias und P. Kalnis: *A Reciprocal Framework for Spatial K-anonymity*. Inf. Syst., 35(3):299–314, Mai 2010.

[93] Ghosh, D., A. Joshi, T. Finin und P. Jagtap: *Privacy Control in Smart Phones Using Semantically Rich Reasoning and Context Modeling*. In: *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*, S. 82–85, May 2012.

[94] Giaglis, G. M., P. Kourouthanassis und A. Tsamakos: *Towards a classification framework for mobile location services*. Mobile commerce: technology, theory, and applications, S. 67–85, 2003.

[95] Gkoulalas-Divanis, A., P. Kalnis und V. S. Verykios: *Providing K-Anonymity in Location Based Services*. SIGKDD Explor. Newsl., 12(1):3–10, Nov. 2010.

[96] Goldwasser, S.: *Multi Party Computations: Past and Present*. In: *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '97, S. 1–6, New York, NY, USA, 1997. ACM.

[97] Golle, P. und K. Partridge: *On the Anonymity of Home/Work Location Pairs*. In: *Proceedings of the 7th International Conference on Pervasive Computing*, Pervasive '09, S. 390–397, Berlin, Heidelberg, 2009. Springer-Verlag.

[98] Gonzalez, M. C., C. A. Hidalgo und A.-L. Barabasi: *Understanding individual human mobility patterns*. Nature, 453(7196):779–782, 2008.

[99] Goodin, D.: *No, this isn't a scene from Minority Report. This trash can is stalking you*, 9. Aug. 2013. http://arstechnica.com/security/2013/08/no-this-isnt-a-scene-from-minority-report-this-trash-can-is-stalking-you/, letzter Abruf: 17.7.2016.

[100] Google: *System and kernel security*. https://source.android.com/security/overview/kernel-security.html, letzter Abruf: 17.7.2016.

[101] Google: *System Permissions*. http://developer.android.com/guide/topics/security/permissions.html, letzter Abruf: 17.7.2016.

[102] Google: *The bright side of sitting in traffic: Crowdsourcing road congestion data*, 25. Aug. 2009. https://googleblog.blogspot.de/2009/08/bright-side-of-sitting-in-traffic.html, letzter Abruf: 17.7.2016.

[103] Gruber, T. R.: *A translation approach to portable ontology specifications*. Knowledge acquisition, 5(2):199–220, 1993.

[104] Gruteser, M. und D. Grunwald: *Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking*. In: *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, MobiSys '03, S. 31–42, New York, NY, USA, 2003. ACM.

[105] Gruteser, M. und B. Hoh: *On the Anonymity of Periodic Location Samples*. In: *Proceedings of the Second International Conference on Security in Pervasive Computing*, SPC'05, S. 179–192, Berlin, Heidelberg, 2005. Springer-Verlag.

[106] Gruteser, M. und X. Liu: *Protecting Privacy in Continuous Location-Tracking Applications*. IEEE Security & Privacy, 2(2):28–34, 2004.

[107] Grün, G.-C.: *Der beste Staumelder ist das eigene Handy*, 21. Juni 2012. http://www.zeit.de/digital/mobil/2012-06/staudaten-handy/komplettansicht, letzter Abruf: 17.7.2016.

[108] Gu, T., X. H. Wang, H. K. Pung und D. Q. Zhang: *An ontology-based context model in intelligent environments*. In: *Proceedings of communication networks and distributed systems modeling and simulation conference*, Bd. 2004, S. 270–275, 2004.

[109] Gustarini, M. und K. Wac: *Mobile Computing, Applications, and Services: 5th International Conference, MobiCASE 2013, Paris, France, November 7-8, 2013, Revised Selected Papers*, Kap. Smartphone Interactions Change for Different Intimacy Contexts, S. 72–89. Springer International Publishing, Cham, 2014.

[110] Guttman, A.: *R-trees: A Dynamic Index Structure for Spatial Searching*. In: *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, SIGMOD '84, S. 47–57, New York, NY, USA, 1984. ACM.

[111] Han, J., E. Owusu, L. T. Nguyen, A. Perrig und J. Zhang: *ACComplice: Location inference using accelerometers on smartphones*. In: *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*, S. 1–9, Jan 2012.

[112] Hart, P. E., N. J. Nilsson und B. Raphael: *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*. IEEE Transactions on Systems Science and Cybernetics, 4(2):100–107, July 1968.

[113] Heide, F. G.: *TomTom verkaufte Nutzerdaten an Regierung*. Handelsblatt, 28. Apr. 2011. http://www.handelsblatt.com/auto/nachrichten/polizei-optimierte-blitzfallen-tomtom-verkaufte-nutzerdaten-an-regierung/4108426.html, letzter Abruf: 17.7.2016.

[114] Hemminki, S., P. Nurmi und S. Tarkoma: *Accelerometer-based Transportation Mode Detection on Smartphones*. In: *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, S. 13:1–13:14, New York, NY, USA, 2013. ACM.

[115] Hendrix, S.: *Traffic-weary homeowners and Waze are at war, again. Guess who's winning?*, 5. Juni 2016. https://www.washingtonpost.com/local/traffic-weary-homeowners-and-waze-are-at-war-again-guess-whos-winning/2016/06/05/c466df46-299d-11e6-b989-4e5479715b54_story.html, letzter Abruf: 17.7.2016.

[116] Henricksen, K., R. Wishart, T. McFadden und J. Indulska: *Extending context models for privacy in pervasive computing environments*. In: *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, S. 20–24, March 2005.

[117] Hicks, J.: *The Wearable Evolution: Emotion Tracker To Debut At CES 2016*. Forbes, 5. Jan. 2016. http://www.forbes.com/sites/jenniferhicks/2016/01/05/the-wearable-evolution-emotion-tracker-to-debut-at-ces-2016/#2d7ebdd32e21, letzter Abruf: 17.7.2016.

[118] Hilbert, D.: *Gesammelte Abhandlungen: Band III: Analysis · Grundlagen der Mathematik Physik · Verschiedenes Lebensgeschichte*, Kap. Über die stetige Abbildung einer Linie auf ein Flächenstück, S. 1–2. Springer Berlin Heidelberg, Berlin, Heidelberg, 1970.

[119] Hoh, B., M. Gruteser, H. Xiong und A. Alrabady: *Enhancing Security and Privacy in Traffic-Monitoring Systems*. IEEE Pervasive Computing, 5(4):38–46, Okt. 2006.

[120] Hoh, B., M. Gruteser, H. Xiong und A. Alrabady: *Preserving Privacy in Gps Traces via Uncertainty-aware Path Cloaking*. In: *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, S. 161–171, New York, NY, USA, 2007. ACM.

[121] Holland, M.: *"Privacy Shield": Safe-Harbor-Nachfolger bedeutet angeblich EU-Kapitulation*, 2. Feb. 2016. `http://www.heise.de/newsticker/meldung/Privacy-Shield-Safe-Harbor-Nachfolger-bedeutet-angeblich-EU-Kapitulation-3096557.html`, letzter Abruf: 17.7.2016.

[122] Hong, J. I. und J. A. Landay: *An architecture for privacy-sensitive ubiquitous computing*. In: *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, S. 177–189. ACM, 2004.

[123] Horrocks, I., O. Kutz und U. Sattler: *The Even More Irresistible SROIQ*. In: Doherty, P., J. Mylopoulos und C. A. Welty (Hrsg.): *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning*, S. 57–67. AAAI Press, 2006.

[124] Horrocks, I. und P. Patel-Schneider: *Reducing {OWL} entailment to description logic satisfiability*. Web Semantics: Science, Services and Agents on the World Wide Web, 1(4):345 – 357, 2004. International Semantic Web Conference 2003.

[125] Hoseini-Tabatabaei, S. A., A. Gluhak und R. Tafazolli: *A survey on smartphone-based systems for opportunistic user context recognition*. ACM Computing Surveys (CSUR), 45(3):27, 2013.

[126] Hu, H. und J. Xu: *Non-Exposure Location Anonymity*. In: *2009 IEEE 25th International Conference on Data Engineering*, S. 1120–1131, March 2009.

[127] Huo, Z., X. Meng, H. Hu und Y. Huang: *You Can Walk Alone: Trajectory Privacy-preserving Through Significant Stays Protection*. In: *Proceedings of the 17th International Conference on Database Systems for Advanced Applications - Volume Part I*, DASFAA'12, S. 351–366, Berlin, Heidelberg, 2012. Springer-Verlag.

[128] Hutter, D., W. Stephan und M. Ullmann: *Security in Pervasive Computing: First International Conference, Boppard, Germany, March 12-14, 2003. Revised Papers*, Kap. Security and Privacy in Pervasive Computing State of the Art and Future Directions, S. 285–289. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[129] Isaacman, S., R. Becker, R. Cáceres, S. Kobourov, M. Martonosi, J. Rowland und A. Varshavsky: *Identifying Important Places in People's Lives from Cellular Network Data*. In: *Proceedings of the 9th International Conference on Pervasive Computing*, Pervasive'11, S. 133–151, Berlin, Heidelberg, 2011. Springer-Verlag.

[130] Jagtap, P., A. Joshi, T. Finin und L. Zavala: *Preserving Privacy in Context-Aware Systems*. In: *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, S. 149–153, Sept 2011.

[131] Jagtap, P., A. Joshi, T. Finin und L. Zavala: *Privacy Preservation in Context Aware Geosocial Networking Applications*. Techn. Ber., University of Maryland, Baltimore County, May 2011.

[132] Jiang, X., J. I. Hong und J. A. Landay: *Approximate information flows: Socially-based modeling of privacy in ubiquitous computing*. In: *UbiComp*, S. 176–193. Springer, 2002.

[133] Jiang, X. und J. A. Landay: *Modeling privacy control in context-aware systems*. IEEE Pervasive Computing, 1(3):59–63, July 2002.

[134] Jøsang, A., C. Keser und T. Dimitrakos: *Can we manage trust?*. In: *Proceedings of the Third international conference on Trust Management*, S. 93–107. Springer-Verlag, 2005.

[135] Kagal, L., T. Finin und A. Joshi: *A policy language for a pervasive computing environment*. In: *Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on*, S. 63–74. IEEE, 2003.

[136] Kalnis, P., G. Ghinita, K. Mouratidis und D. Papadias: *Preventing Location-Based Identity Inference in Anonymous Spatial Queries*. IEEE Transactions on Knowledge and Data Engineering, 19(12):1719–1733, Dec 2007.

[137] Kang, J. H., W. Welbourne, B. Stewart und G. Borriello: *Extracting Places from Traces of Locations*. In: *Proceedings of the 2Nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, WMASH '04, S. 110–118, New York, NY, USA, 2004. ACM.

[138] Kang, W. und Y. Han: *SmartPDR: Smartphone-Based Pedestrian Dead Reckoning for Indoor Localization*. IEEE Sensors Journal, 15(5):2906–2916, May 2015.

[139] Kanjo, E., L. Al-Husain und A. Chamberlain: *Emotions in Context: Examining Pervasive Affective Sensing Systems, Applications, and Analyses*. Personal Ubiquitous Comput., 19(7):1197–1212, Okt. 2015.

[140] Kannenberg, A.: *Merkel: Daten sind Rohstoffe des 21. Jahrhunderts*. heise online, 2. Nov. 2015. `http://www.heise.de/newsticker/meldung/Merkel-Daten-sind-Rohstoffe-des-21-Jahrhunderts-2867735.html`, letzter Abruf: 17.7.2016.

[141] Kantz, H. und T. Schreiber: *Nonlinear time series analysis*, Bd. 7. Cambridge university press, 2004.

[142] Khalil, A. und K. Connelly: *Context-aware Telephony: Privacy Preferences and Sharing Patterns*. In: *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work*, CSCW '06, S. 469–478, New York, NY, USA, 2006. ACM.

[143] Kiani, S. L., M. Knappmeyery, N. Baker und B. Moltchanov: *A Federated Broker Architecture for Large Scale Context Dissemination*. In: *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, S. 2964–2969, June 2010.

[144] Kido, H., Y. Yanagisawa und T. Satoh: *An anonymous communication technique using dummies for location-based services*. In: *ICPS '05. Proceedings. International Conference on Pervasive Services, 2005.*, S. 88–97, July 2005.

[145] Kofod-Petersen, A., E. Klæboe, J. Jervidalo, K. Aaltvedt, M. Romnes und T. M. Nyhus: *Implementing privacy as symmetry in location-aware systems*. In: *Proceedings of the International Workshop on Combining Context with Trust, Privacy and Security (CAT 2008)*, Bd. 371, S. 1–10, 2008.

[146] Korpipää, P. und J. Mäntyjärvi: *An Ontology for Mobile Device Sensor-based Context Awareness*. In: *Proceedings of the 4th International and Interdisciplinary Conference on Modeling and Using Context*, CONTEXT'03, S. 451–458, Berlin, Heidelberg, 2003. Springer-Verlag.

[147] Korpipää, P., J. Mäntyjärvi, J. Kela, H. Keranen und E. J. Malm: *Managing context information in mobile devices*. IEEE Pervasive Computing, 2(3):42–51, July 2003.

[148] Krumm, J.: *Inference Attacks on Location Tracks*. In: *Proceedings of the 5th International Conference on Pervasive Computing*, PERVASIVE'07, S. 127–143, Berlin, Heidelberg, 2007. Springer-Verlag.

[149] Krumm, J.: *Pervasive Computing: 7th International Conference, Pervasive 2009, Nara, Japan, May 11-14, 2009. Proceedings*, Kap. Realistic Driving Trips For Location Privacy, S. 25–41. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[150] Krumm, J.: *A Survey of Computational Location Privacy*. Personal Ubiquitous Comput., 13(6):391–399, Aug. 2009.

[151] Krummenacher, R. und T. Strang: *Ontology-based context modeling*. In: *Proceedings Third workshop on Context-Aware Proactive Systems (CAPS)*, 2007.

[152] Ku, W.-S., Y. Chen und R. Zimmermann: *Privacy Protected Spatial Query Processing for Advanced Location Based Services*. Wireless Personal Communications, 51(1):53–65, 2009.

[153] Kuneva, M.: *Keynote Speech: Roundtable on Online Data Collection, Targeting and Profiling*. European Commission, 31. März 2009. `http://europa.eu/rapid/press-release_SPEECH-09-156_en.htm`, letzter Abruf: 17.7.2016.

[154] Kushilevitz, E. und R. Ostrovsky: *Replication is not needed: single database, computationally-private information retrieval*. In: *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*, S. 364–373, Oct 1997.

[155] Kämper, V.: *Die Kanzlerin entdeckt #Neuland*, 6. Juni 2013. `http://www.spiegel.de/netzwelt/netzpolitik/kanzlerin-merkel-nennt-bei-obama-besuch-das-internet-neuland-a-906673.html`, letzter Abruf: 17.7.2016.

[156] Lane, N. D., J. Xie, T. Moscibroda und F. Zhao: *On the Feasibility of User De-anonymization from Shared Mobile Sensor Data*. In: *Proceedings of the Third International Workshop on Sensing Applications on Mobile Phones*, PhoneSense '12, S. 3:1–3:5, New York, NY, USA, 2012. ACM.

[157] Langheinrich, M.: *Ubicomp 2001: Ubiquitous Computing: International Conference Atlanta Georgia, USA, September 30–October 2, 2001 Proceedings*, Kap. Privacy by Design — Principles of Privacy-Aware Ubiquitous Systems, S. 273–291. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[158] Langheinrich, M.: *A privacy awareness system for ubiquitous computing environments*. In: *UbiComp 2002: Ubiquitous Computing*, S. 237–245. Springer, 2002.

[159] Lederer, S., J. Mankoff, A. K. Dey und C. P. Beckmann: *Managing Personal Information Disclosure in Ubiquitous Computing Environments*. Techn. Ber. UCB/CSD-03-1257, EECS Department, University of California, Berkeley, Jul 2003.

[160] Lee, K. C., W.-C. Lee, H. V. Leong und B. Zheng: *Navigational Path Privacy Protection: Navigational Path Privacy Protection*. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, S. 691–700, New York, NY, USA, 2009. ACM.

[161] Lee, K. C. K., W. C. Lee, H. V. Leong und B. Zheng: *OPAQUE: Protecting Path Privacy in Directions Search*. In: *2009 IEEE 25th International Conference on Data Engineering*, S. 1271–1274, March 2009.

[162] Lex, E., O. Pimas, J. Simon und V. Pammer-Schindler: *Mobile and Ubiquitous Systems: Computing, Networking, and Services: 9th International Conference, MobiQuitous 2012, Beijing, China, December 12-14, 2012. Revised Selected Papers*, Kap. Where am I? Using Mobile Sensor Data to Predict a User's Semantic Place with a Random Forest Algorithm, S. 64–75. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[163] Li, N., T. Li und S. Venkatasubramanian: *t-Closeness: Privacy Beyond k-Anonymity and l-Diversity*. In: *2007 IEEE 23rd International Conference on Data Engineering*, S. 106–115, April 2007.

[164] LiKamWa, R., Y. Liu, N. D. Lane und L. Zhong: *MoodScope: Building a Mood Sensor from Smartphone Usage Patterns*. In: *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, S. 389–402, New York, NY, USA, 2013. ACM.

[165] Lipmaa, H.: *Advances in Cryptology - ASIACRYPT 2003: 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, November 30 – December 4, 2003. Proceedings*, Kap. Verifiable Homomorphic Oblivious Transfer and Private Equality Test, S. 416–433. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[166] Liu, H., H. Darabi, P. Banerjee und J. Liu: *Survey of wireless indoor positioning techniques and systems*. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 37(6):1067–1080, 2007.

[167] Lund, M. S., B. Solhaug und K. Stolen: *Evolution in Relation to Risk and Trust Management*. Computer, 43(5):49–55, 2010.

[168] Ma, C. Y. T., D. K. Y. Yau, N. K. Yip und N. S. V. Rao: *Privacy Vulnerability of Published Anonymous Mobility Traces*. IEEE/ACM Transactions on Networking, 21(3):720–733, June 2013.

[169] Ma, H., D. Zhao und P. Yuan: *Opportunities in mobile crowd sensing*. Communications Magazine, IEEE, 52(8):29–35, 2014.

[170] Machanavajjhala, A., D. Kifer, J. Gehrke und M. Venkitasubramaniam: *L-diversity: Privacy Beyond K-anonymity*. ACM Trans. Knowl. Discov. Data, 1(1), März 2007.

[171] Madan, A., K. Farrahi, D. Gatica-Perez und A. S. Pentland: *Pervasive Computing: 9th International Conference, Pervasive 2011, San Francisco, USA, June 12-15, 2011. Proceedings*, Kap. Pervasive Sensing to Model Political Opinions in Face-to-Face Networks, S. 214–231. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[172] Maier, M. und F. Dorfmeister: *Mobile Computing, Applications, and Services: 5th International Conference, MobiCASE 2013, Paris, France, November 7-8, 2013, Revised Selected Papers*, Kap. Fine-Grained Activity Recognition of Pedestrians Travelling by Subway, S. 122–139. Springer International Publishing, Cham, 2014.

[173] Maier, M., L. Schauer und F. Dorfmeister: *ProbeTags: Privacy-preserving proximity detection using Wi-Fi management frames*. In: *Wireless and Mobile Computing, Networking and Communications (WiMob), 2015 IEEE 11th International Conference on*, S. 756–763. IEEE, 2015.

[174] Makris, P., D. N. Skoutas und C. Skianis: *A survey on context-aware mobile and wireless networking: on networking and computing environments' integration*. Communications Surveys & Tutorials, IEEE, 15(1):362–386, 2013.

[175] Mercati, P., V. Hanumaiah, J. Kulkarni, S. Bloch und T. Rosing: *BLAST: Battery Lifetime-constrained Adaptation with Selected Target in Mobile Devices*. In: *Proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MOBIQUITOUS '15, S. 80–89, ICST, Brussels, Belgium, Belgium, 2015. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[176] Michalevsky, Y., A. Schulman, G. A. Veerapandian, D. Boneh und G. Nakibly: *PowerSpy: Location Tracking Using Mobile Device Power Analysis*. In: *24th USENIX Security Symposium (USENIX Security 15)*, S. 785–800, Washington, D.C., Aug. 2015. USENIX Association.

[177] Mokbel, M. F., C.-Y. Chow und W. G. Aref: *The New Casper: Query Processing for Location Services Without Compromising Privacy*. In: *Proceedings of the 32Nd International Conference on Very Large Data Bases*, VLDB '06, S. 763–774. VLDB Endowment, 2006.

[178] Monreale, A., R. Trasarti, D. Pedreschi, C. Renso und V. Bogorny: *C-safety: A Framework for the Anonymization of Semantic Trajectories*. Trans. Data Privacy, 4(2):73–101, Aug. 2011.

[179] Montjoye, Y.-A. de, C. A. Hidalgo, M. Verleysen und V. D. Blondel: *Unique in the Crowd: The privacy bounds of human mobility*. Nature srep., 3, 2013.

[180] Motik, B., R. Shearer und I. Horrocks: *Optimized Reasoning in Description Logics using Hypertableaux*. In: Pfenning, F. (Hrsg.): *Proc. of the 21st Conference on Automated Deduction (CADE-21)*, Bd. 4603 d. Reihe *LNAI*, S. 67–83, Bremen, Germany, July 17–20 2007. Springer.

[181] Mouratidis, K.: *Strong location privacy: A case study on shortest path queries*. In: *Data Engineering Workshops (ICDEW), 2013 IEEE 29th International Conference on*, S. 136–143, April 2013.

[182] Mouratidis, K. und M. L. Yiu: *Anonymous Query Processing in Road Networks*. IEEE Transactions on Knowledge and Data Engineering, 22(1):2–15, Jan 2010.

[183] Mulloni, A., D. Wagner, I. Barakonyi und D. Schmalstieg: *Indoor Positioning and Navigation with Camera Phones*. IEEE Pervasive Computing, 8(2):22–31, April 2009.

[184] Musen, M. A.: *The Protégé project: A look back and a look forward*. AI Matters, 1(4):4–12, 2015.

[185] Myles, G., A. Friday und N. Davies: *Preserving privacy in environments with location-based applications*. IEEE Pervasive Computing, 2(1):56–64, Jan 2003.

[186] Naghizade, E., L. Kulik und E. Tanin: *Protection of Sensitive Trajectory Datasets Through Spatial and Temporal Exchange*. In: *Proceedings of the 26th International Conference on Scientific and Statistical Database Management*, SSDBM '14, S. 40:1–40:4, New York, NY, USA, 2014. ACM.

[187] Narayanan, A., N. Thiagarajan, M. Lakhani, M. Hamburg und D. Boneh: *Location Privacy via Private Proximity Testing.*. In: *NDSS*, 2011.

[188] Niemelä, J. und J. Borkowski: *Topology planning considerations for capacity and location technique in WCDMA radio networks*. In: *Proc. of EUNICE Conf*, S. 1–8, 2004.

[189] Niu, B., Q. Li, X. Zhu, G. Cao und H. Li: *Achieving k-anonymity in privacy-aware location-based services*. In: *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, S. 754–762, April 2014.

[190] Ouchi, K. und M. Doi: *Indoor-outdoor Activity Recognition by a Smartphone*. In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, S. 537–537, New York, NY, USA, 2012. ACM.

[191] PARK, J.-S. und S.-J. OH: *A New Concave Hull Algorithm and Concaveness Measure for n-dimensional Datasets*. Journal of information science and engineering, 29(2):379–392, 2013.

[192] PEDDINTI, S. T. und N. SAXENA: *On the Limitations of Query Obfuscation Techniques for Location Privacy*. In: *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11, S. 187–196, New York, NY, USA, 2011. ACM.

[193] PERERA, C., A. ZASLAVSKY, P. CHRISTEN und D. GEORGAKOPOULOS: *Context Aware Computing for The Internet of Things: A Survey*. IEEE Communications Surveys Tutorials, 16(1):414–454, First 2014.

[194] PERTTUNEN, M., J. RIEKKI und O. LASSILA: *Context representation and reasoning in pervasive computing: a review*. International Journal of Multimedia and Ubiquitous Engineering, S. 1–28, 2009.

[195] PFITZMANN, A. und M. KÖHNTOPP: *Anonymity, Unobservability, and Pseudeonymity — a Proposal for Terminology*. In: *International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*, S. 1–9, New York, NY, USA, 2001. Springer-Verlag New York, Inc.

[196] PRATAMA, A. R., WIDYAWAN und R. HIDAYAT: *Smartphone-based Pedestrian Dead Reckoning as an indoor positioning system*. In: *System Engineering and Technology (ICSET), 2012 International Conference on*, S. 1–6, Sept 2012.

[197] PROTSCHKY, V. und S. FEIT: *Traffic Light Assistance – Ein innovativer Mobilitätsdienst im Fahrzeug*. In: LINNHOFF-POPIEN, C., M. ZADDACH und A. GRAHL (Hrsg.): *Marktplätze im Umbruch - Digitale Strategien fuer Services im Mobilen Internet*, Xpert.press. Springer Berlin Heidelberg, 2014.

[198] PUTTASWAMY, K. P. N. und B. Y. ZHAO: *Preserving Privacy in Location-based Mobile Social Applications*. In: *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*, HotMobile '10, S. 1–6, New York, NY, USA, 2010. ACM.

[199] REICHLE, R., M. WAGNER, M. U. KHAN, K. GEIHS, J. LORENZO, M. VALLA, C. FRA, N. PASPALLIS und G. A. PAPADOPOULOS: *A comprehensive context modeling framework for pervasive computing systems*. In: *Distributed applications and interoperable systems*, S. 281–295. Springer, 2008.

[200] REST, J. VAN, D. BOONSTRA, M. EVERTS, M. VAN RIJN und R. VAN PAASSEN: *Privacy Technologies and Policy: First Annual Privacy Fo-*

rum, APF 2012, Limassol, Cyprus, October 10-11, 2012, Revised Selected Papers, Kap. Designing Privacy-by-Design, S. 55–72. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[201] Rodriguez, J. J., L. I. Kuncheva und C. J. Alonso: *Rotation Forest: A New Classifier Ensemble Method*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(10):1619–1630, Oct 2006.

[202] Rogaway, P.: *The Moral Character of Cryptographic Work*. Cryptology ePrint Archive, Report 2015/1162, 2015. Online verfügbar unter `http://eprint.iacr.org/`.

[203] Roussaki, I., M. Strimpakou, N. Kalatzis, M. Anagnostou und C. Pils: *Hybrid context modeling: A location-based scheme using ontologies*. In: *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, S. 6–pp. IEEE, 2006.

[204] RP Online: *Weitere Satelliten im All: Start erster Galileo-Dienste rückt näher*, 25. Mai 2016. `http://www.rp-online.de/panorama/wissen/weltraum/eu-navigation-start-erster-galileo-dienste-rueckt-naeher-aid-1.6000007`, letzter Abruf: 17.7.2016.

[205] Ruppel, P., G. Treu, A. Küpper und C. Linnhoff-Popien: *Anonymous User Tracking for Location-based Community Services*. In: *Proceedings of the Second International Conference on Location- and Context-Awareness*, LoCA'06, S. 116–133, Berlin, Heidelberg, 2006. Springer-Verlag.

[206] Sacramento, V., M. Endler und F. N. Nascimento: *A Privacy Service for Context-aware Mobile Computing*. In: *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, S. 182–193, Sept 2005.

[207] Sacramento, V., M. Endler, H. K. Rubinsztejn, L. S. Lima, K. Goncalves, F. N. Nascimento und G. A. Bueno: *MoCA: A Middleware for Developing Collaborative Applications for Mobile Users*. IEEE Distributed Systems Online, 5(10):2–2, Oct 2004.

[208] Samet, H.: *The Design and Analysis of Spatial Data Structures*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.

[209] Sanders, P. und D. Schultes: *Engineering Highway Hierarchies*. In: *Proceedings of the 14th Conference on Annual European Symposium - Volume 14*, ESA'06, S. 804–816, London, UK, UK, 2006. Springer-Verlag.

[210] Scellato, S., M. Musolesi, C. Mascolo, V. Latora und A. T. Campbell: *NextPlace: A Spatio-temporal Prediction Framework for Pervasive Systems*. In: *Proceedings of the 9th International Conference on Pervasive Computing*, Pervasive'11, S. 152–169, Berlin, Heidelberg, 2011. Springer-Verlag.

[211] Schaar, P.: *Von der Informationsgesellschaft, der Privatsphäre und der Ünsexiness"des Datenschutzes*. heise online, 20. Mai 2016. http://www.heise.de/newsticker/meldung/Von-der-Informationsgesellschaft-der-Privatsphaere-und-der-Unsexiness-des-Datenschutzes-3206830.html, letzter Abruf: 17.7.2016.

[212] Schaub, F., B. Könings und M. Weber: *Context-Adaptive Privacy: Leveraging Context Awareness to Support Privacy Decision Making*. IEEE Pervasive Computing, 14(1):34–43, Jan 2015.

[213] Schmid, W.: *Berechnung kürzester Wege in Straßennetzen mit Wegeverboten*. Doktorarbeit, Universität Stuttgart, 2000.

[214] Schmidt, A., M. Beigl und H.-W. Gellersen: *There is more to context than location*. Computers & Graphics, 23(6):893 – 901, 1999.

[215] Schubert, A.: *Gebäude mit eigenen Postleitzahlen*. N-TV, 2. Juli 2013. http://www.n-tv.de/ratgeber/Sendungen/Gebaeude-mit-eigenen-Postleitzahlen-article10897911.html, letzter Abruf: 17.7.2016.

[216] Shankar, P., V. Ganapathy und L. Iftode: *Privately Querying Location-based Services with SybilQuery*. In: *Proceedings of the 11th International Conference on Ubiquitous Computing*, UbiComp '09, S. 31–40, New York, NY, USA, 2009. ACM.

[217] Shebaro, B., O. Oluwatimi und E. Bertino: *Context-Based Access Control Systems for Mobile Devices*. IEEE Transactions on Dependable and Secure Computing, 12(2):150–163, March 2015.

[218] Sheikh, K., M. Wegdam und M. v. Sinderen: *Quality-of-context and its use for protecting privacy in context aware systems*. Journal of Software, 3(3):83–93, 2008.

[219] Shokri, R., G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux und J.-Y. Le Boudec: *Protecting Location Privacy: Optimal Strategy Against Localization Attacks*. In: *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, S. 617–627, New York, NY, USA, 2012. ACM.

[220] Shokri, R., C. Troncoso, C. Diaz, J. Freudiger und J.-P. Hubaux: *Unraveling an Old Cloak: K-anonymity for Location Privacy*. In: *Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society*, WPES '10, S. 115–118, New York, NY, USA, 2010. ACM.

[221] Šikšnys, L., J. R. Thomsen, S. Šaltenis, M. L. Yiu und O. Andersen: *Advances in Spatial and Temporal Databases: 11th International Symposium, SSTD 2009 Aalborg, Denmark, July 8-10, 2009 Proceedings*, Kap. A Location Privacy Aware Friend Locator, S. 405–410. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[222] Šikšnys, L., J. R. Thomsen, S. Šaltenis und M. L. Yiu: *Private and Flexible Proximity Detection in Mobile Social Networks*. In: *2010 Eleventh International Conference on Mobile Data Management*, S. 75–84, May 2010.

[223] Solanas, A., J. Domingo-Ferrer und A. Martínez-Ballesté: *Location privacy in location-based services: Beyond TTP-based schemes*. In: *Proceedings of the 1st International Workshop on Privacy in Location-Based Applications (PILBA)*, S. 12–23, 2008.

[224] Song, C., Z. Qu, N. Blumm und A.-L. Barabási: *Limits of Predictability in Human Mobility*. Science, 327(5968):1018–1021, 2010.

[225] Strang, T.: *Service-Interoperabilität in Ubiquitous Computing Umgebungen*. Doktorarbeit, LMU München, 2004.

[226] Strang, T. und C. Linnhoff-Popien: *A Context Modeling Survey*. In: *In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004-The Sixth International Conference on Ubiquitous Computing, Nottingham/England*, 2004.

[227] Strang, T., C. Linnhoff-Popien und K. Frank: *Distributed Applications and Interoperable Systems: 4th IFIP WG6.1 International Conference, DAIS 2003, Paris, France, November 17-21, 2003. Proceedings*, Kap. CoOL: A Context Ontology Language to Enable Contextual Interoperability, S. 236–247. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[228] Strassner, J., S. van der Meer, D. O'Sullivan und S. Dobson: *The Use of Context-Aware Policies and Ontologies to Facilitate Business-Aware Network Management*. Journal of Network and Systems Management, 17(3):255–284, 2009.

[229] Sweeney, L.: *K-anonymity: A Model for Protecting Privacy*. Int. J. Uncertain. Fuzziness Knowl.-Based Syst., 10(5):557–570, Okt. 2002.

[230] Süddeutsche Zeitung: *NSA-Überwachung von Apps: Angry Birds in Überwachungsmission*, 27. Jan. 2014. http://www.sueddeutsche.de/digital/apps-im-fokus-von-nsa-und-gchq-angry-birds-in-ueberwachungsmission-1.1873548, letzter Abruf: 17.7.2016.

[231] Terrovitis, M.: *Privacy Preservation in the Dissemination of Location Data*. SIGKDD Explor. Newsl., 13(1):6–18, Aug. 2011.

[232] Terrovitis, M. und N. Mamoulis: *Privacy Preservation in the Publication of Trajectories*. In: *Proceedings of the The Ninth International Conference on Mobile Data Management*, MDM '08, S. 65–72, Washington, DC, USA, 2008. IEEE Computer Society.

[233] Thiagarajan, A., L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo und J. Eriksson: *VTrack: Accurate, Energy-aware Road Traffic Delay Estimation Using Mobile Phones*. In: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, S. 85–98, New York, NY, USA, 2009. ACM.

[234] Thompson, D.: *Google's CEO: 'The Laws Are Written by Lobbyists'*. The Atlantic, 1. Okt. 2010. http://www.theatlantic.com/technology/archive/2010/10/googles-ceo-the-laws-are-written-by-lobbyists/63908/, letzter Abruf: 17.7.2016.

[235] Turow, J., M. Hennessy und N. Draper: *The Tradeoff Fallacy: How Marketers Are Misrepresenting American Consumers and Opening Them Up to Exploitation*. Techn. Ber., Annenberg School for Communication, University of Pennsylvania, Philadelphia, PA, USA, Juni 2015.

[236] Vicente, C. R., I. Assent und C. S. Jensen: *Effective Privacy-Preserving Online Route Planning*. In: *2011 IEEE 12th International Conference on Mobile Data Management*, Bd. 1, S. 119–128, June 2011.

[237] Voigtmann, C. und K. David: *A Survey To Location-Based Context Prediction*. In: Springer (Hrsg.): *AwareCast 2012 (Pervasive)*, 2012.

[238] W3C OWL Working Group: *OWL Web Ontology Language Overview*. W3C Recommendation, 10 February 2004. Online verfügbar unter https://www.w3.org/TR/owl-features/.

[239] W3C OWL Working Group: *OWL Web Ontology Language Reference*. W3C Recommendation, 10 February 2004. Online verfügbar unter https://www.w3.org/TR/owl-ref/.

[240] W3C OWL Working Group: *OWL 2 Web Ontology Language*. W3C Recommendation, 11 December 2012. Online verfügbar unter https://www.w3.org/TR/owl2-overview/.

[241] W3C RDF Core Working Group: *RDF/XML Syntax Specification*. W3C Recommendation, 10 February 2004. Online verfügbar unter `https://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/`.

[242] Wang, T. und L. Liu: *Privacy-aware Mobile Services over Road Networks*. Proc. VLDB Endow., 2(1):1042–1053, Aug. 2009.

[243] Wang, T. D., B. Parsia und J. Hendler: *The Semantic Web - ISWC 2006: 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006. Proceedings*, Kap. A Survey of the Web Ontology Landscape, S. 682–694. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[244] Wang, X. H., D. Q. Zhang, T. Gu und H. K. Pung: *Ontology Based Context Modeling and Reasoning Using OWL*. In: *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, PERCOMW '04, S. 18–22, Washington, DC, USA, 2004. IEEE Computer Society.

[245] Want, R., B. N. Schilit, N. I. Adams, R. Gold, K. Petersen, D. Goldberg, J. R. Ellis und M. Weiser: *An overview of the PARCTAB ubiquitous computing experiment*. IEEE Personal Communications, 2(6):28–43, Dec 1995.

[246] Warren, S. D. und L. D. Brandeis: *The Right to Privacy*. Harward Law Review, 4(5):193–220, December 1890.

[247] Wassermann, E. F.: *Navigieren mit Satellit: GPS*. Welt der Physik, 11. Nov. 2011. `http://www.weltderphysik.de/gebiet/planeten/erde/gps/`, letzter Abruf: 17.7.2016.

[248] Weiser, M.: *The computer for the 21st century*. Scientific american, 265(3):94–104, 1991.

[249] Werner, M.: *Security and Privacy in Mobile Information and Communication Systems: Second International ICST Conference, MobiSec 2010, Catania, Sicily, Italy, May 27-28, 2010, Revised Selected Papers*, Kap. A Privacy-Enabled Architecture for Location-Based Services, S. 80–90. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[250] Werner, M., F. Dorfmeister und M. Schönfeld: *AMBIENCE: A Context-Centric Online Social Network*. In: *WPNC 2015, 12th Workshop on Positioning, Navigation and Communications*, 2015.

[251] Wernke, M., P. Skvortsov, F. Dürr und K. Rothermel: *A Classification of Location Privacy Attacks and Approaches*. Personal Ubiquitous Comput., 18(1):163–175, Jan. 2014.

[252] Wiesner, K.: *Datenerfassung und Privatsphäre in partizipativen Sensornetzen*. Doktorarbeit, Ludwig-Maximilians-Universität München, Mai 2015.

[253] Wiesner, K., S. Feld, F. Dorfmeister und C. Linnhoff-Popien: *Right to silence: Establishing map-based Silent Zones for participatory sensing*. In: *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, S. 1–6, April 2014.

[254] Wishart, R., K. Henricksen und J. Indulska: *Location- and Context-Awareness: First International Workshop, LoCA 2005, Oberpfaffenhofen, Germany, May 12-13, 2005. Proceedings*, Kap. Context Obfuscation for Privacy via Ontological Descriptions, S. 276–288. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[255] Wishart, R., K. Henricksen und J. Indulska: *Ubiquitous Intelligence and Computing: 4th International Conference, UIC 2007, Hong Kong, China, July 11-13, 2007. Proceedings*, Kap. Context Privacy and Obfuscation Supported by Dynamic Context Source Discovery and Processing in a Context Management System, S. 929–940. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[256] Xie, J., B. P. Knijnenburg und H. Jin: *Location Sharing Privacy Preference: Analysis and Personalized Recommendation*. In: *Proceedings of the 19th International Conference on Intelligent User Interfaces*, IUI '14, S. 189–198, New York, NY, USA, 2014. ACM.

[257] Xu, T. und Y. Cai: *Feeling-based location privacy protection for location-based services*. In: *Proceedings of the 16th ACM conference on Computer and communications security*, S. 348–357. ACM, 2009.

[258] Xue, M., P. Kalnis und H. K. Pung: *Location and Context Awareness: 4th International Symposium, LoCA 2009 Tokyo, Japan, May 7-8, 2009 Proceedings*, Kap. Location Diversity: Enhanced Privacy Protection in Location Based Services, S. 70–87. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[259] Yang, J., A. Varshavsky, H. Liu, Y. Chen und M. Gruteser: *Accuracy Characterization of Cell Tower Localization*. In: *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, UbiComp '10, S. 223–226, New York, NY, USA, 2010. ACM.

[260] Yigitoglu, E., M. L. Damiani, O. Abul und C. Silvestri: *Privacy-Preserving Sharing of Sensitive Semantic Locations under Road-Network Constraints*. In: *2012 IEEE 13th International Conference on Mobile Data Management*, S. 186–195, July 2012.

[261] Yin, H., D. Song, M. Egele, C. Kruegel und E. Kirda: *Panorama: Capturing System-wide Information Flow for Malware Detection and Analysis*. In: *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, S. 116–127, New York, NY, USA, 2007. ACM.

[262] Yiu, M. L., C. S. Jensen, J. Møller und H. Lu: *Design and Analysis of a Ranking Approach to Private Location-based Services*. ACM Trans. Database Syst., 36(2):10:1–10:42, Juni 2011.

[263] Zang, H. und J. Bolot: *Anonymization of Location Data Does Not Work: A Large-scale Measurement Study*. In: *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, MobiCom '11, S. 145–156, New York, NY, USA, 2011. ACM.

[264] Zhang, C. und Y. Huang: *Cloaking Locations for Anonymous Location Based Services: A Hybrid Approach*. Geoinformatica, 13(2):159–182, Juni 2009.

[265] Zhao, Y., J. Ye und T. Henderson: *Privacy-aware location privacy preference recommendations*. In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, S. 120–129. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[266] Zheng, Y.: *Tutorial on Location-Based Social Networks*. In: *WWW '12: Proceedings of the 21st International Conference on World Wide Web*, New York, NY, USA, May 2012. ACM.

[267] Zhong, G., I. Goldberg und U. Hengartner: *Privacy Enhancing Technologies: 7th International Symposium, PET 2007 Ottawa, Canada, June 20-22, 2007 Revised Selected Papers*, Kap. Louis, Lester and Pierre: Three Protocols for Location Privacy, S. 62–76. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[268] Zhou, C., N. Bhatnagar, S. Shekhar und L. Terveen: *Mining Personally Important Places from GPS Tracks*. In: *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, S. 517–526, April 2007.

[269] Zhou, C., D. Frankowski, P. Ludford, S. Shekhar und L. Terveen: *Discovering Personally Meaningful Places: An Interactive Clustering Approach*. ACM Trans. Inf. Syst., 25(3), Juli 2007.