Dezentrales Cloud-Netzwerk

Konzeption und Implementierung eines datenschutzgerechten dezentralen Informationssystems mit weit verteilten Datenbeständen

Von der Wirtschaftswissenschaftlichen Fakultät
der Universität Leipzig
genehmigte

DISSERTATION

zur Erlangung des akademischen Grades
DOKTOR-INGENIEUR
DR.-ING.
vorgelegt

von M.Sc. Wirt.-Inf. André Müller
geboren am 02.12.1986 in Leipzig

Gutachter: Prof. Dr. Johannes Ruhland
Prof. Dr. Susanne Strahringer

Tag der Verleihung: 07.07.2021

Bibliographische Beschreibung

Müller, André

Dezentrales Cloud-Netzwerk: Konzeption und Implementierung eines datenschutzgerechten dezentralen Informationssystems mit weit verteilten Datenbeständen

Universität Leipzig, Dissertation
160 (206) S., 126 Lit., 63 Abb., 41 Tab., 6 Listings, 6 Anlagen

Referat:
Datenschutz gewinnt zunehmend an Bedeutung bei der Entwicklung und Bereitstellung moderner IT-Lösungen. Neben zukünftigen technischen Herausforderungen ergeben sich Fragestellungen wirtschaftlicher, rechtlicher, politischer und gesellschaftlicher Natur. Die Wirtschaftsinformatik hat hierbei die Aufgabe verschiedene Teilbereiche zu verbinden und Lösungen zu generieren, welche das Problem der informationellen Selbstbestimmung gezielt adressiert, sowohl für private Endnutzer als auch für KMUs. In diesem Zusammenhang bieten dezentrale Systeme eine Möglichkeit den Datenschutz gezielt zu steigern, da die Nutzer eine höhere Verantwortung innerhalb des gesamten Ökosystems übernehmen müssen. Gleichzeitig steigt dadurch aber die Komplexität sowohl der Systeme als auch der Anwendbarkeit für den Nutzer. Die vorliegende Dissertation beschäftigt sich mit der Gestaltung eines dezentralen Informationssystems, welches den Schutz und die Kontrolle von Datenbeständen ermöglicht und in seiner Komplexität beherrschbar bleibt. Dies beinhaltet die Betrachtung von Dezentralisierung aus technischer und organisatorischer Perspektive. Weiterhin ist die Erstellung von Anforderungen in verschiedene Kategorien eingeteilt. Dies ermöglicht die Konstruktion eines konzeptionellen Architekturmodells. Das Ergebnis der Bearbeitung sind vier Artefakte: ein Modell der organisatorischen Dezentralisierung, ein Anforderungskatalog, ein Architekturmodell und ein erweitertes Architekturmodell. Das Architekturmodell folgt der Grundidee die Daten der Nutzer in deren eigenen Speichersystemen (Storage Clouds) zu hinterlegen und innerhalb des Systems nur Daten für die Administration zu speichern. Der gesamtheitliche Lösungsansatz besteht aus einer externen Sicht (Umwelt/Ökosystem) und einer internen Sicht für das System als solches. Die externe Sicht beinhaltet die Speicher-Clouds und die Endgeräte der Nutzer. Die interne Sicht basiert auf dem SOA-Paradigma, abgewandelt von einer Serviceorientierung hin zu einer Datenorientierung. Für die Evaluation wurde ein Szenario ausgewählt und anschließend eine softwaretechnische Implementierung in Form eines Prototyps vorgenommen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listingverzeichnis

Abkürzungsverzeichnis

ACM	Association for Computing Machinery
API	Application Programming Interface
BDSG	Bundesdatenschutzgesetz
BISE	Business & Information Systems Engineering
BMBF	Bundesministerium für Bildung und Forschung
CO	Concept (Konzept)
DAV	Distributed Authoring and Versioning
DCN	Decentral Cloud Network
DIN	Deutsches Institut für Normung
DPDP	Dynamic Provable Data Possession
DS	Data and Security (Datenschutz)
DT	Dezentrale Technologien
DWS	Data-Warehouse-Systeme
ECIS	European Conference on Information Systems
EJIS	European Journal of Information Systems
EN	Europäische Normen
EU	Europäische Union
EuGH	Europäischer Gerichtshof
FF	Forschungsfrage
GG	Grundgesetz
GUI	Graphical User Interface (deutsch: Grafische Benutzerschnittstelle)
HCI	Human-Computer Interaction
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HuaaS	Human as a Service
IaaS	Infrastructure-as-a-Service
ICIS	International Conference on Information Systems
ID	Identifikationsnummern
IoT	Internet of Things
IP	Internetprotokoll
IS	Informationssystem
ISJ	Information Systems Journal
ISO	Internationale Organisation für Normung
ISR	Information Systems Research
IT	Informationstechnik
ITWM	Institut für Techno- und Wirtschaftsmathematik

IuK	Informations- und Kommunikationstechnik
JACM	Journal of the ACM
JAIS	Journal of the Association for Information Systems
JMIS	Journal of Management Information Systems
JSON	JavaScript Object Notation
JSP	Java Server Pages
KMU	Kleine und mittlere Unternehmen
LaaS	Library as a Service
LAN	Local Area Networks
MAN	Metropolitan Area Networks
MAS	Multi-Agentensysteme
MIS	Management Information Systems
MISQ	Management Information Systems Quarterly
Mrd	Milliarden
NAS	Network Attached Storage
OS	Operating System
P2P	Peer-to-Peer
PaaS	Platform-as-a-Service
PN	Personal Network
QoS	Quality of Service
RAM	Random-Access Memory
REST	Representational State Transfer
RM	Rechtemanagement
SaaS	Software-as-a-Service
SLA	Service Level Agreements
SNS	Social Network Service
SOA	Serviceorientierte Architektur
SOAP	Simple Object Access Protocol
SSH	Secure Shell
StaaS	Storage as a Service
TKG	Telekommunikationsgesetz
TR	Vertrauen (Trust)
TTIP	Transatlantisches Freihandelsabkommen
TÜV	Technischer Überwachungsverein
UFF	Unterforschungsfrage
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
US	United States

USA	United States of America
VHB	Verband der Hochschullehrer für Betriebswirtschaft
VIS	Virtual Individual Server
VM	Virtuelle Maschine
WAN	Wide Area Networks
WAR	Web Application Archive
WBS	Wissensbasierte Systeme
WFMS	Workflow-Management-Systeme
WI	Wirtschaftsinformatik
WSDL	Web Services Description Language
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol

1 Einführung

In einer stetig komplexer werdenden Welt ergeben sich zunehmend gesellschaftliche, wirtschaftliche und politische Fragestellungen, deren Beantwortung einer umfassenden Auseinandersetzung mit spezifischen Themengebieten bedarf. Eines der grundlegenden Aspekte der heutigen Zeit ist der gerade stattfindende Wandel Europas von einer Dienstleistungsgesellschaft hin zu einer Informationsgesellschaft. Im Mittelpunkt steht hierbei die Digitalisierung mit all ihren Facetten und Ausprägungen in den Bereichen Innovation, Integration und Inklusion. Das Internet und die Vernetzung von Maschinen jeglicher Art ermöglichen eine Transformation klassischer Strukturen und zeigen sich unter anderem in den Sektoren Web 2.0, Industrie 4.0 und smarte Anwendungen. Informationstechnologie sollte innerhalb dieses Prozesses stets als Ermöglicher und nicht als Treiber fungieren und die Bedürfnisse all jener zufriedenstellen, die mit bereits bestehenden Lösungen Probleme und Aufgaben nicht bestmöglich bewältigen konnten.

Mit dem Einsatz neuer technischer Möglichkeiten, die es erlauben eine unvorstellbar große Menge an Daten zu sammeln und zu verarbeiten, ergibt sich eines der bedeutendsten Spannungsfelder der gegenwärtigen Zeit: Digitalisierung und Datenschutz. Nach Roßnagel [2012, 331 ff.] befinden wir uns bezogen auf die Datenverarbeitung auf der dritten Entwicklungsstufe der Digitalisierung. Davon ausgehend ist es notwendig, neue Schutzkonzepte für die informationelle Selbstbestimmung in den Bereichen Transparenz, Zweckbindung, Erforderlichkeit und bei der Wahrnehmung von Betroffenenrechten gegenüber neuen Technikanwendungen zu entwickeln. Hierfür ist es unter anderem geboten, den Kunden als Anwender stärker in den Fokus der Betrachtung zu rücken. Die vorliegende Dissertation leistet einen Beitrag dazu, offene Fragestellungen zu beantworten und technische, infrastrukturelle und organisatorische Lösungsstrategien zu erarbeiten. Dazu wird in dieser wissenschaftlichen Abhandlung ein Architekturmodell vorgestellt, welches durch Analysen gewonnene Anforderungen an ein solches Konzept erfüllt.

Der Abschnitt Einführung gibt in Kapitel 1.1 einen umfangreichen Einblick zur Motivation und Problemstellung des Themas. Daraus wird in Kapitel 1.2 eine Forschungsfrage abgeleitet. Für die Beantwortung der Frage erfolgt in Kapitel 1.3 die Vorstellung der verwendeten Forschungsmethodik. Abschließend wird eine Übersicht zum Aufbau der Dissertation vorgestellt (Kapitel 1.4).

1.1 Motivation und Problemstellung

Die Motivation hat das Ziel, die Relevanz der Thematik der vorliegenden Dissertation aufzuzeigen, und darzustellen, welche Bedeutung eine forschungsbezogene Bearbeitung besitzt. Hierbei beziehen sich die Ausführungen der Publikation auf das Memorandum der Wirtschaftsinformatik (siehe [Österle et al. 2010]). Betrachtet wird die wissenschaftliche, wirtschaftliche und gesellschaftlich-politische Motivation. Aufbauend auf diesen wird die Problemstellung identifiziert, welche den Gegenstand der wissenschaftlichen Abhandlung darstellt. Anschließend erfolgt eine Einordnung und Abgrenzung der Beteiligten und Anwendergruppen, die in dieser Dissertation adressiert werden. Dies schließt ebenfalls die Darstellung der jeweiligen Interessen ein, welche die Adressaten an einer Problemlösung aufweisen. Nachfolgend werden zunächst die verschiedenen Aspekte der Motivation näher erläutert.

1.1.1 Wissenschaftliche Aspekte

Die wissenschaftliche Motivation zeigt Themenbereiche auf, die Bestandteil des zu bearbeitenden Themengebietes sind. Hierbei werden Schwerpunkte gewählt, welche sich noch in einem frühen Bearbeitungsstadium befinden. Zu nennen sind hierbei das Fog Computing und der daraus abgeleitete Storage Fog. Ausgehend von einer zunehmenden Dezentralisierung und wachsenden Datenmengen, bei gleichzeitigen Übertragungsproblemen, entstand das Feld Fog Computing. Hierbei handelt es sich um riesige Mengen an heterogenen, dezentralen, überall vorkommenden Endgeräten für Speicher- und Bearbeitungsaufgaben ohne Beteiligung Dritter, die kommunizieren und kooperieren (vgl. [Vaquero/Rodero-Merino 2014, 30]). Vaquero und Rodero-Merino [2014, 30] stellten fest, dass Fog Computing durch die Dezentralisierung einen positiven Einfluss auf den Datenschutz (engl.: *Privacy*) hat. Einhergehend mit diesem neuen Ansatz ergeben sich sieben Herausforderungen, die es zukünftig zu lösen gilt (vgl. [Vaquero/Rodero-Merino 2014, 31]):

- Auffinden und Synchronisierung
- Berechnungs- und Speicherlimitierung
- Management
- Sicherheit
- Standardisierung
- Monetarisierung
- Programmierbarkeit

Daraus abgeleitet ergibt sich der Bereich des Storage Fog. Dieser kann als eine Weiterentwicklung zu der Storage Cloud gesehen werden. Im Sinne des Dezentralisierungsgedankens behalten Unternehmen eine eigene IT für die Speicherung ihrer Unternehmensdaten

(Private Cloud) und Ressourcen für eine eventuelle Kooperation mit anderen Marktteilnehmern. Schlussendlich wird Software dort ausgeführt, wo sie benötigt wird und nicht zentral bei einem Anbieter.

1.1.2 Wirtschaftliche Aspekte

Aus wirtschaftlicher Sicht ist das Marktvolumen für Lösungen dieser Art von Bedeutung. Die Schätzung im privaten Endkonsumentenbereich ist sehr aufwändig. Bei Unternehmen wie Facebook und Google zeigt sich dennoch ein hohes Volumen. Aus deutscher Sicht ist es zielführend, die Anzahl und Marktmacht von KMU zu betrachten. Günterberg [2012] zeigt statistisch auf, wie hoch die Anzahl an KMU in Deutschland ist und welchen anteiligen Umsatz sie generieren:

- 99,7 % aller Unternehmen in Deutschland sind KMU
  - Daten für 2009 - [Günterberg 2012, 3]
- 39,1 % Gesamtumsatz in Deutschland (1.948 Mrd. €)
  - Daten für 2009 - [Günterberg 2012, 3]

Somit ist die wirtschaftliche Relevanz gegeben, sich dieser Thematik anzunehmen. Werden Lösungen entwickelt, welche einfach in der Handhabung sind und den Datenschutz einhalten, kann ein großer Teil der deutschen Wirtschaft davon profitieren und sich zukünftigen digitalen Herausforderungen stellen.

1.1.3 Politische und gesellschaftliche Aspekte

Die Politik sowie Privatpersonen sehen sich zunehmend in einer Abhängigkeitsposition gegenüber großen Unternehmen. Damit diese aufgelöst werden kann, benötigt es innovative Lösungsansätze. Weiterhin machen die technische Entwicklung und das Anwachsen an Datenmengen ein Umdenken im Bereich Management von Daten notwendig. Nachfolgend wird diese Thematik näher beleuchtet, da sie einen direkten Einfluss auf den Trend zur Dezentralisierung hat. Wurden im Jahr 2005 130 Exabyte an Daten erzeugt, so ist 2020 mit 40.026 Exabyte erzeugter Daten zu rechnen (vgl. Abbildung 1). Dies zeigt einen rasanten Anstieg und eine gleichzeitig steigende Digitalisierung der Unternehmenslandschaft. Die Technologie für die Übertragung der Daten kann nicht im gleichen Maße mitwachsen, was an der Anzahl an gesendeten Daten zu erkennen ist (vgl. Abbildung 2). Diese steigt von 718,2 Exabyte im Jahr 2014 auf voraussichtlich 2015,64 Exabyte im Jahr 2019 an.

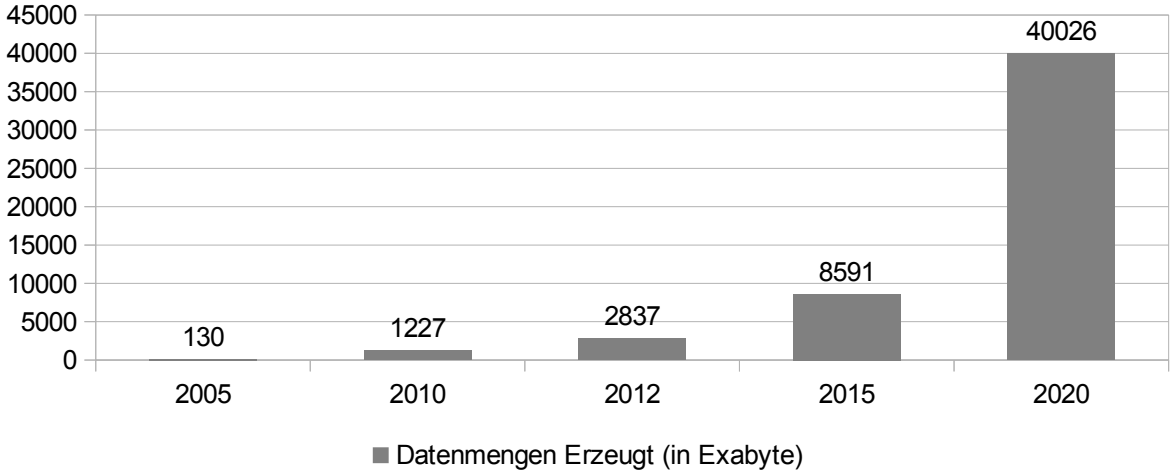


Abbildung 1: Weltweit erzeugte Datenmengen [Statista 2012]

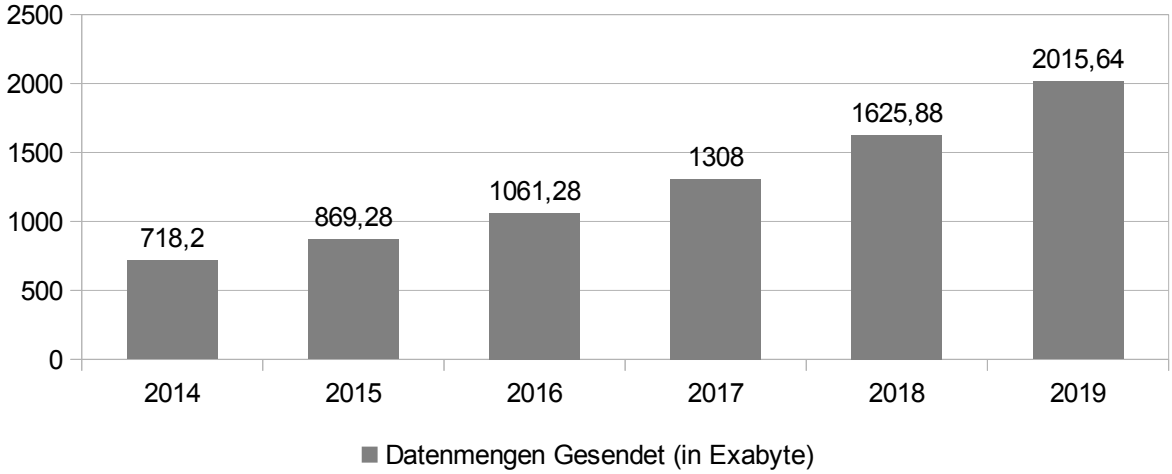


Abbildung 2: Weltweit gesendete Datenmengen (vgl. [Statista 2015])

Werden die erzeugten Datenmengen mit den gesendeten verglichen, wird ein zunehmendes Auseinanderdriften deutlich (vgl. Abbildung 3). Dies kann und wird sich zukünftig als ein Problem herausstellen. Zentralisierung von Daten ist nicht mehr in vollem Umfang möglich. Gerade im Bezug auf Big Data müssen Lösungen gefunden werden, dieses Problem zu adressieren. Im Sinne der Dezentralisierung haben sich Ansätze entwickelt, welche sogenannte Data Lakes oder Data Spaces für die Speicherung von Daten verwenden. Dadurch ist eine Teil- bzw. Voranalyse der Daten möglich, ohne dem Zwang der Zentralisierung der Daten ausgesetzt zu sein.

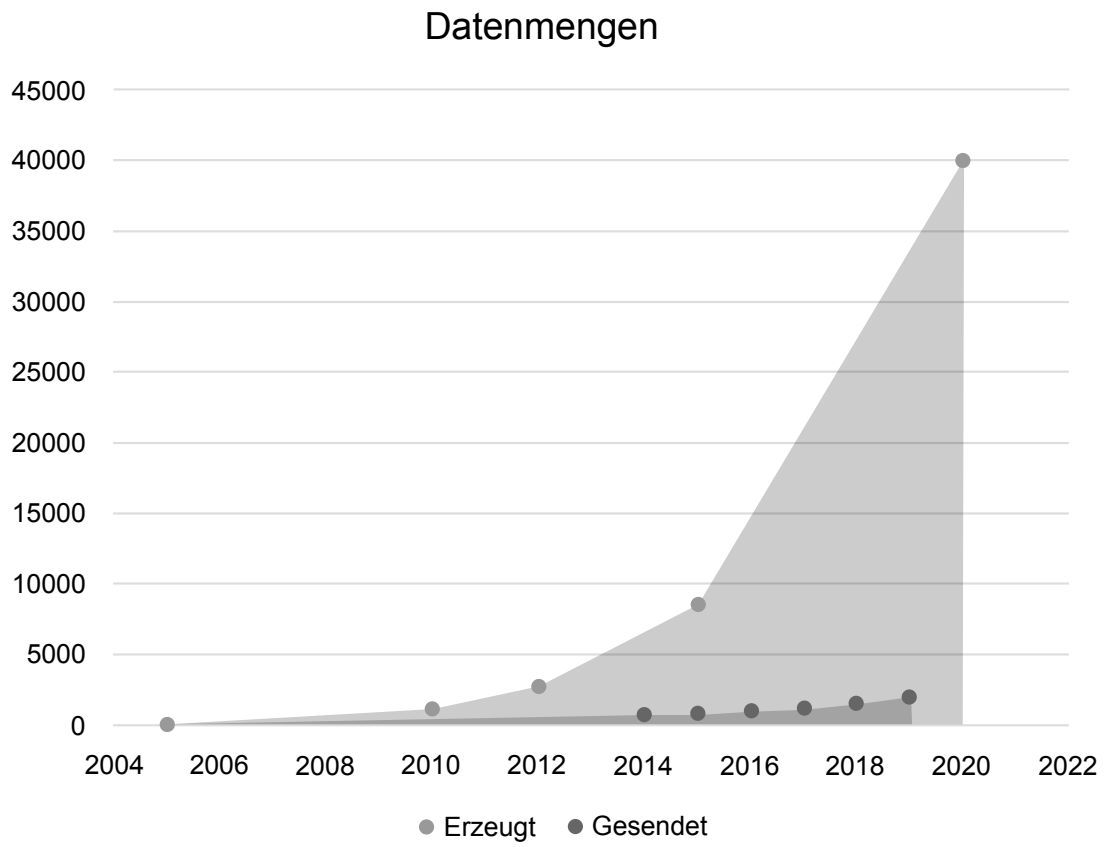


Abbildung 3: Differenz zwischen erzeugten und gesendeten Datenmengen

Neben technischen Einflussgrößen haben auch politische Faktoren eine Relevanz für gesellschaftliche Aspekte. Dies wird unter anderem an der neuen Hightech-Strategie Deutschlands sowie am Cyber Security Report 2015 sichtbar. *Die neue Hightech-Strategie Innovationen für Deutschland* [2014] steht für eine neue Innovationspolitik. Sie setzt sich zum Ziel, „Deutschland auf dem Weg zum weltweiten Innovationsführer voranzubringen" [BMBF 2014, 10]. Weiterhin, „Deutschlands Position als führende Wirtschafts- und Exportnation [zu stärken]" [BMBF 2014, 3]. Dies kann gelingen, wenn neue Ansätze eine Konkurrenz zu etablierten Unternehmen darstellen. Doch das wird aus Sicht der durch den Cyber Security Report Befragten eher kritisch gesehen. In dieser Befragung wurden Topentscheider aus Politik und Wirtschaft zu relevanten Themen der Digitalisierung befragt. Die Notwendigkeit zum Aufbau von Konkurrenzunternehmen zu Google, Facebook oder Apple halten 66 % für wichtig. Dennoch wird dies von zwei Dritteln der Befragten für wenig realistisch gehalten. (vgl. [IfD et al. 2015, 12f]) Dies wäre neben dem Auflösen von partiellen Monopolen auch eine Chance für die Steigerung des Datenschutzes. Es handelt sich somit um rechtliche (Datenschutz), technische und organisatorische Gesichtspunkte.

Diese müssen disziplinübergreifend gelöst werden. Die Frage ist, wie ein solches Informationssystem konzipiert werden muss, um die identifizierten Anforderungen zu lösen.
Nachfolgend wird aus den Aspekten der Motivation jeweils eine Anwendergruppe herausgenommen, welche in dieser Arbeit adressiert wird.

1.1.4 Beteiligte und Anwendergruppen

Die Motivation für die Beschäftigung mit dem Themenbereich dezentrale Informationssysteme basiert grundsätzlich auf drei Anwendergruppen: Forschung, kleine und mittlere Unternehmen (KMU) und Privatpersonen. Nachfolgend werden die drei Gruppen detailliert vorgestellt und deren derzeitige Situation näher erläutert.
Die Forschung, die gerade in der Wirtschaftsinformatik eine starke Verknüpfung mit der Wirtschaft besitzt, sieht sich oft der Problematik ausgesetzt, gewonnene Daten mit anderen Wissenschaftlern zu teilen. Dabei muss sensibel mit den Daten der Partner sowie mit Bestandteilen der Forschungsergebnisse, z. B. entwickelte Algorithmen, umgegangen werden. Aus datenschutzrechtlichen Gründen und der Wahrung von Geschäftsgeheimnissen ist dies oft nicht möglich. Dies widerstrebt einer möglichen Überprüfung der Ergebnisse und einer eventuellen Falsifizierung. Neue Lösungsansätze könnten helfen, Datenbestände zumindest zu einem gewissen Teil bereitzustellen.
Bei KMU steht die Bewahrung der eigenen Daten als wirtschaftlicher Faktor im Mittelpunkt. KMU sehen sich zunehmenden Herausforderungen bezüglich der modernen Datenverwaltung ausgesetzt. Technologischer sowie monetärer Veränderungszwang üben einen enormen Druck auf Firmen aus. Ähnlich verhält sich dies bei Privatpersonen, die durch einen gesellschaftlichen Druck, zur Veränderung des Verhaltens bezüglich der Freigabe persönlicher Daten gezwungen werden. Bei beiden Beteiligungsgruppen besteht das Problem darin, dass das Risiko des Verlustes der Datenhoheit allgegenwärtig ist und durch die zunehmende Komplexität der Anwendungen noch steigt. Daher ist das primäre Ziel die Erreichung eines höherern Schutzes der persönlichen und wirtschaftsbezogenen Daten. KMU sind eine große wirtschaftlich relevante Gruppe in Deutschland. Da diese sich zumeist keine hochqualifizierte IT-Abteilung leisten können oder auf Eigenlösungen setzen müssen, entstehen hohe Kosten. Eine mögliche Lösung ist der Einsatz moderner Technologien, wie etwa das Cloud Computing. Hierbei wird ein Teil oder die gesamte technologische Infrastruktur ausgelagert und von einem Dienstleister betrieben. Trotz Vorteilen im Bereich Kosten und Verwaltung weigern sich viele Unternehmen diesen Schritt zu gehen. Eine Um-

frage von TecChannel [2014] zeigte, warum sich deutsche KMU nicht für den Einsatz von Cloud-Diensten entscheiden. So ist das wichtigste Argument die Datensicherheit und der Datenschutz (64,17 %). Gefolgt von dem Kontrollverlust über den Speicherort der ausgelagerten Daten (45,83 %). Eine weitere Erkenntnis betraf die Wichtigkeit der Speicherung der Daten auf Servern in Deutschland. Dies kann mit insgesamt 81,19 % als entscheidend eingestuft werden. Abbildung 4 zeigt die genauen Ergebnisse nochmals grafisch.

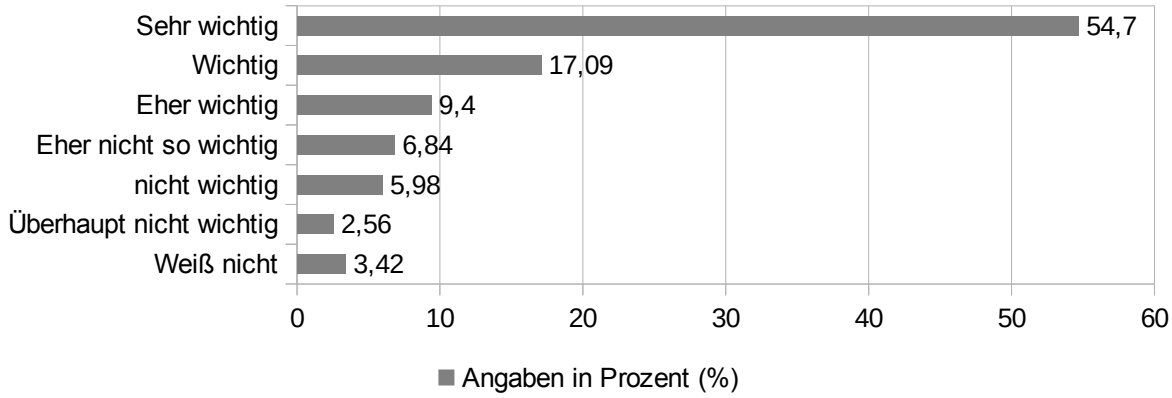


Abbildung 4: Wie wichtig ist die Speicherung der Daten ausschließlich in Deutschland? [TecChannel 2014]

Zusammenfassend kann festgestellt werden, dass die Datensicherheit und der Datenschutz von Dienstleistern garantiert, aber nicht gewährleistet werden kann. Wo die Daten schlussendlich wirklich liegen und wer auf diese Zugriff hat, kann nicht mit Sicherheit gesagt werden. Hierbei ist der Aspekt der Industriespionage ein bedeutendes Hemmnis. Somit ist es nicht verwunderlich, dass sich Ansätze etabliert haben, eine eigene Cloud innerhalb des Unternehmens zu betreiben. Zu nennen sind hierbei ownCloud, Seafile, Pydio, aber auch Systeme wie Protonet und NAS. Dies führt auf lange Sicht zu einer massiven dezentralen Datenhaltung. Ziel muss es sein, Unternehmen unter Einbezug ihrer jeweiligen Bedenken an die Cloud-Technologie heranzuführen.

Privatpersonen nutzen angebotene Dienstleistungen im Internet, um für sich einen Mehrwert zu generieren. Meist ist die Teilnahme bzw. Verwendung mit der Preisgabe persönlicher Daten verbunden. Hierbei ist zu beachten, dass Unternehmen nicht zwangsläufig einen Mehrwert durch diese Daten erhalten. So ist das Geschäftsmodell von Onlinehändlern nicht ursächlich auf das Sammeln von Daten über ihre Kunden ausgelegt. Technologisch sind diese dennoch dazu gezwungen, Daten ihrer Kunden zu erhalten, um den Kaufprozess durchführen zu können. Die Verwaltung dieser Daten stellt aus datenschutzrechtlicher Sicht eine kostenintensive Aufgabe dar. Wenn diese abgegeben werden kann, hätten beide

Seiten, Kunden und Unternehmen, einen Vorteil. Beachtet werden muss hierbei, dass große Onlinehändler versuchen, ihren Gewinn zu erhöhen, indem sie als Reaktion auf das Kaufverhalten ihrer Kunden gezielte Angebote schalten. Dies wird von Kunden meist nicht negativ wahrgenommen. Ein typisches Anwendungsszenario des Datenschutzes sind soziale Netzwerke im Internet. Diese generieren aus den Daten ihrer Kunden Einnahmen und stehen gerade deswegen zunehmend in der Kritik. Ziel muss es sein, dass Privatpersonen die Vergabe und die Zugriffsverweigerung ihrer Daten zentral steuern können.

1.1.5 Resümee

Zusammenfassend ist festzuhalten, dass das aufgezeigte Themenfeld im Bereich Wissenschaft eine hohe Relevanz besitzt. Aus wirtschaftlicher Sicht verspricht die Beteiligung an der Lösungsgenerierung eine zukünftige Rentabilität. Die Erreichung von neuen Lösungsstrategien ist richtungsweisend für Staat und Gesellschaft. Nachfolgend wird aus der aufgezeigten Motivation und Problemstellung die Forschungsfrage dieser Arbeit abgeleitet.

1.2 Forschungsfrage

Für die Bearbeitung und Lösung der aufgezeigten Problemstellungen wird eine Forschungsfrage mit vier Unterforschungsfragen gebildet. Dies dient der Strukturierung, sowie Konkretisierung der Forschung. Die wichtigsten Aspekte betreffen den Datenschutz, die Dezentralisierung und die Erreichung einer geringen Komplexität des Gesamtsystems. Als Ergebnis dieser Arbeit wird ein prototypisches Informationssystem entworfen, welches Anforderungen aus Wissenschaft und Praxis erfüllt. Nachfolgend sind die Forschungsfrage und die vier daraus abgeleiteten Unterforschungsfragen aufgelistet.

FF Wie muss ein Informationssystem (IS) gestaltet sein, welches den Schutz und die Kontrolle von Datenbeständen in zentralen Anwendungssystemen mit dezentraler Datenhaltung ermöglicht und welches in seiner Komplexität beherrschbar bleibt?

UFF 1 Welche Arten der Dezentralisierung gibt es in einem IS auf organisatorischer und technischer Ebene?

UFF 2 Welche Anforderungen ergeben sich in einem zentralen Anwendungssystem mit dezentraler Datenhaltung im Bereich Datenschutz und -kontrolle?

UFF 3 Wie können bei der Gestaltung eines Informationssystems die Anforderungen hinsichtlich Datenschutz und Komplexität in einem dezentralen Anwendungssystem berücksichtigt werden?

UFF 4 Welche Softwaretechniken eignen sich, um den Anforderungen in den Bereichen Sicherheit und Performanz gerecht zu werden?

Für die Beantwortung aller Fragen wird im nächsten Abschnitt die eingesetzte Forschungsmethodik festgelegt und umfassend erläutert.

1.3 Forschungsmethodik

Die Forschungsmethodik dieser Arbeit basiert auf dem Rahmenwerk Design Science nach Hevner et al. [2004]. Hierbei handelt es sich um ein Forschungsparadigma, welches der Entwicklung von innovativen Artefakten zur Lösung von relevanten Problemen aus Wissenschaft und Praxis dient. Der Problemlösungsprozess teilt sich ausgehend von Hevner et al. [2004, 78] in einen Entwurfsprozess und in einen Evaluationsprozess (engl.: *build and evaluate process*) auf. Der Entwurfsprozess dient der Entwicklung von Artefakten, um ungelöste Probleme zu adressieren. Der Evaluationsprozess bewertet anschließend die Nützlichkeit der Artefakte. Das wissenschaftliche Vorgehen dieser Arbeit basiert auf einer einführenden theoretischen Grundlage, welche als Fundament für die zu entwickelnden Artefakte dient. Hierbei von besonderer Bedeutung sind Theorien aus der Forschung und Praxis in den Themenbereichen dezentrale Informationssysteme, Cloud und Fog Computing sowie Datenschutz.

Artefakt-Typen, die aus der wissenschaftlichen Bearbeitung eines Themengebietes resultieren können, sind: Konstrukte, Modelle, Methoden und Instanzen. Konstrukte werden in Form von Vokabular und Symbolen dargestellt. Modelle sind Abstraktionen sowie Repräsentationen der Realität. Methoden werden in Form von Algorithmen und Praktiken abgebildet. Instanzen sind die Realisierung von Implementierungen und Prototypen. Das Design Science Framework wird in dieser Arbeit im Sinne eines Referenzmodells konfiguriert, damit die Zielstellung zur Durchführung der wissenschaftlichen Bearbeitung der Fragestellungen erfüllt werden kann. Abbildung 5 zeigt die konkrete Ausprägung des Rahmenwerkes. Es ist zu erkennen, dass die wirtschaftliche Umgebung in einem großen Umfang zur Bearbeitung des Themengebietes beiträgt. Grundlagen aus der Wissenschaft werden aus Konstruktionen, Modellen und Instanzen gewonnen. Für die Beantwortung der Fragestellungen werden Artefakte erzeugt, welche experimentell und analytisch evaluiert werden. Eine ausführliche Auswertung der Ergebnisse, die einen Teil des Evaluationsprozesses darstellt, schließt die wissenschaftliche Bearbeitung ab.

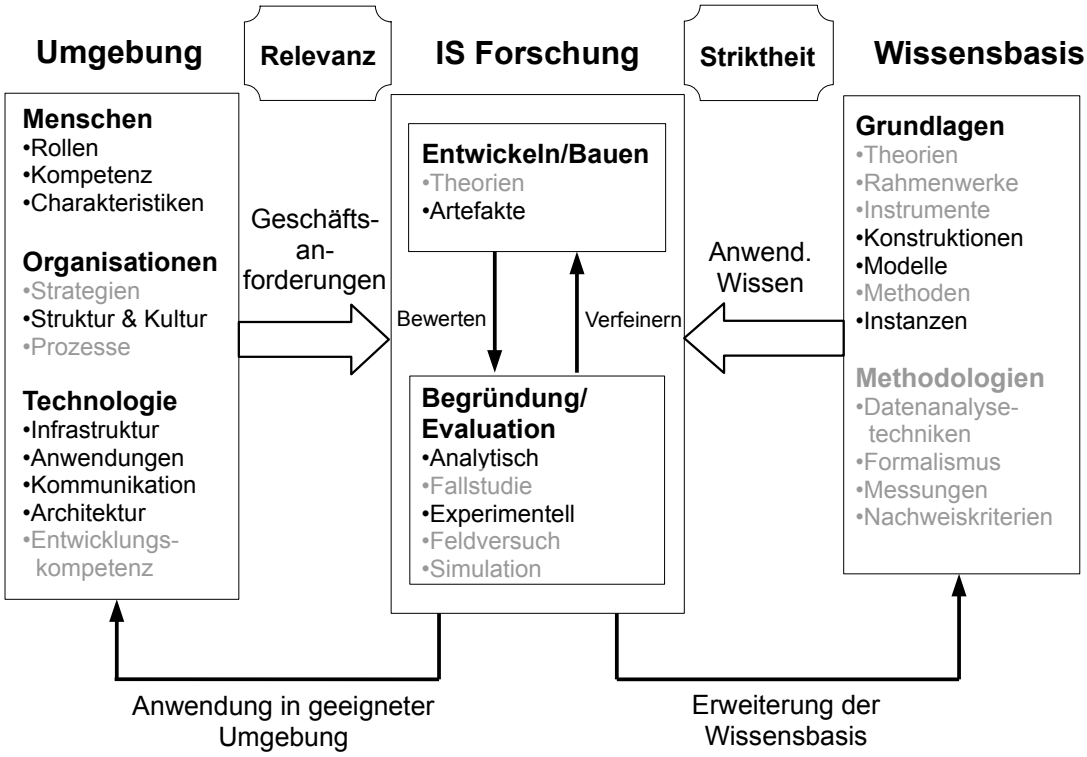


Abbildung 5: Ausprägung des Design Science Frameworks (vgl. [Hevner et al. 2004, 80])

Für eine erfolgreiche Durchführung eines Forschungsvorhabens haben Hevner et al. [2004] sieben Forschungsrichtlinien aufgestellt. In Tabelle 1 sind diese Richtlinien mit einer Beschreibung aufgelistet. Richtlinie 1 wird erfüllt durch die Konstruktion eines Architekturmodells, welches so abstrakt gestaltet wird, dass es für viele Anwendungsbereiche verwendbar ist. Die Motivation und Problemstellung zeigt die Problemrelevanz des Themas auf (R2). Die Evaluation in Form eines Prototypen zeigt Nutzen, Qualität und Wirksamkeit in Form eines Demonstrators. Richtlinie 4 ist Teil des Diffusionsprozesses und wird neben dem Veröffentlichen von Publikationen in der Fachwelt ebenfalls durch diese Dissertation erfüllt. Richtlinie 5 bedient sich wissenschaftlicher Methoden wie zum Beispiel einer systematischen Literaturanalyse. Die Problemumgebung konzentriert sich auf das aktuell sehr relevante Thema Datenschutz (R6). Das Ergebnis ist sowohl relevant für die Wissenschaft als auch für die Praxis (R7). Insgesamt kann so sichergestellt werden, dass alle Richtlinien für die Bearbeitung der Forschungsfrage erfüllt werden.

Design-Science Forschungsrichtlinien	
Richtlinie	Beschreibung
R1: Design als Artefakt	Die Design-Science-Forschung muss ein brauchbares Artefakt erzeugen in Form eines Konstrukts, eines Modells, eines Verfahrens oder einer Instanziierung.
R2: Problemrelevanz	Das Ziel der Design-Science-Forschung ist die Entwicklung technologiebasierter Lösungen für wichtige und relevante Geschäftsprobleme.
R3: Design Evaluation	Nutzen, Qualität und Wirksamkeit eines Design-Artefaktes muss mit gut ausgeführten Evaluationsmethoden rigoros demonstriert werden.
R4: Forschungsbeiträge	Eine effektive Design-Science-Forschung muss klare und überprüfbare Beiträge in den Bereichen Artefakt-Design, Grundlagen-Design und/oder Design-Methoden liefern.
R5: Forschungsstriktheit	Die Design-Science-Forschung beruht auf der Anwendung von rigorosen Methoden, sowohl in der Konstruktion als auch in der Bewertung von Design-Artefakten.
R6: Design als Suchprozess	Die Suche nach einem wirksamen Artefakt erfordert die Verwendung verfügbarer Mittel, um bei gleichzeitiger Erfüllung der Gesetze der Problemumgebung die gewünschten Ergebnisse zu erreichen.
R 7: Forschungskommunikation	Die Design-Science-Forschung muss wirkungsvoll sowohl technologieorientierte als auch managementorientierte Zielgruppen erreichen.

Tabelle 1: Richtlinien der Design-Science-Forschung (vgl. [Hevner et al. 2004, 83])

Für eine erfolgreiche Beantwortung der Forschungsfrage wird der Ablauf ausgehend von [Österle et al. 2010, 4] und den Design Science Research Cycles (vgl. [Hevner 2007, 88] und [vom Brocke/Buddendick 2006, 582]) ausführlich geplant. Der Ablauf teilt sich in vier Phasen: Analyse, Entwurf, Evaluation und Diffusion. Die Analysephase dient der Extraktion von vorhandenem Wissen aus der Forschung und wird mit einer systematischen Literaturanalyse vollzogen. Es werden zwei Modelle als Artefakte erzeugt: Organisatorische Dezentralisierung und Anforderungskatalog. Das Modell zur organisatorischen Dezentralisierung betrachtet die verschiedenen Arten der Interaktion von Gruppen und Individuen unter- und zueinander. Der Anforderungskatalog listet zu erfüllende Eigenschaften eines zu entwickelnden Informationssystems auf. Die Entwurfsphase nutzt die Methode der Konstruktion, um ein Architekturmodell zu erzeugen. Weiterhin wird ein Artefakt der Kategorie Praktik erstellt, welches aufzeigt wie Performanz- und Sicherheitsanforderungen durch aktuelle technische Möglichkeiten positiv beeinflusst werden. Die Evaluation erfolgt mit Hilfe eines Prototypen. In der Diffusionsphase werden die gewonnenen Ergebnisse einem breiten Publikum vorgestellt. Abbildung 6 zeigt das soeben Beschriebene nochmals gra-

fisch.

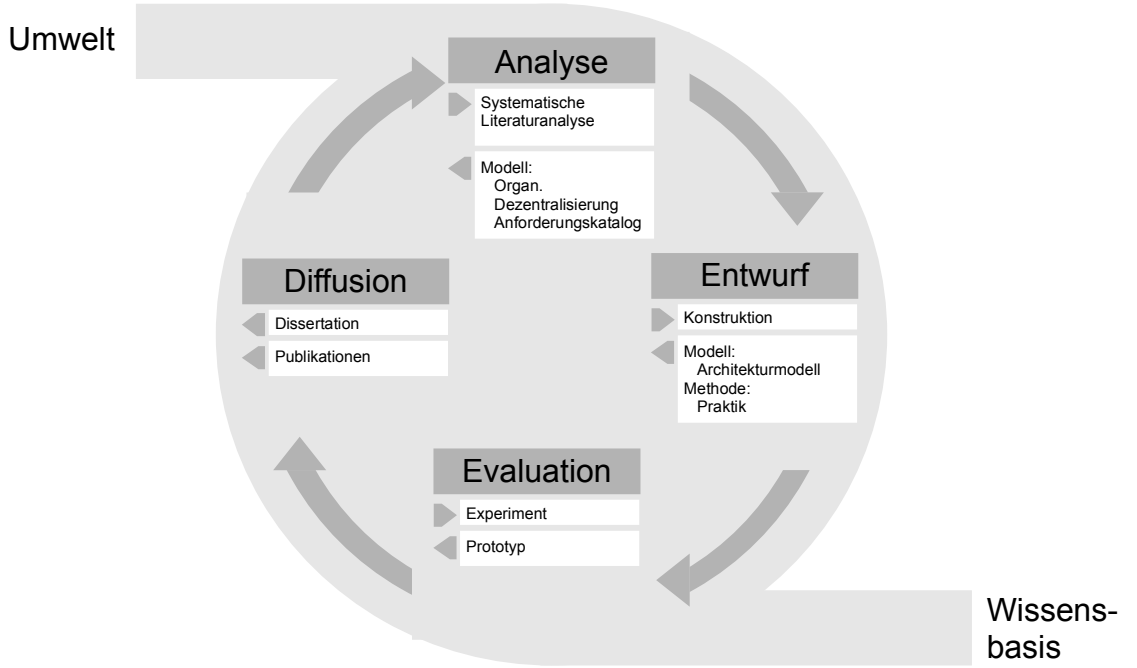


Abbildung 6: Ablauf der wissenschaftlichen Bearbeitung (in Anlehnung an [Österle et al. 2010, 4])

Damit schon im Anfangsstadium der Arbeit der Forschungsrahmen definiert ist, werden im Folgendem tabellarisch verschiedene Spektren der Arbeit festgelegt. Es ist zu bemerken, dass die zu erstellende Arbeit Bezug auf Design Science nimmt und daher eher einen beschreibenden und entwickelnden Charakter und weniger einen analytischen besitzt. Tabelle 2 zeigt die ausgeprägte Forschungsentscheidung.

Spektrum			
Name	Ausprägung	vs.	Ausprägung
Methode	**Qualitativ**		Quantitativ
Ziel	Entdecken		**Beschreiben**
Rahmen	**Fall**		Statistik
Umfeld	Feld		**Labor**
Zeit	**Querschnitt**		In Längsrichtung
Ergebnis	Beschreibung		**Begründung**
Ambition	Verstehen		**Entwerfen**

Tabelle 2: Ausgeprägte Forschungsentscheidung (in Anlehnung an [Recker 2012, 34])

Nachfolgend wird der Aufbau dieser Dissertation festgelegt, welches in einem engen Zusammenhang mit dem soeben vorgestellten Ablaufplan steht.

1.4 Aufbau der Dissertation

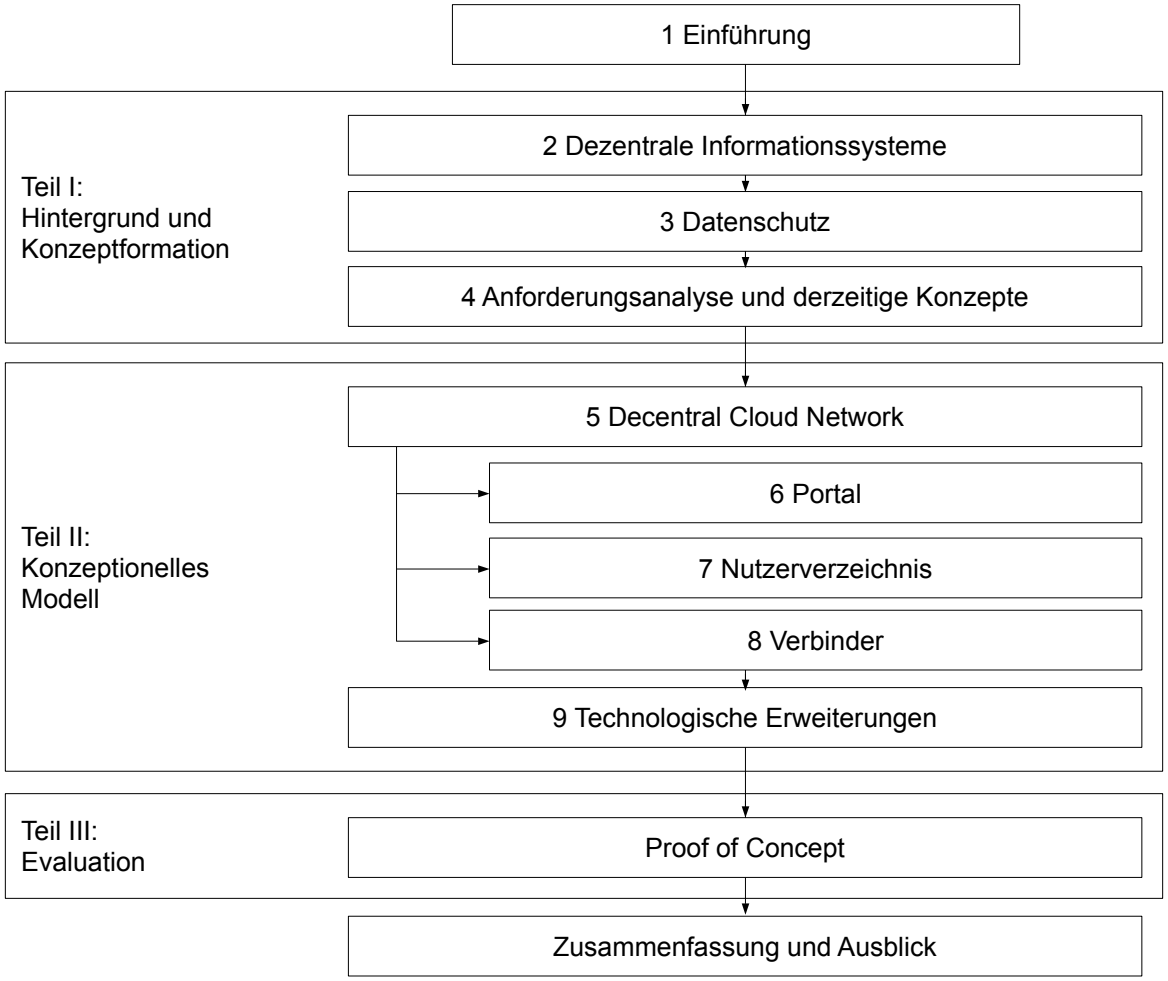


Abbildung 7: Struktur dieser Dissertation

Teil I: Hintergrund und Konzeptformation

Der Hintergrund und die Konzeptformation bilden die einführende Betrachtung des Themengebietes sowie die Analyse bestehender Problemfelder. Sie beinhalten theoretische Grundlagen und den aktuellen Stand der Forschung. Die vorliegende Publikation ordnet sich in eines der grundlegenden Themenfelder der Wirtschaftsinformatik ein: Informations- und Kommunikationssysteme. Adressiert werden hierbei betriebliche Informationssysteme und Kommunikations- und Kollaborationssysteme in Form von Cloud-Ökosystemen. Übergreifend ist die Dezentralisierung zentraler Bestandteil der Betrachtung.
In den theoretischen Grundlagen werden technologische Aspekte in den Bereichen dezentrale Informationssysteme, Cloud/Fog Computing und Emergent Software betrachtet. Das Kapitel Datenschutz erläutert die datenschutzrechtlichen Grundlagen, die aktuelle Situation in Deutschland sowie den Datentransfer zwischen der Europäischen Union und den USA. Das Kapitel Anforderungsanalyse und derzeitige Konzepte beinhaltet eine systematische Literaturanalyse, einen Systemvergleich und die Erstellung eines Anforderungskataloges. Zusammenfassend bildet Teil I dieser Arbeit die Grundlage für die anschließende Konstruktion eines Architekturmodells.

2 Dezentrale Informationssysteme

Dieses Kapitel dient der Darstellung von aktuellen Entwicklungen im Bereich dezentrale Informationssysteme. Der Fokus liegt hierbei auf der Dezentralisierung und auf verteilten Anwendungen. Neue Technologien ermöglichen eine Konzipierung und Umsetzung von Systemen, welche sich an den aktuellen Herausforderungen des Marktes und den Bedürfnissen der Kunden orientieren. Nachfolgend werden Informationssysteme als solche (Kapitel 2.1), das Cloud Computing (Kapitel 2.2), die Emergent Software (Kapitel 2.3) und das Fog Computing (Kapitel 2.4) näher vorgestellt.

2.1 Informationssysteme

Informationssysteme (IS) sind soziotechnische Systeme, welche Informationsnachfragen von Anwendern abdecken. Sie bilden einen elementaren Bestandteil der Wirtschaftsinformatik und sind von besonderer Bedeutung in dieser Arbeit. Zunächst erfolgt eine Definition und Einordnung mit einer anschließenden Typisierung von Informationssystemen. Abschließend wird das Mensch/Aufgabe/Technik-System näher erläutert.

2.1.1 Definition und Einordnung

Nach Laudon et al. [2009, 31] gilt für ein Informationssystem folgende Definition:

„Ein System, das für die Zwecke eines Teils eines bestimmten Unternehmens geschaffen bzw. in diesem Betrieb eingesetzt wird. Ein Informationssystem enthält die dafür notwendige Anwendungssoftware und Daten und ist in die Organisations-, Personal- und Technikstrukturen des Unternehmens eingebettet.“

Neben dem betrieblichen Aufgabengebiet, in Form von notwendiger Anwendungssoftware und Daten, berücksichtigt es zusätzlich die Organisationsstruktur und die Menschen, die mit dem System arbeiten (siehe Abbildung 8) (vgl. [Laudon et al. 2009, 31]). Es handelt sich grundsätzlich um eine betriebsindividuelle Anpassung eines Systems. Es muss individuell entwickelt und angepasst werden, damit es seine volle Wirkung entfalten kann. Dadurch hilft es dem Unternehmen und seinen Mitarbeitern, Probleme zu analysieren, komplizierte Sachverhalte zu durchblicken und neue Produkte zu entwickeln. (vgl. [Laudon et al. 2009, 31/32]) Neben technischen und betriebswirtschaftlichen Aspekten wie der IT-Infrastruktur und der Datenverwaltung, ist der Mensch in seinen Rollen als Mitarbeiter und im Management ein wesentlicher Bestandteil. Ausgehend von einer sich fortwährend än-

dernden digitalen Unternehmenslandschaft und neuen Technologien erfahren IS eine stete Weiterentwicklung. Nachfolgend werden verschiedene Typisierungen von IS vorgenommen.

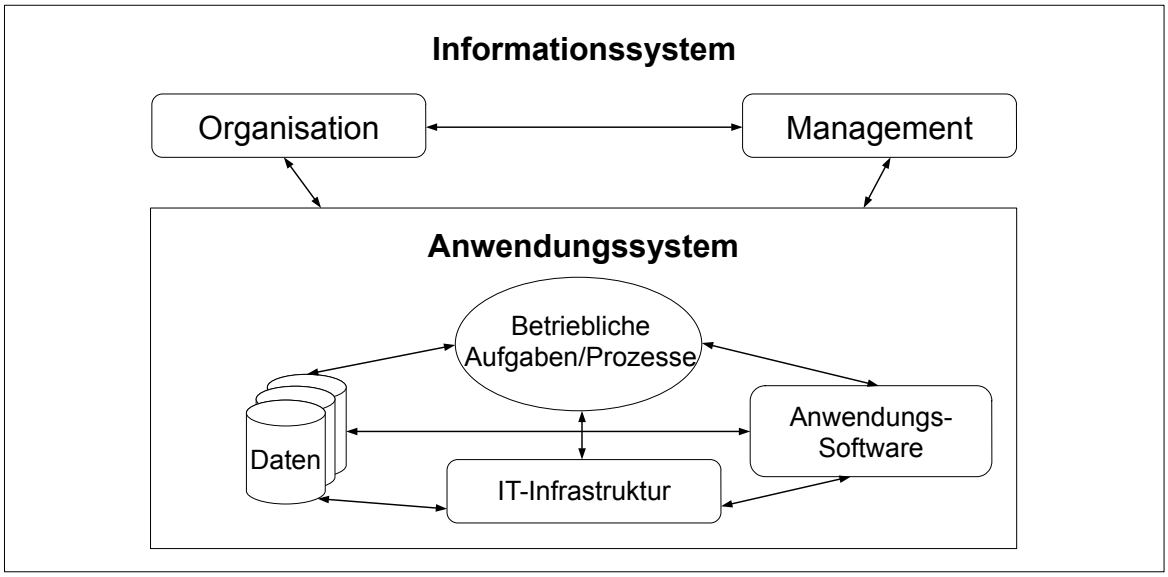


Abbildung 8: Zusammenhang Informationssystem und Anwendungssystem (in Anlehnung an [Laudon et al. 2009, 32])

2.1.2 Typisierung von Informationssystemen

In diesem Abschnitt werden Informationssysteme durch fünf Typisierungsansätze basierend auf Typisierungsmerkmalen charakterisiert: Mensch, Benutzereigenschaften, betriebliche Aufgaben, Technik und Phasen im Informationsverhalten.

Das Typisierungsmerkmal Mensch ist kategorisiert nach Benutzern, Benutzertypen und Benutzereigenschaften. Hierbei können zunächst Endbenutzersysteme und Führungsinformationssysteme unterschieden werden. Endbenutzersysteme ermöglichen es, Aufgabenträgern in Fachabteilungen ohne Hilfe von IT-Spezialisten Programme zu entwickeln und auszuführen. Führungsinformationssysteme dienen der Planung und Steuerung von Unternehmensaufgaben. Sie helfen beim Erkennen von Chancen und Risiken, zeigen Planabweichungen auf und unterstützen durch das Festlegen von Leistungskennzahlen den Vergleich von Ist- und Soll-Zustand. (vgl. [Heinrich et al. 2010, 252f]) Die Entwicklung von IS nach Benutzereigenschaften basiert auf der Unterschiedlichkeit der Nutzer, unter anderem in den Bereichen Alter, Geschlecht, Qualifikation und Erfahrung im Umgang mit IuK-Technik. Ausprägungen in Form von Typ-Beispielen sind Kinder und Jugendliche, Senioren, Gestaltung von Produktangeboten für Frauen anders als für Männer und Systeme für Computer-

Laien. (vgl. [Heinrich et al. 2010, 253]) Hierbei zeigt sich die Stärke von modernen Softwaresystemen, Menschen unterschiedlich zu integrieren, um eine höchstmögliche Befriedigung der Nutzer-Bedürfnisse zu erreichen.

Das Typisierungsmerkmal betriebliche Aufgabe ist analog zu der Typisierung aus der Betriebswirtschaftslehre. Hierbei werden die Bereiche Aufgabenphasen, Aufgabentypen, Aufgabenreichweite und Betriebstypen unterschieden (vgl. [Heinrich et al. 2010, 253]). Somit folgt es dem Prinzip, Softwaresysteme ausgehend von einem Problemraum zu konzipieren, und nicht, durch neue Technologien der Wirtschaft bei der Leistungserbringung Vorgaben zu diktieren.

Das Typisierungsmerkmal Technik unterscheidet sich zu dem letztgenannten, durch eine von IuK-Technik getriebene Herangehensweise. Beispiele für Systeme dieser Art sind: Workflow-Management-Systeme (WFMS), Wissensbasierte Systeme (WBS), Multi-Agentensysteme (MAS) und Data-Warehouse-Systeme (DWS). (vgl. [Heinrich et al. 2010, 256f])

Bei dem Typiserungsmerkmal Phasen im Informationsverhalten steht die Information im Mittelpunkt der Betrachtung. Ein Informationssystem hat die Aufgabe, eine Informationsnachfrage der Nutzer zu decken. Der Benutzer weißt hierfür ein spezielles Informationsverhalten auf. Insgesamt existieren fünf Phasen des Informationsverhaltens:

- Informationswahrnehmung
- Informationssammlung
- Informationsstrukturierung und -organisation
- Informationsproduktion
- Informationspflege

(vgl. [Heinrich et al. 2010, 257f])

2.1.3 Mensch/Aufgabe/Technik-System

Wie bereits in den vorhergehenden Abschnitten beschrieben, ist der Mensch ein wichtiger Bezugspunkt für die Entwicklung von Informationssystemen. Die Strukturelemente Mensch, Aufgabe und Technik stehen in einem engen Zusammenhang. Somit werden die Merkmale der beteiligten Menschen mit den Anforderungen der Aufgaben und den Eigenschaften der eingesetzten IuK-Technik abgeglichen, um ein qualitativ hochwertiges Gesamtsystem zu konzipieren. Sogenannte Task-Technologie-Fit-Modelle, welche die Kongruenz zwischen Aufgabe und Technik betrachten, besitzen den Mangel, den Faktor

Mensch nicht in die Betrachtung einzubeziehen. Daher fehlen bisher Modelle, welche alle drei Elemente behandeln. (vgl. [Heinrich et al. 2010, 257])

Das Ziel muss es sein, den Menschen stärker in den Fokus der Betrachtung zu rücken. Zum Beispiel durch die Integration der Beziehungen von Menschen innerhalb eines Ökosystems und der (Unternehmens-)Umwelt. Gerade bei dezentral organisierten Systemen kommt dem Individuum in Form eines Menschen, eines Unternehmens oder einer Gruppe an Beteiligten ein bedeutender Faktor zu. Im nächsten Kapitel wird Cloud Computing als eine Implementierungsstrategie moderner IS vorgestellt.

2.2 Cloud Computing

Das Cloud Computing ist DIE technologische Revolution der letzten Jahre in der IT-Landschaft. Nach mehreren Jahren im produktiven Einsatz konnte es seinen Mehrwert beweisen und findet neben der Verwendung im Unternehmensumfeld auch den Weg zum privaten Endanwender. Sowohl die Virtualisierung von Infrastruktur Inhouse als auch die Nutzung großer Anbieter, wie etwa Amazon Web Services, Microsoft Azure oder Salesforce, bieten den Anwendern eine steigende Flexibilität, bessere Ressourcennutzung und sinkende Kosten. Bei den Endkonsumenten sind Cloud-Speicher (engl.: *Storage Clouds*) für die externe Sicherung ihrer Daten sehr beliebt. Beispiele sind: DropBox, Apple iCloud, Google Drive und Microsoft OneDrive. Auf die Problematik der Nutzung solcher Dienste, welche von US-amerikanischen Unternehmen angeboten werden, wird im Abschnitt Datenschutz (Kapitel 3) ausführlich eingegangen. In den nachfolgenden Unterkapiteln wird zunächst der Begriff Cloud Computing erläutert und dessen Charakteristiken vorgestellt. Anschließend werden die Organisation und die Architektur sowie die Service-Modelle erläutert.

2.2.1 Begriffsbestimmung und Charakteristiken

Grundsätzlich beschreibt Cloud Computing die Trennung von Programmen, Informationen und der physischen Infrastruktur auf der einen Seite und die Art der Bereitstellung für den Nutzer auf der anderen (vgl. [Vossen et al. 2012, 19]). Somit kann es, sofern ein externer Anbieter involviert ist, als erweiterte „Spielart“ des IT-Outsourcings gesehen werden (vgl. [Vossen et al. 2012, 20]).

In dieser Arbeit wird die Definition von Baun et al. [2009] verwendet, welche sich auf die Begriffsbestimmung des National Institute of Standards and Technology stützt:

„Unter Ausnutzung virtualisierter Rechen- und Speicherressourcen und moderner Web-Technologien stellt Cloud Computing skalierbare, netzwerk-zentrierte, abstrahierte IT-In-

frastrukturen, Plattformen und Anwendungen als on-demand Dienste zur Verfügung. Die Abrechnung dieser Dienste erfolgt nutzungsabhängig.“ [Baun et al. 2009, 4]

Aus dieser Definition lassen sich vier Eigenschaften ableiten, die das Cloud Computing näher beschreiben: ein großer Vorrat auf leicht zugreifbare, virtualisierte Ressourcen, deren Zugriff über das Internet realisiert wird, eine dynamische Anpassung auf veränderte Anforderungen (Skalierung), die Abrechnung der Leistung auf Grundlage der Nutzung (engl.: *Pay-per-Use*) und die Verwendung von Service Level Agreements (SLA) zur Abstimmung der Vertragsparteien. (vgl. [Vossen et al. 2012, 20]) Darüber hinaus besitzt es fünf Charakteristiken, die vom National Institute of Standards and Technology [2011] definiert werden:

Fünf Charakteristiken des Cloud Computings	
Eigenschaft	Beschreibung
Resource Pooling	Gemeinsame Nutzung physischer Ressourcen
Rapid Elasticity	Unverzügliche Anpassbarkeit an aktuellen Ressourcenbedarf
On-Demand Self Service	Selbstbedienung nach Bedarf
Broad Network Access	Umfassender Netzwerkzugriff
Mesured Service	Messung der Servicenutzung

Tabelle 3: Fünf Charakteristiken des Cloud Computings (vgl. [Mell/Grance 2011, 2])

Durch den Einsatz des Cloud Computings ergeben sich vornehmlich zwei Vorteile: Zum einen die Senkung der Kosten und zum anderen die Steigerung der Flexibilität. Die Senkung der Kosten ergibt sich aus der Effizienzsteigerung beim Einsatz und der Erübrigung von bestimmten Aufgaben. Hauptsächlich wird hierbei an Personalkosten gespart. (vgl. [Vossen et al. 2012, 32ff]) Die Steigerung der Flexibilität entsteht durch die Umstellung auf das serviceorientierte Paradigma, die Verbesserung der eigenen IT und die Möglichkeit der flexiblen Reaktion auf schwankende Anforderungen. (vgl. [Vossen et al. 2012, 33] und [Repschläger et al. 2014, 14]) Im nächsten Abschnitt erfolgt die Beschreibung organisatorischer Aspekte sowie die Architektur-Varianz beim Einsatz einer Cloud.

2.2.2 Organisation und Architektur

Die Organisation einer Cloud-Infrastruktur beschreibt die Art der Öffnung beziehungsweise die Zugriffsbeschränkungen nach außen (vgl. [Vossen et al. 2012, 30]). Zu unterscheiden sind hierbei vier Arten des Einsatzes: Public Cloud (deutsch: *öffentliche Cloud*), Private Cloud (deutsch: *nichtöffentliche Cloud*), hybride Cloud und Community-Cloud (vgl. [Mell/Grance 2011, 3]).

Die Public Cloud bietet, gegebenenfalls gegen eine Nutzungsgebühr, einen Service an, den jeder bzw. eine hinreichend große Menge an Personen nutzen kann. Im Gegensatz dazu steht eine Private Cloud nur einer einzigen Organisation zur Verfügung und ist somit auf den Zugriff durch seine Mitglieder beschränkt. Die hybride Cloud beschreibt einen Zusammenschluss mehrerer anderer Clouds, die meist durch proprietäre Standards miteinander verknüpft sind. Eine Community-Cloud ist eine Sonderform der Private Cloud, die sich dadurch auszeichnet, dass sich mehrere Organisationen mit ähnlichen Anforderungen deren Nutzung teilen. (vgl. [Vossen et al. 2012, 30ff])

Die Klassifikation von Cloud Computing-Technologien nach deren Dienstleistungsangebot erfolgt im Cloud Architecture Stack. Dieser beschreibt den Grad an Outsourcing der Infrastruktur. Unterschieden wird hierbei in die folgenden drei Bereiche (vgl. [Mell/Grance 2011, 3]):

- Software-as-a-Service (SaaS)
- Platform-as-a-Service (PaaS)
- Infrastructure-as-a-Service (IaaS)

Abbildung 9 zeigt eine feingranulare Einordnung der verschiedenen Cloud Architekture Stack-Ebenen mit deren jeweiligen virtualisierten Elementen.

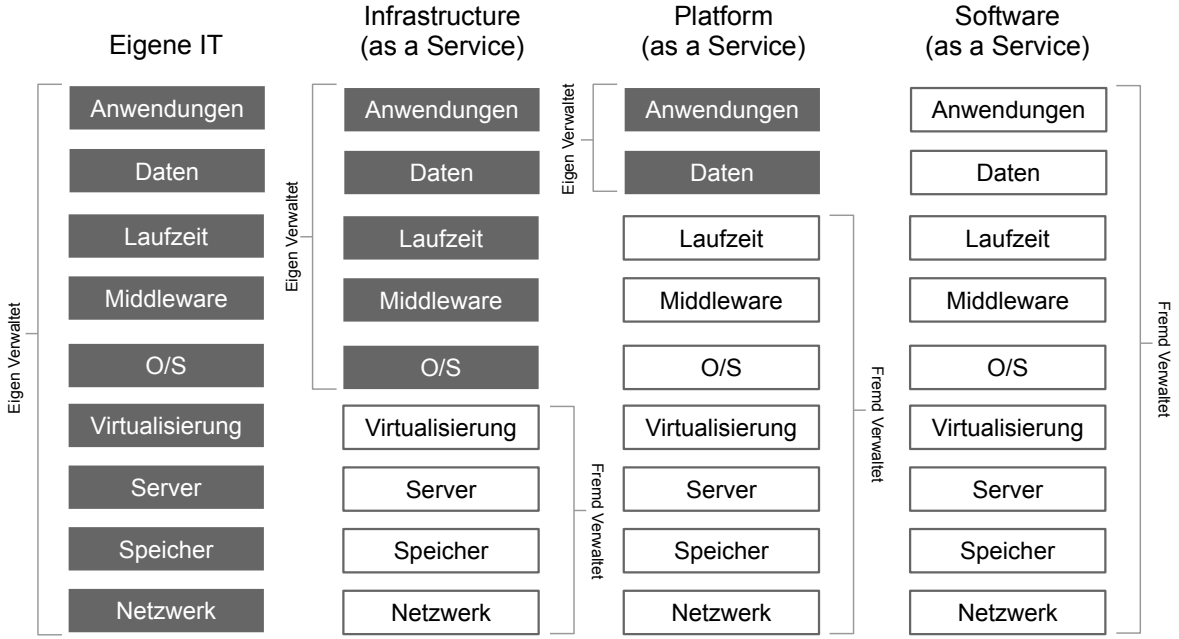


Abbildung 9: Cloud Computing Stack (in Anlehnung an [Kii 2013])

SaaS ist die oberste Ebene des Stacks, in der ein Anbieter seinen Kunden Software anbietet. Er übernimmt die Wartung, Aktualisierung und Fehlerbehebung der Software und ver-

waltet zusätzlich die Lizenzierung für eventuell benötigte zusätzliche Soft- und Hardware. Der Zugriff erfolgt dabei meist über einen Webbrowser. (vgl. [Vossen et al. 2012, 28]) PaaS ist die darauffolgende Ebene, die dem Anwender die Möglichkeit gibt, eigene Programme auf einer bereitgestellten Plattform zu installieren. Dieses Angebot wird typischerweise von Webentwicklern wahrgenommen. Der Provider übernimmt hierbei die Verwaltung der Plattform bis hin zur gesamten Infrastruktur. (vgl. [Vossen et al. 2012, 29]) IaaS beschreibt die dritte Ebene des Stacks, in der der Nutzer alle bisher beschriebenen Schichten oberhalb selbst verwalten muss. Der Provider stellt dabei die virtuelle Hardware oder Infrastrukturdienste zur Verfügung. (vgl. [Vossen et al. 2012, 30])

2.2.3 Erweiterung der Servicearten

Aus den drei Grundformen des Cloud Architecture Stack hat sich eine Vielzahl an Unterformen entwickelt. Diese dienen zum einen der besseren Beschreibung und Abgrenzung der Dienstleistung als auch der besseren Vermarktung. Bespiele der Integration weiterer Aspekte sind: Menschen/Nutzer, Daten oder Hochleistungsinfrastruktur.

Nachfolgend werden drei Spezialarten des As-a-Service-Paradigmas näher erläutert: Human as a Service (HuaaS), Storage as a Service (StaaS) und Library as a Service (LaaS).

Human as a Service erweitert die technische Sicht um Dienstleistungen, die von Menschen erbracht werden. Der Anreiz besteht für diese in kleinen monetären und nicht-monetären Belohnungen und diese bieten dem Cloud-Nutzer den Vorteil, Aufgaben abzugeben, in denen der Mensch dem Rechnersystem überlegen ist.

Das Crowdsourcing[1] ist hierbei die dominierende Unterkategorie. (vgl. [Baun et al. 2009, 37]) Bemerkenswert ist, dass die Crowd nicht nur Arbeitsleistungen zur Verfügung stellt, sondern auch ihre Daten und Informationen. Das Unternehmen, welches die Arbeitskraft einer solchen Crowd nutzen möchte, muss gegebenenfalls eigene Daten preisgeben. So stellt z. B. die Bereitstellung einer Programmierschnittstelle und deren Verwendung eine Offenlegung von Unternehmenseigentum dar welche das Risiko eines Abflusses von Know-how beinhalten kann. (vgl. [Hammon/Hippner 2012, 165ff]) Ein Beispiel für einen realen Einsatz ist die Plattform Amazon Mechanical Turk (https://www.mturk.com). Hierbei werden den Unternehmen gegen Bezahlung menschliche Leistungen bereitgestellt.

Storage as a Service (StaaS), ebenfalls bekannt unter den Namen Cloud Storage und Sto-

1 Crowdsourcing bezeichnet ein Verfahren, in dem Aufgaben, die traditionell innerhalb eines Unternehmens stattfinden, nach außen an eine undefinierte, gewöhnlich große Gruppe von Leuten vergeben werden (vgl. [Howe 2008, 6ff] und [Howe 2010])

rage Cloud, ist ein Ansatz, privaten Endanwendern und Unternehmen virtuellen Festplattenspeicher über das Internet bereitzustellen. Es kann aus organisatorischer Sicht den Bereichen IaaS und PaaS zugeordnet werden (vgl. [Bond 2015]). Die gespeicherten Daten werden zumeist von den Anbietern verwaltet und bieten eine Versionskontrolle. Zu unterscheiden ist hierbei in zwei Dienstleistungsabnehmer: private Endanwender und Unternehmen.

Private Endanwender haben beim Verwenden dieser Technologie einen Mehrwert durch das Bereitstellen und Synchronisieren von Daten über verschiedene Endgeräte hinweg (vgl. [Bond 2015]). Zusätzlich haben sie die Möglichkeit, Daten mit anderen Teilnehmern zu tauschen und gemeinsam zu bearbeiten. Dieser Service wird den Endverbrauchern von einer Vielzahl von Anbietern kostenlos zur Verfügung gestellt. Bezahlt wird für eine Vergrößerung des Speicherplatzes. Die Auswahl einer Storage Cloud nach deutschem Datenschutzrecht kostet tendenziell einen monatlichen Beitrag. Beispiele von Dienstleistungen speziell für den Endkonsumentenbereich sind: DropBox, Apple iCloud, Google Drive, Microsoft OneDrive oder TeamDrive.

Für Unternehmen bietet StaaS eine kostengünstige und flexible Lagerung von Daten, welche sich, ausgehend vom Umfang der Nutzung, erweitert oder verkleinert. Hierbei wird zwischen verschiedenen Arten der Speicherung differenziert: Network Attached Storage (NAS), Block Storage, Object-based Storage und Backup. NAS benötigt für die Speicherung der Daten keinen eigenen Server. Diese Speicherart wird häufig dazu verwendet, unstrukturierte Daten wie Dokumente, Präsentationen und Grafikdateien zu hinterlegen. Block Storage benötigt einen Host-Server oder eine virtuelle Maschine (VM) und ist gleich einem lokalen Festplattenspeicher. Object-based Storage funktioniert ohne einen Server oder VM. Der objektbasierte Speicher verwendet eine spezielle Technik für das Schreiben von Daten und Metadaten. Diese Technologie wird für die Langzeitdatenspeicherung und Archivierung verwendet. Schlussendlich bietet das Backup die Möglichkeit, Unternehmensdaten extern zu verwahren und bei einer Beschädigung wiederherzustellen. (vgl. [Bond 2015])

Neben den vielen Vorteilen dieser neu organisierten Speicherverwaltung existieret auch eine Reihe von Nachteilen: eine hohe Übertragungsrate an Daten, Vertrauen zu dem Dienstleistungsanbieter ist notwendig und es herrscht eine unsichere, sich in den letzten Jahren mehrfach geänderte Rechtslage in Deutschland bei der Verwendung US-amerikani-

scher Anbieter. Abbildung 10 zeigt den Vergleich des Einsatzes einer Storage Cloud aus Unternehmens- und Endverbrauchersicht.

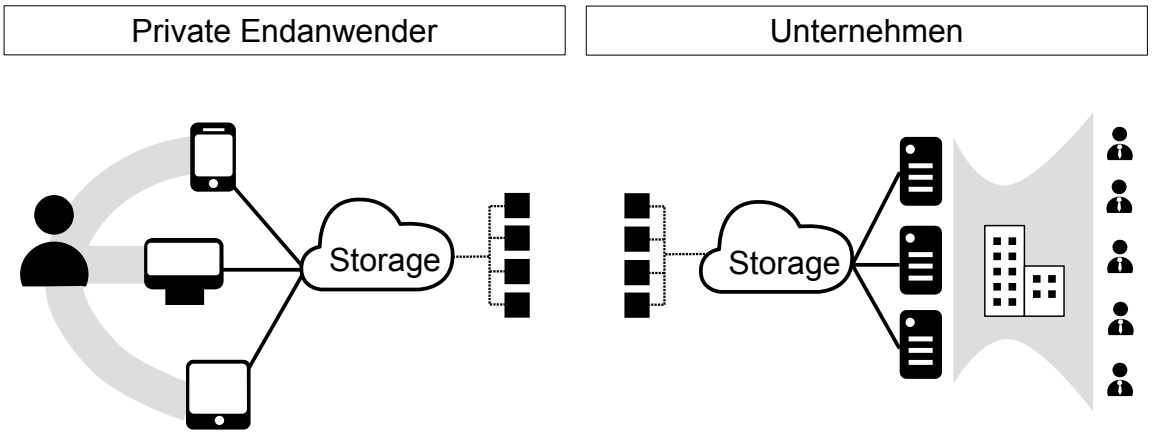


Abbildung 10: Storage Clouds von privaten Endanwendern und Unternehmen

Auch der Autor dieser Arbeit lässt es sich nicht nehmen, eine weitere Kategorisierung des As-a-Service-Paradigmas vorzunehmen. Diese basiert auf den Erfahrungen, traditionell entwickelte Systeme in eine Cloud-Infrastruktur zu integrieren und dabei, im Gegensatz zur Übertragung von ganzen monolithischen Anwendungssystemen, Teilleistungen anzubieten. Der Ansatz **Library as a Service** (LaaS) setzt gezielt am Microservice-Paradigma an und verwendet auf Basis des Container as a Service-Ansatzes nur die zu verwendende Bibliothek im Softwareentwicklungsprozess. Abbildung 11 zeigt eine schematische Darstellung traditionelle Programmier-Bibliotheken als Service anzubieten. Dieser Ansatz kann als Unterkategorie des Software as a Service-Paradigmas gesehen werden und erweitert diese nur um die notwendigen Softwarebibliotheken. Dadurch werden Vorteile in den Bereichen Performanz, Skalierbarkeit und Ausfallsicherheit erreicht. Die Bibliotheken stehen dann für eine Vielzahl von Anwendungen bzw. Nutzern bereit und werden stets in der aktuellsten Version angeboten. Ein Beispiel für die Bereitstellung einer Bibliothek ist das Einbinden mathematischer Softwarebibliotheken. Diese können gegebenenfalls in der Cloud-Infrastruktur des Anbieters betrieben werden. Hierbei ist ein Pay-per-Use-Bezahlmodell für die jeweils durchgeführte Berechnung zielführend.

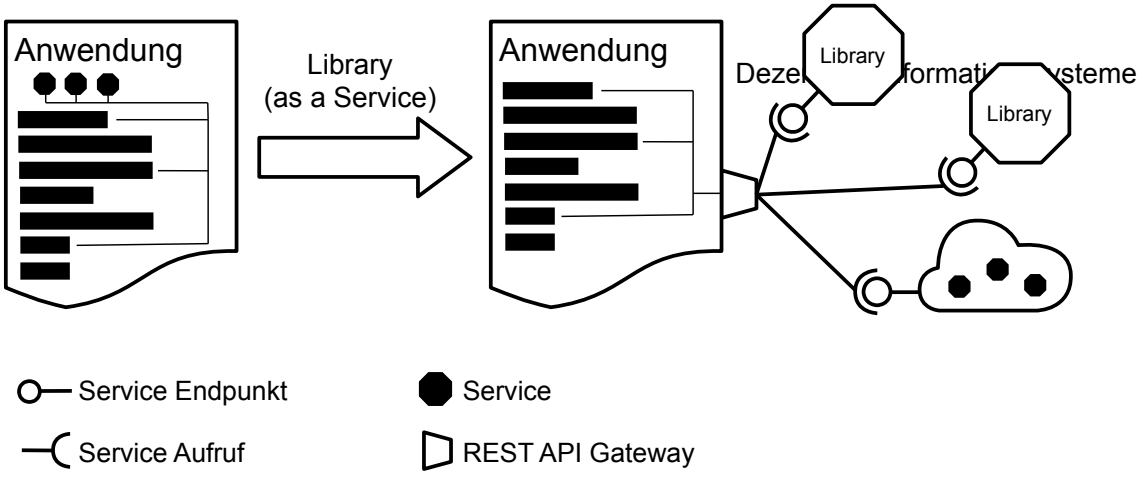


Abbildung 11: Library as a Service-Paradigma

Die Funktionsweise der Lösung wird ausgehend von den Best Practice-Angaben des Microservice-Ansatzes kurz beschrieben. Für die Kommunikation wird eine auf HTTP-basierende REST-API-Schnittstelle verwendet. Diese liefert das Ergebnis in Form eines JSON-Strings zurück. Die Anfragen werden durch einen GET-Request bzw. bei größeren Anfragen in Form eines POST-Requests durchgeführt. Für die Bereitstellung des Service wird ein Webserver eingesetzt, der die Anfragen bearbeitet. Die Umsetzung kann beispielhaft in Java mit Reflections erfolgen. Ist eine Anfrage fehlerhaft, so wird eine Exception in Form eines JSON-Strings zurückgegeben. Listing 1 zeigt vereinfacht die verschiedenen Komponenten, die notwendig sind, um diesen Ansatz in Java umzusetzen.

```
# Pseudo-Beschreibung der URI
http://localhost:8080/library/ [library name] / [method name] / parameter

# Beispiel einer Anfrage
http://localhost:8080/library/lang/math/atan2/0.25/0.5

# Klasse für die Zurückgabe der Ergebnisse
public class Result
{
    Object result;
    public Object getResult()
    {return result;}
    public void setResult(Object obj)
    {this.result = obj;}
}

# Verarbeitung der Anfrage
Result erg = new Result();
erg.setResult(Math.atan2(0.25, 0.5));
```

```
System.out.println(new Gson().toJson(erg));

# Ergebnis der Anfrage als JSON-String
{
       "result" : 0.4636476090008061
}

# Verwendung in der Zielanwendung ohne Umwandlung in Double
RestTemplate restTemplate = new RestTemplate();
String result =
restTemplate.getForObject("http://localhost:8080/library/lang/math/
atan2/0.25/0.5", String.class);
```

Listing 1: Umsetzung des Library as a Service-Ansatzes

Die angesprochene URL wird dynamisch vom Webserver aufgelöst und die Methode der Bibliothek dementsprechend aufgerufen und mit den Parametern ausgeführt. Das Ergebnis muss in der ausführenden Anwendung in das jeweilige Datenformat umgewandelt werden. Die Beschreibungen der einzelnen Methoden der Bibliothek stimmen mit der Beschreibung der Bibliothek für die Integration in ein Softwareprojekt überein. Die vorgestellte Lösung kann bei der Steigerung der Performanz eines in dieser Arbeit entwickelten Architekturmodells als sinnvoll erachtet werden.

Nachfolgend werden zwei aktuelle Forschungsgebiete näher vorgestellt, die als eine konsequente Erweiterung des Cloud Computings gesehen werden können: Emergent Software und Fog Computing.

2.3 Emergent Software

Die Ausführungen zu Emergent Software beziehen sich auf Aussagen des Fraunhofer-Instituts für Techno- und Wirtschaftsmathematik ITWM und dessen BMBF-gefördertes Projekt Software-Cluster.

Bei Emergent Software handelt es sich um das unternehmensübergreifende Zusammenspiel einzelner Komponenten und Dienste im Internet der Dinge. Diese passen sich dynamisch an die Anforderungen des Marktes an. Dadurch werden komplexe und dynamische Geschäftsprozesse ermöglicht und unterstützt. Das grundsätzliche Ziel ist die Steigerung der Wertschöpfung von KMU durch den Einsatz neuer Technologien. (vgl. [Software-Cluster 2015a]) Abbildung 12 zeigt eine schematische Darstellung von Emergent Software.

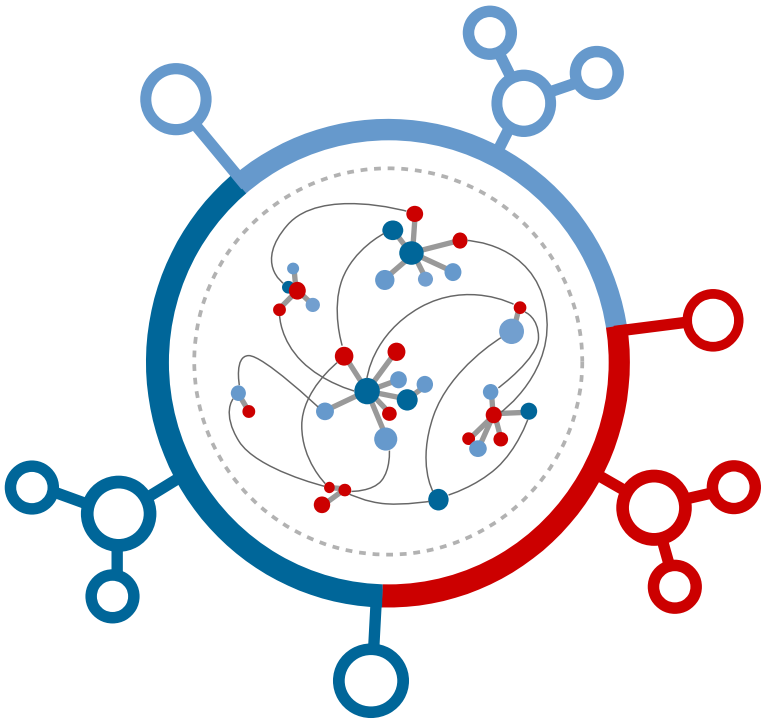


Abbildung 12: Schematische Darstellung von Emergent Software (in Anlehnung an [Software-Cluster 2015a])

Emergent Software folgt dem Prinzip der Emergenz.

Dadurch ist die flexible und dynamische Kombination einer Vielzahl an Komponenten unterschiedlicher Hersteller möglich. Softwaredienstleistungen passen sich dynamisch an die Anforderungen aus dem Markt und dem Geschäftsumfeld an und erfüllen die hochkomplexen Anforderungen digitaler Unternehmen. Dies führt zu neuen Angeboten und Geschäftsmodellen. Es kann somit als nächster Evaluationsschritt von Unternehmenssoftware gesehen werden. Grundsätzlich handelt es sich um ein Paradigmenwechsel hin zu digitalen Unternehmen. Diese nutzen Cloud-Technologien sowie kollaborative unternehmensübergreifende Software und binden mobile Endgeräte ebenso wie soziale Netzwerke in den Erstellungsprozess mit ein. Hierbei muss die Sicherheit von Anwendungen trotz der Öffnung durch die oben genannten Innovationen gewährleistet werden. (vgl. [Software-Cluster 2015b]) Fog Computing, welches im nächsten Kapitel vorgestellt wird, kann für diese Innovation den technologischen Unterbau bieten.

2.4 Fog Computing

Fog Computing, eingeführt von dem Telekommunikationsunternehmen Cisco Systems Inc., ist ein neues Modell, um eine drahtlose Datenübertragung auf verteilte Geräte für das Netzwerk-Paradigma Internet der Dinge (engl.: *Internet of Things [IoT]*) zu erleichtern. Grundsätzlich ist es als Erweiterung des Cloud Computings zu verstehen und stellt Daten-, Berechnungs-, Speicher- und Anwendungsfunktionalitäten für Endnutzer bereit. (vgl. [Abdelshkour 2015]) Unter technischen Gesichtspunkten kann es als sensorähnlicher Ansatz wie etwa das Peer-to-Peer (P2P) verstanden werden (vgl. [Vaquero/Rodero-Merino 2014, 28]). Vaquero und Rodero-Merino [2014, 30] stellten eine erste umfassende Definition auf:

„Fog computing is a scenario where a huge number of heterogeneous (wireless and sometimes autonomous) ubiquitous and decentralised devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of thirdparties. These tasks can be for supporting basic network functions or new services and applications that run in a sandboxed environment. Users leasing part of their devices to host these services get incentives for doing so.“

Die Vorteile dieses Paradigmas sind die Reduzierung der Servicelatenz, sowie die Erhöhung der Quality of Service[2] (QoS). Weiterhin erlaubt es Big Data[3] in Echtzeit unter ande-

2 Güte eines Kommunikationsdienstes aus der Sicht der Anwender

3 Große Datenmengen verwaltet von Unternehmen, Massendaten

rem im Bereich Echtzeitanalysen[4] (engl.: *Real Time Analytics*). (vgl. [Abdelshkour 2015]) Dies ist von besonderer Bedeutung, da mit der zunehmenden Anzahl von Daten eine Übertragung in einen zentralen Speicher deutlich erschwert wird. Durch das Ersetzen von zentralisierten Ansätzen und Anbietern steigert es gezielt den Datenschutz (vgl. [Vaquero/Rodero-Merino 2014, 30]).

Zusätzlich zu den offensichtlichen Vorteilen stehen diesem neuen Ansatz sieben Herausforderungen entgegen:

Herausforderungen des Fog Computings	
Name	Beschreibung
Discovery/ Sync	Anwendungen, die auf Endgeräten laufen benötigten eventuell eine (teil-) zentrale Anlaufstelle.
Compute/Storage limitation	Der aktuelle Trend zeigt eine Steigerung der Leistungsfähigkeit von Endgeräten. Dennoch benötigt es stetige Neuerungen bei Nicht-Endkonsumenten-Geräten.
Management	Milliarden von potentiellen kleinen Endgeräten müssen konfiguriert und verwaltet werden. Grundsätzlich wird es keine vollständige Kontrolle über die Fog geben. Deklarative Konfigurationstechniken werden zunehmen.
Security	Das Fog Computing stellt neue interessante Herausforderungen in den Bereichen Vertrauen und Datenschutz. Sandboxing für Endgeräte, die Anwendungen und Daten von Drittanbietern bearbeiten.
Standardisation	Zur Zeit existiert kein Standard für die Zusammenarbeit.
Accountability/ Monetisation	Teilnehmende Nutzer stellen dem Fog Ressourcen bereit. Hierfür werden Verantwortung und ein Monetarisierungs- und Refinanzierungskonzept benötigt.
Programmability	Den Anwendungslebenszyklus in Cloud-Umgebungen zu kontrollieren, ist bereits eine Herausforderungen. Diese steigt durch den Einsatz von kleinen Funktionalitäten auf sehr vielen Endgeräten.

Tabelle 5: Herausforderungen des Fog Computings (In Anlehnung an [Vaquero/Rodero-Merino 2014, 30])

Diese Arbeit nimmt sich besonders der Punkte Management und Sicherheit an. Hierbei wird durch eine gesteigerte Dezentralisierung die Verfügbarkeit der Ressourcen negativ beeinflusst. Ziel muss es sein, Lösungskonzepte zu entwickeln, welche die Verfügbarkeit gewährleisten, ohne auf Sicherheitsaspekte zu verzichten. Für eine bessere Abgrenzung gegenüber dem Cloud Computing bietet sich ein Vergleich beider Paradigmen an. (vgl. Tabel-

4 Analyse von Daten für die Verwendung für Geschäftsaufgaben zum benötigten Zeitpunkt

le 6)

Vergleich Fog und Cloud Computing		
Eigenschaft	Cloud Computing	Fog Computing
Latenz	Hoch	Niedrig
Verzögerungsjitter[5]	Hoch	Sehr niedrig
Servicelokalisation	Im Internet	An den Rändern des lokalen Netzwerkes
Distanz Client zu Server	Mehrere Hops[6]	Ein Hop
Sicherheit	Undefiniert	Kann definiert werden
Angriff bei Datenübertragung	Hohe Wahrscheinlichkeit	Sehr geringe Wahrschl.
Bewusstsein der Lokalisation	Nein	Ja
Geografische Verteilung	Zentral	Dezentral
Anzahl an Serverknoten	Wenige	Sehr viele
Unterstützung von Mobilität	Eingeschränkt	Unterstützt
Echtzeitinteraktionen	Unterstützt	Unterstützt
Art der letzten Meile	Mietleitungen	Kabellos

Tabelle 6: Vergleich Fog und Cloud Computing (vgl. [Abdelshkour 2015])

Basierend auf einem gut spezifizierten Fog Computing-Paradigma haben sich erste Umsetzungen etabliert. Diese beziehen sich auf eine grundlegende Architektur des Gesamtsystems und konkrete technische Ansätze mit bestehenden Technologien. Vaquero und Rodero-Merino [2014, 29] zeigen ein Konzept mit einer sehr nah am Nutzer befindlichen Edge Cloud. Hierbei handelt es sich zum Beispiel um private Clouds von Anwendern oder Anwendergruppen. Wer diese Cloud verwaltet und wie die technischen Geräte in diese eingebunden werden, ist bisher noch nicht spezifiziert. Es ist davon auszugehen, dass wirtschaftlich orientierte Beteiligte sich dieser Aufgabe annehmen. Ein Monetarisierungs- und Realisierungsmodell, welches in diesem Zusammenhang interessant ist, ist die des Anbieters FON ltd. (www.fon.com). Hier wird, in Kooperation mit großen Telekommunikationsunternehmen, weltweit ein flächendeckendes Hotspot-Netz für Endkunden bereitgestellt. An diesem Beispiel ist zu erkennen, wie ein Unternehmen bestehende Strukturen für das Anbieten eines neuen Geschäftsmodells nutzt und für Kunden einen Mehrwert generieren kann. Erstaunlich ist, dass den Nutzern des jeweiligen Telekommunikationsproviders keine

5 Zeitliches Taktzittern bei der Übertragung von Digitalsignalen, eine leichte Genauigkeitsschwankung im Übertragungstakt

6 Der Weg in Rechnernetzen von einem Netzknoten zum nächsten

Mehrkosten entstehen und dass die Dienstleistung sich dennoch für den Anbieter rentiert. Abbildung 13 zeigt die grafische Repräsentation des Fog Computing-Paradigmas, welches sich ähnlich des zuvor genannten Beispiels etablieren könnte.

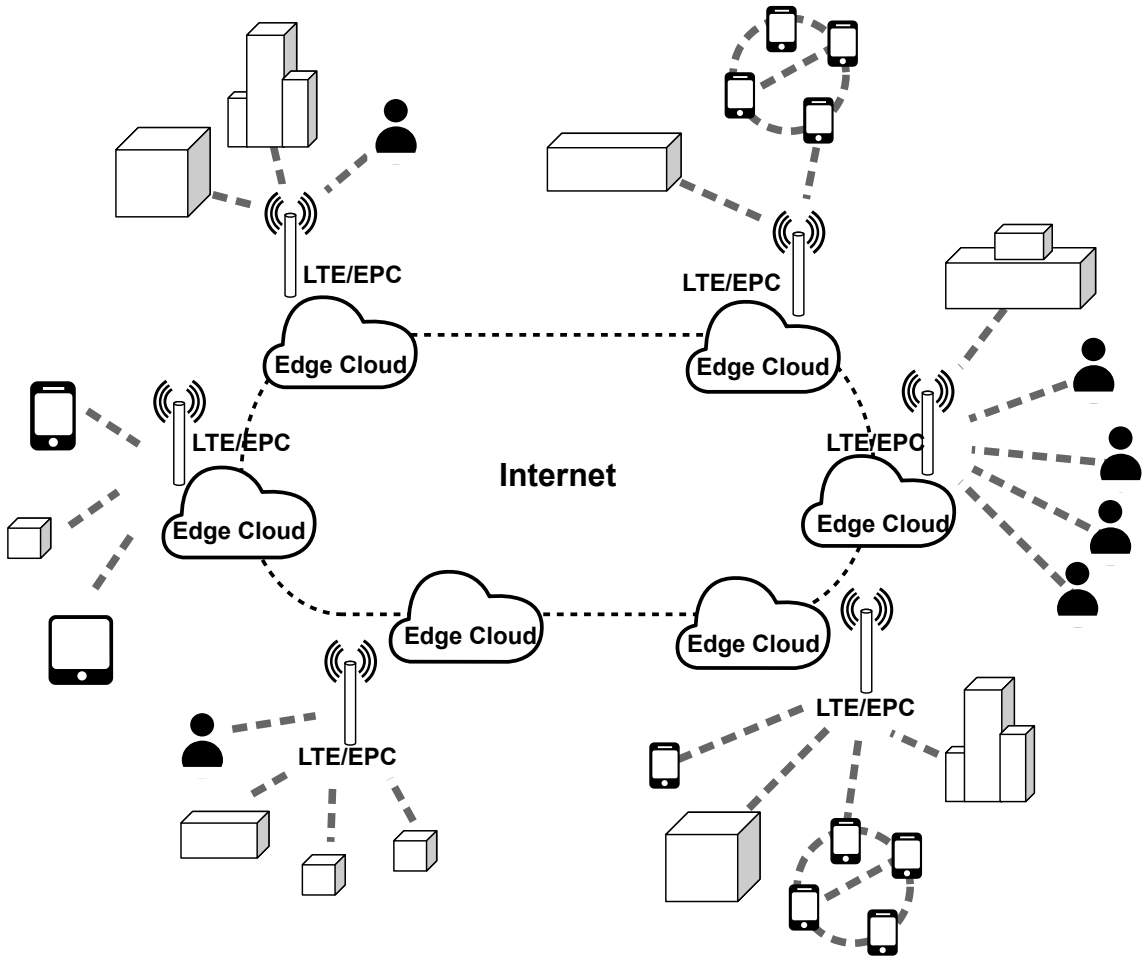


Abbildung 13: Edge Clouds als Eintrittspunkte für IoT und virtualisierte Sensornetzwerke (In Anlehnung an [Vaquero/Rodero-Merino 2014, 29])

Die technische Realisierung kann hierbei mit Hilfe von Wide/ Metropolitan Area Networks (WAN and MAN), Local Area Networks (LAN) oder Personal Networks (PN) erfolgen. Für die netzwerktechnische Umsetzung eignet sich der Peer-to-Peer-Ansatz.

Ein Anwendungsbereich, in dem Fog Computing seine Stärken ausspielen kann, ist im Big Data. In großen Cloud-Systemen mit Big Data-Strukturen wachsen die Schwierigkeiten bei einem Zugriff auf Objekte an. Gerade die Verteilung von Inhalten stellt ein Problem dar. Ein Aspekt, der eine Lösung darstellen könnte, ist das Swarm Computing. (vgl. [Bar-Magen Numhauser et al. 2013, 26ff])

Grundsätzlich können beim Fog Computing bezüglich der Beteiligten zwei Annahmen ge-

troffen werden: Die Bedeutung von Abonnentenmodellen steigt und neue Formen des Wettbewerbes und der Kooperationen werden sich etablieren. Beispiele für Bereiche, in denen Abonnements entstehen, sind Infotainment in verbundenen Fahrzeugen, Smart Grid, Smart Cities und Gesundheitswesen. Wettbewerbs- und Kooperationstransformationen beziehen sich primär auf global agierende Anbieter von (Software-)Dienstleistungen. In diesem Bereich werden sich neue Amtsinhaber sowohl als Nutzer als auch als Anbieter beteiligen, einschließlich Versorgungsunternehmen, Automobilhersteller, öffentliche Verwaltungen und Transportunternehmen. (vgl. [Bonomi et al. 2012, 14]) Wird Fog Computing mit neuen Technologien wie der Virtualisierung und der Teilvirtualisierung und neuen technischen Verfahren wie Microservices in Verbindung gebracht, liegt großes Potential in diesem neuen Paradigma.

2.5 Zusammenfassung

Dieses Kapitel stellte Informationssysteme und deren Typisierung als Fundament für die anschließende Betrachtung moderner Technologieansätze vor. Cloud Computing wurde hierbei als Grundlage in den Bereichen Eigenschaften und Merkmale näher beleuchtet. Ergänzt wurde diese Darstellung um die Erweiterung Everything as a Service. Mit Emergent Software als neuer Technologie- und Organisationsbeschreibung konnten neue Ansätze zur Leistungserstellung unter der Teilnahme verschiedener Hersteller und Dienstleister aufgezeigt werden. Abschließend wurde Fog Computing als Erweiterung des Cloud Computing-Paradigmas beschrieben. Dies zeigt nochmals verstärkt den Trend zur Dezentralisierung, wie es bereits in der Einführung thematisiert wurde. Das nächste Kapitel thematisiert den Datenschutz als eine bedeutende Einflussgröße für die Konzipierung aktueller Informationssysteme.

3 Datenschutz

Datenschutz besitzt in der heutigen digitalisierten Welt einen besonders hohen Stellenwert. Für die Erstellung eines Informationssystems mit dem Fokus auf Datenschutz ist es notwendig, diesen umfassend zu betrachten. Der Duden [2016] gibt für den Datenschutz eine allgemeine Definition:

„Schutz des Bürgers vor Beeinträchtigungen seiner Privatsphäre durch unbefugte Erhebung, Speicherung und Weitergabe von Daten [...], die seine Person betreffen“

In dieser Dissertation wird eine erweiterte Definition des Datenschutzes für kleine und mittlere Unternehmen eingeführt, da diese ebenfalls das Bedürfnis besitzen, eigene Daten zu schützen. Der KMU-Datenschutz ist eine Erweiterung des Datenschutzes über Privatpersonen hinaus. Bei diesem fordern KMU einen gleichen Schutz für ihre Geschäftsdaten, gerade im Bezug auf die Verwendung von Internetdienstleistungen aus dem US-amerikanischen Raum.
Durch die in den letzten Jahren gestiegene Verwendung von Computernetzen, haben Computer einen enormen Bedeutungszuwachs in verschiedenen Lebens- und Wirtschaftsbereichen erhalten. Dies ist unter anderem in den Feldern Arbeitswelt, Gesundheitswesen, Geschäfts- und Unterhaltungsnetzwerken ersichtlich. Eine erhebliche Rolle spielt hierbei das World Wide Web, durch welches Einzelplatzrechner oder Heimcomputer miteinander vernetzt werden und im Zuge dessen neue Datenschutzfragen aufgeworfen wurden. Die allgegenwärtige Computernutzung (engl.: *Ubiquitous Computing*) stellte Datenschützer hierbei vor neue Herausforderungen. (vgl. [Bieber 2012, 38])
Im Zusammenhang mit dem Datenschutz ist die Privatsphäre, welche eher dem US-amerikanischen Begriff Privacy entspricht, von zentraler Bedeutung. „Privatsphäre beschreibt, inwieweit ein Mensch anderen Menschen Zutritt zu seiner eigenen Welt gewährt.“ [Trepte 2012, 59] Es kann hierbei von einem Optimierungsprozess, der während der Interaktion mit anderen Menschen abläuft gesprochen werden. (vgl. [Trepte 2012, 59])
Es können vier Arten der Privatsphäre nach Trepte [2012] unterschieden werden. Die physische Privatsphäre beschreibt „inwieweit Menschen für andere physisch zugänglich sind“. Die psychologische Privatsphäre „beschreibt unsere Kontrolle über emotionalen und gedanklichen In- und Output.“ Die soziale Privatsphäre „umfasst den dialektischen Prozess,

Nähe zu bestimmten Menschen herzustellen und Distanz zu anderen aufzubauen“. Die informationsbezogene Privatsphäre „bezieht sich darauf, ob ein Mensch Kontrolle darüber hat, welche und wie viel Informationen an andere Personen weitergegeben werden“.

Es ist anzumerken, dass sich die Menschen heutzutage wesentlich intensiver mit der Frage auseinandersetzen, was mit ihren persönlichen Daten geschieht. Besonders der Informationsfluss im Internet, zum Beispiel in Form von sozialen Netzwerken, bedroht oder verletzt, auch durch eine aktive Beteiligung der Nutzer selbst, Datenschutzgrundregeln. (vgl. [Trepte 2012, 65]) Durch neue Möglichkeiten der Interaktion von Teilnehmern und neuen Formen von Datenverarbeitung ist zukünftig mit einer steigenden Preisgabe von privaten Informationen zu rechnen (vgl. [Trepte 2012, 65]).

Für eine umfassenden Überblick über die Datenschutzausprägungen weltweit bietet sich eine Gegenüberstellung der verschiedenen Länder an. Abbildung 14 gibt eine Übersicht zu der Datenschutzausprägung weltweit.

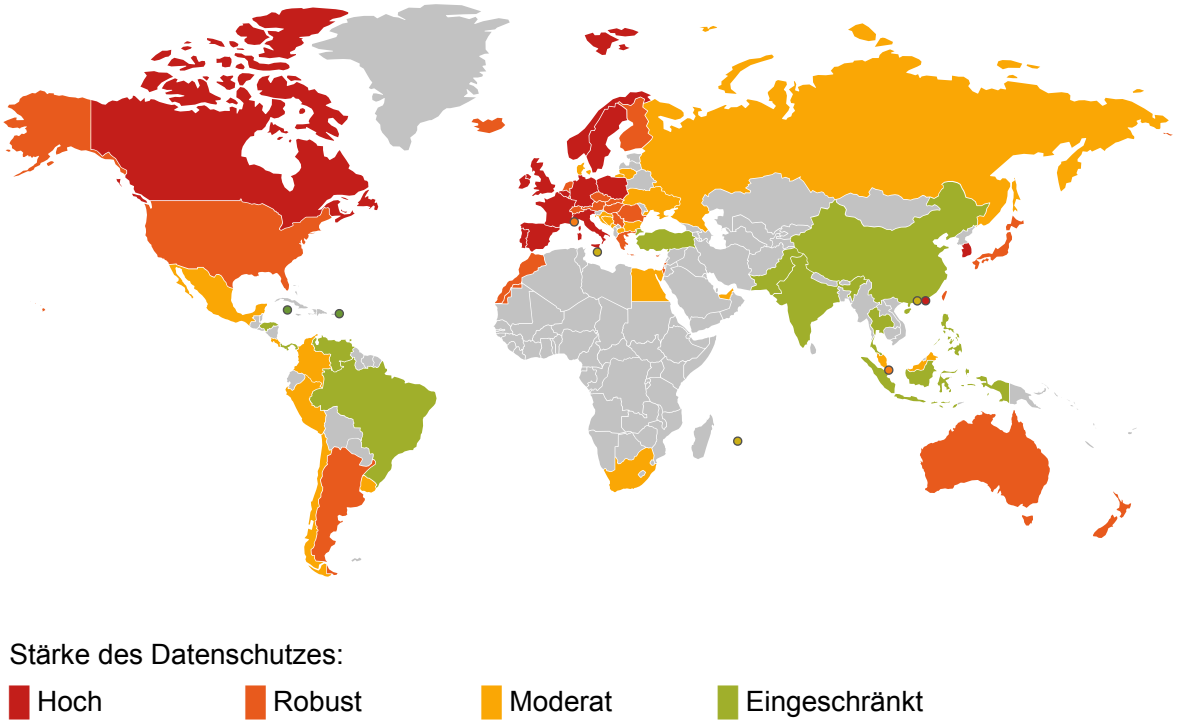


Abbildung 14: Weltkarte des Datenschutzes [DLA Piper 2015]

Es werden vier Kategorien festgelegt: Hoher, robuster, moderater und eingeschränkter Datenschutz. Bei hohem Datenschutz ist ein Datenschutzgesetz im jeweiligen Land vorhanden. Zudem erfolgt dessen strikte Umsetzung (Beispiel: Deutschland/ EU). Bei einer robusten Kategorisierung ist ein Datenschutzgesetz vorhanden, welches weitreichend umge-

setzt wird (Beispiel: USA). Länder mit einer moderaten Ausprägung besitzen zumindest Datenschutzrichtlinien, haben hierfür aber nur eine unzureichende Umsetzung. Bei einem eingeschränkten Datenschutz sind Datenschutzaspekte im Gesetzt vorhanden, diese werden aber nur unzureichend umgesetzt (Beispiel: China).

In dieser Dissertation ist der Vergleich des europäischen und des US-amerikanischen Datenschutzes von Bedeutung. Es ist zu erkennen, dass der Datenschutz in den USA geringer ausgeprägt ist als in der EU. Der nachfolgende Abschnitt erläutert für eine genauere Einordnung die Prinzipien und Aspekte des Datenschutzes.

3.1 Datenschutzaspekte und -prinzipien

Für eine zielgenaue Betrachtung von Datenschutz bietet es sich an, Datenschutzaspekte und -prinzipien, welche durch staatliche Institutionen aufgestellt werden, genauer zu betrachten. Im Folgenden werden die Prinzipien des Datenschutzes, sowie Privacy by Design und Privacy by Default vorgestellt.

Es existieren elf international anerkannte Datenschutzprinzipien, die auf der 27. internationalen Datenschutzkonferenz 2005 in Montreux festgelegt wurden und Teil der Erklärung von Montreux [o. A. 2005, 2 f] sind:

- Prinzip der Zulässigkeit und Rechtmäßigkeit der Erhebung und Verarbeitung der Daten
- Prinzip der Richtigkeit
- Prinzip der Zweckgebundenheit
- Prinzip der Verhältnismäßigkeit (vgl. Verhältnismäßigkeitsprinzip)
- Prinzip der Transparenz
- Prinzip der individuellen Mitsprache und namentlich der Garantie des Zugriffsrechts für die betroffenen Personen
- Prinzip der Nicht-Diskriminierung
- Prinzip der Sicherheit
- Prinzip der Haftung
- Prinzip einer unabhängigen Überwachung und gesetzlicher Sanktionen
- Prinzip des angemessenen Schutzniveaus bei grenzüberschreitendem Datenverkehr

Weiterhin existiert der Ansatz des Privacy by Design. Dieser fordert, Datenschutz bereits zu einem frühen Zeitpunkt bei der Entwicklung zu berücksichtigen. Dadurch können Datenschutzprobleme bereits frühzeitig identifiziert werden und neue Technologien positiv

beeinflussen. Dies bedeutet, Datenschutz in die Gesamtkonzeption zu integrieren. (vgl. [BFDI 2010, 1]) In einem engen Zusammenhang dazu steht Privacy by Default. Hierbei wird gefordert, die Grundeinstellungen innerhalb eines Systems auf ein hohes Datenschutzniveau voreinzustellen. Dadurch wird der Nutzer aktiv vor dem Missbrauch seiner persönlichen Daten geschützt und gerade technisch weniger versierte Nutzer werden sinnbildlich „an die Hand genommen". Somit zielt Privacy by Default auf eine „datenschutzfreundliche Grundeinstellung" der Unternehmen ab. (vgl. [EU-Kommission 2012b, 2]) Nachfolgend wird der deutsche Datenschutz betrachtet, da er für die in dieser Dissertation vorgestellte Konzeption wichtig ist.

3.2 Deutscher Datenschutz

Der deutsche Datenschutz gilt als einer der am stärksten kontrollierten und ausdifferenzierten weltweit. In dieser Dissertation ist er von besonderem Interesse, da deutsche KMU in die Betrachtung einbezogen werden. Diese müssen sich an den deutschen Datenschutz und seine Richtlinien halten. Die folgende Betrachtung bezieht die informationelle Selbstbestimmung, das Post- und Fernmeldegeheimnis, das Telekommunikationsgesetz (TKG) und das Bundesdatenschutzgesetz (BDSG) in die Aufbereitung ein.

Die informationelle Selbstbestimmung ist erstmals in dieser Begrifflichkeit 1983 mit dem Volkszählungsurteil in Erscheinung getreten. Das Bundesverfassungsgericht hatte sich intensiv mit der Frage auseinandergesetzt, welche Daten für eine Volkszählung aller Bürger von Deutschland gesammelt und gespeichert werden dürfen. In diesem Zusammenhang wurde die informationelle Selbstbestimmung erstmals festgelegt:

„Mit dem Recht auf informationelle Selbstbestimmung wären eine Gesellschaftsordnung und eine diese ermöglichende Rechtsordnung nicht vereinbar, in der Bürger nicht mehr wissen können, wer was wann und bei welcher Gelegenheit über sie weiß." [BFDI 2014]

Es handelt sich hierbei um ein Grundrecht, welches sich aus dem Grundgesetz ableiten lässt: Artikel 1 Absatz 1 und Artikel 2 Absatz 1 GG [GG 2014]. In diesem Zusammenhang sind Artikel 19 Absatz 1 und Artikel 19 Absatz 2 GG [GG 2014] von Interesse, die explizit aussagen, dass dieses Recht nur bei einem allgemeinen Gesetz mit Nennung des Grundrechtes eingeschränkt werden darf und dass es ein Verbot der Einschränkung des Grundgesetzes gibt. Einhergehend mit der Ableitung aus dem deutschen Grundgesetz sieht auch das Europarecht mit Artikel 8 Absatz 1 der Europäischen Menschenrechtskonvention eine ver-

gleichbare Einschränkung vor:

„Jede Person hat das Recht auf Achtung ihres Privat- und Familienlebens, ihrer Wohnung und ihrer Korrespondenz.“ [EMRK 1950]

In der digitalisierten Gesellschaft betrifft diese Regelung im besonderen Maße das Social Web und das Web 2.0. Derzeit verändern Facebook, YouTube oder Twitter die Art und Weise, wie Menschen personenbezogene Daten preisgeben. Dadurch verschiebt sich die Grenze zwischen Privatsphäre und Öffentlichkeit. (vgl. [Schmidt 2012, 220]) Im Web 2.0 reichen datenschutzrechtliche Aspekte über das Handeln der Nutzer hinaus. Sie umfassen letztlich alle Fähigkeiten, Nutzungspraktiken und sozialen Rahmenbedingungen der online-basierten Kommunikation. (vgl. [Schmidt 2012, 222])
Das Post- und Fernmeldegeheimnis wird in Artikel 10 des Grundgesetzes [GG 2014] festgelegt. Es weißt auf dessen Unverletzlichkeit hin und legt die Sicherheit der Kommunikationsmittel, wie etwa Post, Telefon, E-Mail, Voice Over IP und andere, ausdrücklich fest. Zusätzlich wird das Fernmeldegeheimnis nochmals genauer dargestellt in §88 Absatz 1 des Telekommunikationsgesetzes (TKG). Explizit heißt es da: „*Inhalt und nähere Umstände einer Kommunikation unterliegen der Geheimhaltung.*“ (siehe [TKG 2016])
Den gesetzlich orientierten Datenschutz in Deutschland regelt das Bundesdatenschutzgesetz (BDSG). In diesem findet die Ab- bzw. Eingrenzung zu personenbezogenen Daten statt. Der Schutz personenbezogener Daten ist in Deutschland im Landesrecht verankert. In §3 Absatz 1 (BDSG) wird der Begriff der personenbezogenen Daten explizit erläutert:

„Personenbezogene Daten sind Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person (Betroffener).“ siehe §3 Absatz 1 Bundesdatenschutzgesetz (BDSG) [BDSG 2015]

Holoubek [2007, 300] gibt eine weitere Beschreibung für Datenschutz:

„Anliegen des Datenschutzes ist es, [...] Vorgänge zur Gewährleistung des Persönlichkeitsschutzes der Betroffenen und sonstiger schutzwürdiger Geheimhaltungsinteressen rechtlichen Schranken iVm [sic!] geeigneten Rechtsschutzinstrumenten zu unterwerfen.“

Ausdrücklich zu erwähnen ist hierbei, dass sachbezogene Daten oder Daten juristischer Personen (Unternehmen, Vereine, Gemeinden, Landkreise, etc.) nicht durch das BDSG ge-

schützt sind. (vgl. [Heckmann 2012, 268])

Nachfolgend wird die Auswirkung des strikten Datenschutzes innerhalb Deutschlands beim Datentransfer mit den USA näher untersucht und die Probleme aufgezeigt.

3.3 Datentransfer EU-USA

Sehr viele Anwendungen, die im Bereich Internet von einer großen Anzahl von Menschen verwendet werden, bestehen aus Dienstleistungsangeboten von US-amerikanischen Unternehmen. Wie zuvor erwähnt, ist der Datenschutz in den USA geringer ausgeprägt als in Deutschland. Dies führt zu der Annahme, dass hierbei eine Unterordnung deutscher Interessen in Bezug auf den Datenschutz stattfindet. Europäische Konkurrenzprodukte stehen vor den Herausforderungen, einen gleichwertigen Dienst mit einem höheren Datenschutz anzubieten. Dies hat zur Folge, dass Europa nur einen geringen Anteil an Marktmacht bei Internetdienstleistungen besitzt. Aus diesem Umstand heraus ist es zielführend den Datentransfer zwischen EU und USA näher zu betrachten. Durch den Transfer entsteht die Gefahr eines potentiellen Missbrauchs und einer Zuwiderhandlung nach den Grundregeln des deutschen bzw. europäischen Datenschutzraumes. Damit dies vermieden wird, wurden bereits in der Vergangenheit diverse Abkommen zwischen USA und EU getroffen. Nach dem Aufheben des Safe-Harbor-Abkommens durch den Europäischen Gerichtshof, ist deren Vereinbarung mit dem Datenschutzrecht aus Europa aber fraglich. Abbildung 15 gibt eine Übersicht zum Zeitverlauf der heutigen und zukünftigen Abkommen zwischen EU und USA.

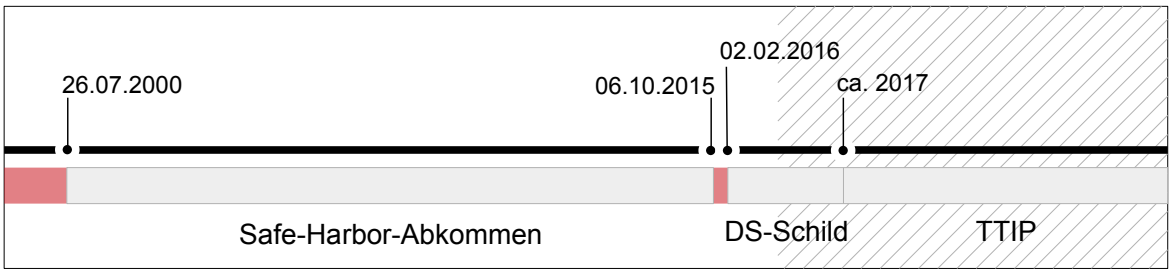


Abbildung 15: Zeitverlauf heutigen und zukünftigen Abkommen zwischen EU und USA

Nachfolgend wird zunächst die EU-Datenschutz-Grundverordnung (EU-DSGVO) erläutert und anschließend werden die Abkommen mit den USA betrachtet.

3.3.1 Europäische Datenschutz-Grundverordnung

Die Europäische Datenschutz-Grundverordnung ist ein Ansatz, sich den neuen Herausforderungen des technologischen Fortschritts im Internet zu stellen. Dadurch sollen die Rech-

te der Verbraucher gestärkt und die Pflichten der Anbieter erhöht werden. Im Mittelpunkt der Europäischen Datenschutz-Grundverordnung - KOM(2012) 11 steht die Stärkung der Rechte der betroffenen Personen. Diese lässt sich anhand von fünf Prinzipien identifizieren (vgl. [EU-Kommission 2012a, 9 ff.]):

- Transparenz und Modalitäten
- Information und Auskunftsrecht
- Berechtigung und Löschung
- Widerspruchsrecht und Profiling
- Beschränkungen

Die wesentlichen Ziele dieser neuen Verordnung haben eine größere Sicherheit für den Konsumenten zur Folge. Weiterhin soll das Vertrauen in Online-Dienstleistungen gestärkt werden. Durch einen vertrauenswürdigen Datenschutzrahmen kann das Wachstum von Unternehmen unterstützt werden. Dies wird unter anderem dadurch erreicht, dass es einen leichteren Zugang zu einer größeren Vielfalt an Waren und Dienstleistungen zu günstigeren Preisen gibt. (vgl. [EU-Kommission 2012b, 2])

Die grundsätzliche Datenschutzregulation umfasst eine Vielzahl von Punkten, von denen im Folgenden einige kurz vorgestellt werden. Eine große Besorgnis vieler Personen ist die dauerhafte Speicherung von Daten, die negative Folgen auf den Alltag haben kann. Diese sind nur sehr schwer aus dem Internet zu entfernen. Die Europäische Kommission nimmt sich dieses Punktes an und hat das Recht auf Vergessenwerden eingeführt. Somit ist es Privatpersonen möglich, auf Wunsch personenbezogene Daten löschen zu lassen, falls keine legitimen Gründe für deren weitere Speicherung vorliegen [EU-Kommission 2012b, 2]. Um zu überprüfen, welche Daten genau von Unternehmen gespeichert sind, ist der Zugang zu diesen Informationen erleichtert worden. Bei Fehlverhalten einer Institution (z. B. Unternehmen) muss der Kunde innerhalb von 24 Stunden darüber informiert werden, falls Daten versehentlich oder rechtswidrig vernichtet wurden, verloren gingen, geändert oder Unbefugten offengelegt worden sind. Damit Privatpersonen sich im Ernstfall zu helfen wissen, sind verbesserte Rechtsbehelfe bei Verstößen gegen den Datenschutz eingeführt worden. Neben Datenschutzbeauftragten innerhalb von Unternehmen werden behördliche Datenschutzbeauftragte eingesetzt. Die umfassende Neuregelung und Anpassung der Datenschutzbestimmungen innerhalb der EU an aktuelle Herausforderungen sorgte für ein relativ hohes Datenschutzniveau innerhalb Europas. Abbildung 16 zeigt Länder innerhalb

Europas und die Höhe des jeweiligen Datenschutzes.

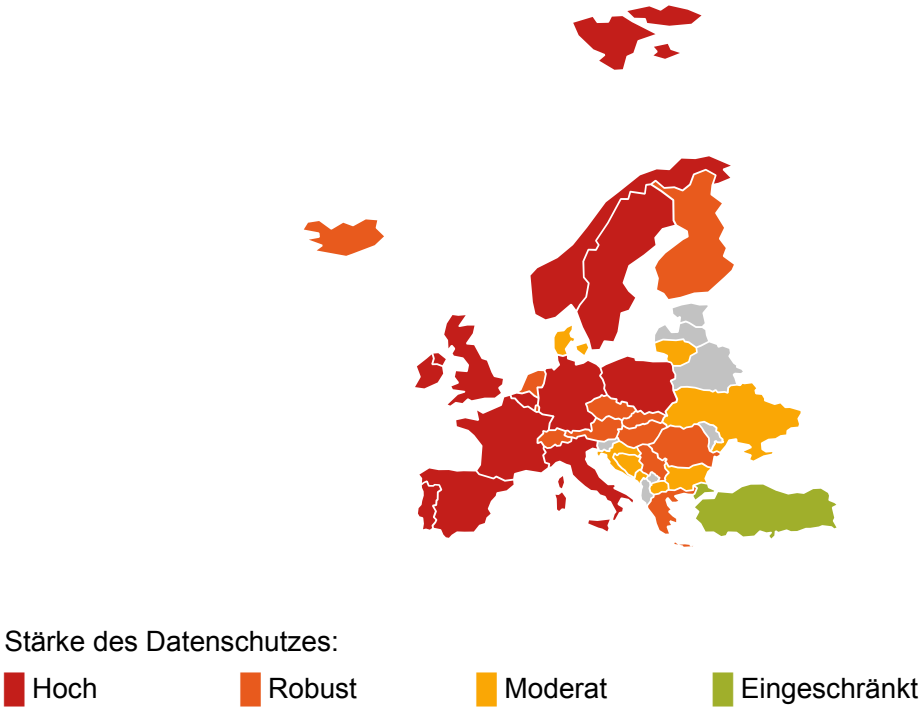


Abbildung 16: Europakarte des Datenschutzes [DLA Piper 2015]

Bezogen auf den Datentransfer in andere nicht zur EU gehörige Länder wurden Regelungen der Europäischen Kommission festgelegt. Die Richtlinie 95/46/EG der Europäischen Kommission verbietet es grundsätzlich, personenbezogene Daten in das Ausland zu transferieren. Dies ist speziell in Artikel 25 Absatz 4 geregelt:

„Stellt die Kommission nach dem Verfahren des Artikels 31 Absatz 2 fest, daß ein Drittland kein angemessenes Schutzniveau im Sinne des Absatzes 2 des vorliegenden Artikels aufweist, so treffen die Mitgliedstaaten die erforderlichen Maßnahmen, damit keine gleichartige Datenübermittlung in das Drittland erfolgt.“

Ist das Datenschutzniveau eines Drittstaates von der EU-Kommission anerkannt und als gleichwertig eingestuft, gelten dieselben Grundsätze wie innerhalb der EU. Ist dies nicht der Fall, muss die Aufsichtsbehörde den Datentransfer genehmigen. (vgl. [Däubler 2012, 196])
Nachfolgend werden Abkommen mit den USA näher betrachtet.

3.3.2 Safe-Harbor-Abkommen

Das Safe-Harbor-Abkommen wurde am 26.07.2000 eingeführt und regelte den Transfer von personenbezogenen Daten in die USA. Dies geschah ausdrücklich in Übereinstimmung mit den europäischen Datenschutzrichtlinien. Weichert und Schmidt [2012, 441] geben für das Safe-Harbor-Abkommen eine fundierte Beschreibung:

„Eine Vereinbarung des US Department of Commerce mit der Europäischen Kommission. US-Unternehmen, die sich den Safe-Harbor-Prinzipien unterwerfen, verpflichten sich selbst, die wichtigsten europäischen Datenschutzstandards einzuhalten. Damit dürfen personenbezogene Daten an sie oder von ihnen aus der Europäischen Union in die USA übermittelt werden.“ [Weichert/Schmidt 2012, 441]

Dies stellte in der Praxis aber eine Kontroverse da, wie Lüke [2012, 162] feststellt:

„Insbesondere die rechtliche Durchsetzung europäischer Datenschutzstandards gegenüber Firmen mit Sitz in den USA stellt den Verbraucherschutz vor ein unlösbares Problem. Zwar gibt es mit der → SafeHarbor-Regelung ein Abkommen, das Mechanismen für die Durchsetzung vorsieht. Doch praktisch ist das für Verbraucherseite kaum durchsetzbar.“ [Lüke 2012, 162]

Aus diesem Umstand heraus beschäftigte sich das Europaparlament mit der Aussetzung des Abkommens. Hierfür wurde eine Vorlage eingereicht, welche im März 2014 mit einer deutlichen Mehrheit angenommen wurde. Dieser demokratisch legitimierte Prozess wurde durch den Europäischen Gerichtshof (EuGH) unterbrochen. Dieser hat am 6. Oktober 2015 das Safe-Harbor-Abkommen für ungültig erklärt.
Dadurch kam es in Folge zu einer unsicheren Rechtslage. Diese wurde durch das EU-US Datenschutzschild, welches nachfolgend erläutert wird, vorläufig geschlossen.

3.3.3 EU-US-Datenschutzschild

Nach dem gesetzlichen Vakuum verursacht durch die Aufhebung des Safe-Harbor-Abkommens wurde am 02.02.2016 ein neuer Rahmen für die transatlantische Übermittlung von Daten ins Leben gerufen, das EU-US-Datenschutzschild. Dies war eine direkte Antwort auf die Forderungen, die der Gerichtshof der Europäischen Union in seinem Urteil vom 6. Oktober 2015 gestellt hatte. Damit unterliegen US-Unternehmen strengeren Auflagen bezüglich dem Schutz personenbezogener Daten aus der EU. Die USA verpflichten sich, den

Zugriff durch Behörden nur unter rechtlich ganz klar festgelegten Bedingungen, in einem begrenzten Umfang durchzuführen. Ein allgemeiner Zugriff auf die Daten ist nicht gestattet. (vgl. [EU-Kommission 2016])

Die Inhalte des neuen Abkommens bestehen aus drei Elementen:

- Strenge Auflagen für Unternehmen, die personenbezogene Daten europäischer Bürgerinnen und Bürger verarbeiten, sowie konsequente Durchsetzung
- Klare Schutzvorkehrungen und Transparenzpflichten bei Zugriff durch US-Regierung
- Wirksamer Schutz der Rechte der EU-Bürgerinnen und -Bürger durch verschiedene Rechtsbehelfe

Einhergehend mit dieser neuen Regelung meldeten sich erste kritische Stimmen, die dieses Abkommen ebenfalls als unzureichend betrachten. So sieht der Juristische Dienst des EU-Parlaments eine Unvereinbarkeit des neuen Rahmenvertrages mit den Grundrechten. (vgl. [Statewatch 2016]) Weitere Kritik betrifft das vorgesehene Klagerecht in Datenschutzfragen:

„Das vorgesehene Klagerecht in Datenschutzfragen in den USA erstrecke sich dagegen nur auf EU-Bürger, nicht jedoch auf Angehörige von Drittstaaten, die in einem Mitgliedsland lebten und so ebenfalls unter europäisches Recht fielen.“ [Krempl 2016]

Insgesamt muss davon ausgegangen werden, dass das aktuelle Abkommen, ebenso wie das Safe-Harbor-Abkommen, unzureichend ist. Grundsätzlich stellt sich die Farge, ob ein Abkommen mit Einhaltung des Grundrechtes überhaupt geschlossen werden kann. Einen erheblichen Einfluss auf diesen Prozess könnte das transatlantische Freihandelsabkommen (TTIP) haben, welches nachfolgend vorgestellt wird.

3.3.4 Datenschutz und Transatlantisches Freihandelsabkommen

Das Transatlantische Freihandelsabkommen wird derzeit geheim zwischen der EU und den USA verhandelt und soll zukünftig den Freihandel zwischen beiden Nationen erleichtern. Neben der Absenkung von Standards sehen viele Experten ein Eingreifen in den Datenschutz als Gefahr.

Basierend auf dem Comprehensive Economic and Trade Agreement (CETA) zwischen Kanada und der EU wurde TTIP konzipiert. Dadurch können erste zukünftige Konkretisierungen abgeleitet werden. Für den Datenschutz ist die Klausel von staatsunabhängigen Ge-

richten von Bedeutung, da sie es US-Internetkonzernen theoretisch ermöglicht, gegen die Datenschutz-Grundverordnung zu klagen. Inwieweit diese Vereinbarung zukünftig in das Abkommen integriert wird, kann zu diesem Zeitpunkt nicht seriös eingeschätzt werden. Dennoch ist es im Rahmen dieser Arbeit in die Betrachtung mit einbezogen.

Der Bundesverband IT-Sicherheit e. V. (TeleTrusT) beschäftigte sich intensiv mit TTIP in Verbindung mit Datenschutz und warnt vor einer Absenkung des deutschen bzw. europäischen Datenschutz- und IT-Sicherheitsstandards. Weiterhin wird die Gefahr einer Wettbewerbsverzerrung gesehen, wenn unterschiedliche Anforderungen an Datenschutzstandards für Unternehmen aus den USA und der EU bestehen. (vgl. [TeleTrusT 2015]) Diese Gefahr wird bestärkt durch das Drängen der USA auf eine Absenkung der Standards (vgl. [ZeitOnline 2015]). Nachfolgend wird betrachtet, ob ein Abkommen zwischen beiden Nationen auf einer rechtlich sicheren Grundlage zustande kommen kann und welche Implikationen die derzeitigen Schwierigkeiten auf den Standort Deutschland haben.

3.4 Implikationen für Wirtschaft und Gesellschaft

Die rechtlichen Schwierigkeiten des Datentransfers zwischen den USA und der EU zeigen deutlich auf, dass aktuelle Regelungen nicht ausreichen, um einen nationenübergreifenden Datenschutz zu konzipieren. Durch eine Globalisierung des Handels, gerade in Bezug auf digitale Güter und Dienstleistungen, entsteht der Bedarf für die Gestaltung neuer Normen in einem supranationalen Kontext, dem Codex Digitalis Universalis. Hierfür ist das europäische Datenschutzrecht als wegweisend zu erachten. Nur durch die Zusammenarbeit verschiedener Nationen ist es möglich, einen globalen Datenschutz basierend auf allgemeingültigen Standards zu etablieren. (vgl. [Weichert 2012, 348ff]) Es ist notwendig aktuelle Regelungen zu modernisieren. Hierfür legt Roßnagel [2012, 331 f.] drei Entwicklungsstufen der Digitalisierung bezogen auf Datenverarbeitung fest:

- Stufe 1: Die Datenverarbeitung betraf nur einen kleinen Ausschnitt des Lebens und war weitestgehend kontrollierbar.
- Stufe 2: Die weltweite Vernetzung von Rechnern sorgte für einen Kontrollverlust der Erhebung, Verbreitung und Verwendung von Daten durch den Betroffenen.
- Stufe 3: Das allgegenwärtige Rechnen der Datenverarbeitung findet Einzug in die Alltagsgegenstände der körperlichen Welt.

Darauf aufbauend wird der jeweilige Modernisierungsbedarf erläutert (vgl. [Roßnagel 2012, 332 f.]):

Stufe 1: Vereinfachung, Systematisierung und Effektivierung von Datenschutzregelungen mit unterschiedlichen Datenschutzniveaus, Ausnahmen und Gegenausnahmen.

Stufe 2: Sicherung der informationellen Selbstbestimmung auch in neuen Nutzungsformen. Weiterhin neue Konzepte des Selbst- und Systemdatenschutzes.

Stufe 3: Entwicklung von neuen Schutzkonzepten für die informationelle Selbstbestimmung in den Bereichen Transparenz, Zweckbindung, Erforderlichkeit und der Wahrnehmung von Betroffenenrechten gegenüber neuen Technikanwendungen.

Die aufgezeigte Entwicklung verdeutlicht die bevorstehenden Herausforderungen für Politik und Gesellschaft. Unabhängig von zukünftigen Bedarfen nach Modernisierungen ist die Verwendung der derzeitigen rechtlichen Rahmenbedingung für die Beteiligten Pflicht. Aus diesem Grund wird nachfolgend die derzeitige Ausgangslage in Deutschland näher beleuchtet. Die Situation in Deutschland gestaltet sich folgendermaßen: Ein deutsches Unternehmen muss den deutschen Datenschutz einhalten. Ein Unternehmen aus den USA agierend in Deutschland muss den deutschen Datenschutz nicht einhalten. Dieser Tatsache wird in Form von Abkommen zwischen den Nationen entgegengewirkt. Zu beachten ist hierbei, dass die USA sich in ihrer Souveränität nicht einschränken lassen und durch den Patriot Act auch nicht können. Dieser erlaubt zwangsweise den vollständigen Zugriff durch US-Behörden auf Daten von US-amerikanischen Unternehmen. Grundsätzlich behindern Datenschutzrichtlinien Unternehmen beim Anbieten von Dienstleistungen aller Art. Auf der anderen Seite schützen sie den Bürger eines Landes und damit letztendlich die Kunden. Basierend auf dieser Gegebenheit kann eine deutsche Strategie für Unternehmen abgeleitet werden. Der offensichtliche wirtschaftliche Nachteil kann in ein Alleinstellungsmerkmal deutscher Unternehmen umgewandelt werden. So kann Deutschland mit Berufung auf die Datenschutz-Grundordnung den Bedürfnissen der Kunden nach Datenschutz nachkommen. Dadurch ist es möglich, eine Bedeutung bei Internet-Dienstleistungen zu erlangen, welche derzeit nicht gegeben ist. Es handelt sich grundsätzlich um einen Export des deutschen Datenschutzes in andere Länder. Im übertragenen Sinne kann Deutschland somit die „Schweiz der Daten“ werden und als vertrauenswürdige Instanz (Trusted Party) auftreten. Ein weiteres Ziel ist die Speicherung deutscher Daten innerhalb Deutschlands, um eine höhere Kontrolle über diese Daten zu besitzen. Daraus abgeleitet ergeben sich erste Anforderungen an ein zu entwickelndes Informationssystem. Dieses hält den deutschen Datenschutz zum Schutz der Bürger und beteiligten Unternehmen ein. Aus gesellschaftlicher

Sicht wird somit die demokratisch getroffene Vereinbarung erfüllt und dadurch dem „Willen des Volkes“ nachgekommen. Ein weiteres Ziel muss die Transparenz und Kontrolle über die erzeugten Daten sein. Die Trusted Party als vertrauenswürdige Instanz bzw. Institution kann hierbei sowohl staatlich als auch privatwirtschaftlich organisiert sein. Das Vertrauen wird unter anderem geschaffen durch ausgestellte Zertifikate, externe Überprüfungen (z. B. TÜV), vollständige Transparenz und das Einhalten von Gesetzen.

3.5 Zusammenfassung

Dieses Kapitel gab eine aktuelle Übersicht zum Datenschutz. Zunächst fand eine allgemeine Einführung in den Datenschutz statt, ergänzt um Aspekte und Prinzipien. Spezifischer wurde der deutsche Datenschutz vorgestellt, da er in dieser Dissertation von Bedeutung ist. Bezogen auf die aktuelle Situation von Datenschutz und modernen Softwaredienstleistungen wurde der Datentransfer zwischen der Europäischen Union und den USA analysiert. Hierbei wurden Probleme und Herausforderungen aufgezeigt. Abschließend fand eine Darstellung von Implikationen für Wirtschaft und Gesellschaft statt. Aus den aufgestellten Grundlagen im Bereich Datenschutz können im nächsten Kapitel konkrete Voraussetzungen in Form von Anforderungen für ein Informationssystem ermittelt werden.

4 Anforderungsanalyse und derzeitige Konzepte

Die Anforderungsanalyse und die derzeitigen Konzepte bereiten das Fundament für die Gestaltung wissenschaftlicher Artefakte. Im Mittelpunkt der Untersuchung steht eine systematische Literaturanalyse, die dazu dient, Anforderungen aus der Literatur und einem Systemvergleich in den Anforderungskatalog eines zu entwickelnden Informationssystems (Cloud Network) zu übertragen.

4.1 Systematische Literaturanalyse

Eine Literaturanalyse eignet sich im besonderen Maße dazu, ein wissenschaftliches Thema, welches aktuell von Interesse ist, tiefgreifend zu beleuchten und aufzubereiten. Es dient bei einer wissenschaftlich gut angefertigten Arbeit, sowohl der Absicherung, ein Problem nicht erneut mit einem gleichen oder ähnlichen Ansatz zu lösen, als auch dazu, das Themengebiet fundiert einzuordnen und zu charakterisieren. Die Entwicklung von Konzepten softwaretechnischer Natur, kann mit einer umfangreichen Literaturanalyse sowohl eingeordnet als auch abgegrenzt werden. Nachfolgend wird zunächst der Ablauf beschrieben, Informationen und Wissen, aus der derzeitigen Forschungslandschaft zu extrahieren. Anschließend werden die Fragestellung und die Quellen aufgezeigt. Das Resultat ist eine Voranalyse der Ergebnisse in textueller und grafischer Form. Als Nächstes wird die Ergebnismenge erweitert und es erfolgen eine Analyse und eine Synthese der Inhalte. Schlussendlich werden die gefundenen Ergebnisse bewertet und in die Dissertationsschrift eingeordnet.

4.1.1 Ablauf

Für den Ablauf der systematischen Literaturanalyse werden in dieser Arbeit die folgenden fünf Phasen des Projektmanagements durchlaufen (vgl. [Burghardt 2012, 17ff]):

1. Initialisierung
2. Definition
3. Planung
4. Steuerung
5. Abschluss

Dadurch wird ein qualitativ hochwertiger Prozess für die Durchführung der Literaturanalyse garantiert. Die Initialisierungsphase beschäftigt sich mit der Definition des Ablaufs der Analyse im Allgemeinen. Hierbei werden die Grundlagen für eine erfolgreiche Durchführung gelegt. Weiterhin dient sie der Vorbereitung für die nachfolgenden Phasen. In der De-

finitionsphase (Kapitel 4.1.2) werden die Rahmenbedingungen für die Literaturanalyse festgelegt. Hierbei wird zunächst die Zielstellung der Literaturanalyse erläutert. Im Bereich Art und Umfang wird bestimmt, wie hoch das Volumen der zu untersuchenden Journalauswahl ist und welche Art von Literaturanalyse eingesetzt wird. Der Bereich Themengebiet legt den inhaltlichen Rahmen fest und dient als Fundament für die Auswahl an Such-Schlüsselwörtern und Journals. Die Planungsphase (Kapitel 4.1.3) dient als Vorbereitung der Durchführung und beinhaltet die Bildung einer Fragestellung. Weiterhin wird die Suchanfrage konstruiert. Der Bereich Journalauswahl konkretisiert die Festlegung der Journals und beschreibt deren Auswahl, basierend auf Themengebieten, näher. Dies beinhaltet ebenfalls die Auswahl an Quellen und Datenbanken für die spätere Abfrage. In der Steuerungsphase (Kapitel 4.1.4, 4.1.5, 4.1.6) wird die eigentliche systematische Analyse durchgeführt. Diese orientiert sich an dem Vorgehen von Denyer und Tranfield [2009]. Hierzu werden die Suchbegriffe in die jeweiligen Datenbanken der Journals eingegeben und die Ergebnisse gesammelt. Dadurch ist es im nächsten Schritt möglich, eine Voranalyse der Ergebnisse zu erstellen. Diese Ergebnismenge wird in einem nächsten Abschnitt um weitere externe Quellen ergänzt, die durch eine vorhergehende Recherche und ein Feedback von Experten bereits als zielführend identifiziert wurden. Diese Gesamtheit wird anschließend textuell und grafisch strukturiert und analysiert. Darauf aufbauend erfolgt eine Analyse und Synthese der Inhalte. In der Abschlussphase (Kapitel 4.1.7) erfolgt die Bewertung der Ergebnisse sowie eine Einordnung in die Dissertationsschrift. Der beschriebene Ablauf ist in Abbildung 17 schematisch dargestellt.

Abbildung 17: Schematische Darstellung Ablauf Literaturanalyse

4.1.2 Rahmenbedingungen

Die Rahmenbedingungen beleuchten die Zielstellungen, Art und Umfang sowie das Themengebiet näher. Dies bildet die Grundlage für die eigentlich Suche.

Die Zielstellungen dieser Literaturanalyse sind vielfältig in ihrer Ausprägung. Im Fokus steht die Identifikation der aktuellen Problemstellungen des Themengebietes. Hierbei sind offene Fragen und Forschungslücken von besonderer Bedeutung. Weiterhin werden eventuelle Lösungsansätze ermittelt und deren Herausforderungen dargestellt. Grundsätzlich soll die Analyse die Realisierung eines verteilten Informationssystem mit dem besonderen Interesse am Datenschutz beleuchten. Weiterhin interessiert der Einbezug neuer Technologien bei deren Umsetzung. Diese sind: Cloud Computing, Storage Clouds und Peer-to-Peer-Technologien.

In dieser Arbeit wird eine systematische Literaturanalyse nach Denyer/Tranfield [2015] durchgeführt. Diese definieren acht Schritte: Vorplanung, umfassende Suche, Titel- und Abstractprüfung, explizite Auswahlkriterien, Evaluation, Extraktion und Synthese, Berichterstattung und Verwertung (vgl. Abbildung 18). Die Berichterstattung erfolgt in diesem Abschnitt der Arbeit. Die Verwertung findet in Form eines Anforderungskatalogs für ein zu entwickelndes Informationssystem in einem späteren Teil der Arbeit statt. Ergänzt wird dieses Vorgehen um weitere Publikationen einer vorhergehenden Recherche sowie durch Expertenfeedbacks. Papers werden hierbei ab dem Jahr 2010 in die Ergebnismenge aufgenommen. Diese Entscheidung basiert auf einem initialen Forschungsbeitrag aus dem Jahr 2009. Yeung et al. [2009] haben hierbei erste Ansätze eines dezentralen Internets bezogen auf Datenschutz vorgestellt. Von besonderem Interesse ist, dass Tim Berners-Lee der Erfinder des HTML und Mitbegründer des World Wide Web ebenfalls an diesem Paper beteiligt war. Durch diese Festlegung wird eine Aktualität gewährleistet, gerade in Bezug auf den schnellen Wandel der IT-Landschaft.

Der Umfang der Analyse soll die aktuellen hoch bewerteten Inhalte (Ranking A+ bis B) der derzeitigen Forschungslandschaft abdecken. Daher werden 14 Journals in die Analyse einbezogen. Die genaue Auswahl erfolgt in Kapitel 4.1.3. Hierbei ist das Fachgebiet grundsätzlich der Wirtschaftsinformatik zuzuordnen.

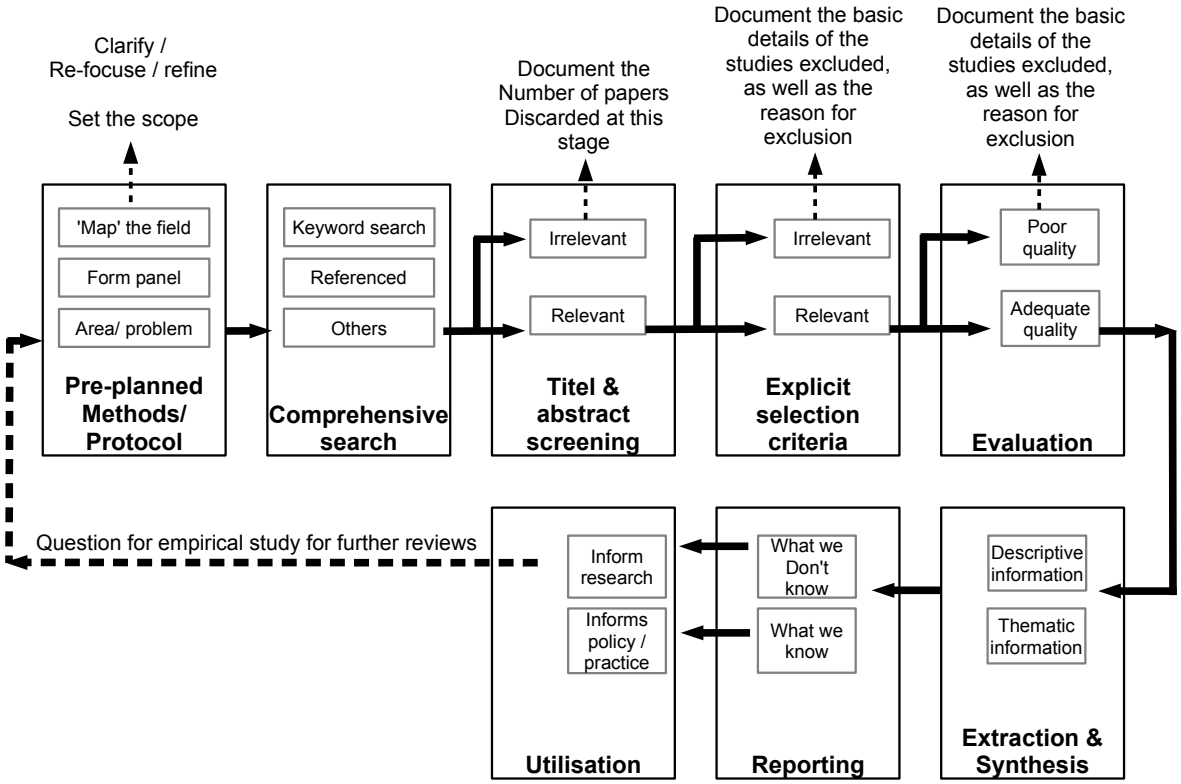


Abbildung 18: Acht Schritte der systematischen Literaturanalyse [Denyer/Tranfield 2015]

Das Themengebiet ordnet sich ein in die Konzeption von dezentralen Informationssystemen. Damit handelt es sich grundsätzlich um verteilte Anwendungen. Betrachtet werden in dieser Literaturanalyse die möglichen Umsetzungen sowie die innovativen Ansätze. Weiterhin werden potentielle Problemstellungen und Herausforderungen extrahiert. Im besonderen Fokus stehen hierbei Aspekte des Datenschutzes, sowohl für Privatanwender als auch für Unternehmen. Es konnte bei bisherigen Recherchen festgestellt werden, dass der Forschungsbereich soziale Netzwerke sich diesem Themengebiet im besonderen Maße angenommen hat. Daher werden diese mit in die Untersuchung einbezogen.

4.1.3 Fragestellung, Suchanfrage und Journalauswahl

Die Fragestellung orientiert sich an dem oben definierten Themengebiet. Hierbei werden alle genannten Aspekte in einer Frage formuliert, die nach der Analyse beantwortet werden kann. Ein weiteres Ergebnis ist der Vergleich verschiedener Konzepte. Dies ermöglicht in einem späteren Teil dieser Arbeit die Aufstellung von Anforderungen an ein zu entwickelndes System. Das Themengebiet umfasst grundsätzlich Informationssysteme, die dezentral organisiert sind.

Fragestellung:

„Wie können dezentrale Informationssysteme gestaltet werden, damit Datenschutzaspekte von Anwendern gewahrt werden? Welche Problemstellungen und Herausforderungen ergaben sich bei der Konzipierung?“

Aus der aufgestellten Fragestellung leitet sich die Verwendung einer zweistufig-parallelen Suchanfrage ab. Hierbei werden zunächst dezentrale Informationssysteme ermittelt. Diese bilden die eigentliche Grundlage und zeigen inwieweit diese Thematik von Interesse in den jeweiligen Journals ist. Darauf aufbauend wird für die Themengebiete soziale Netzwerke/ Datenschutz und dezentrale Technologien eine Untersuchung durchgeführt. Tabelle 7 gibt eine Übersicht sowohl über die Schlüsselwörter als auch über die eigentlichen Suchanfragen.

Suchbegriffe für Literaturanalyse			
Typ	#	Begriffe	Anfrage
Informationssystem (IS)	1	• decentral • decentralization • distributed • information • system	(decentral **OR** decentralization **OR** distributed) **AND** (information **OR** system)
Soziale Netzwerke Datenschutz (SNS)	2	• privacy • personal data • social network service • online social network	(privacy) **OR** (personal data) **OR** (social network service) **OR** (online social network)
Dezentrale Technologien (DT)	2	• cloud computing • storage cloud • peer-to-peer	(cloud computing) **OR** (storage cloud) **OR** (peer-to-peer)

Tabelle 7: Suchbegriffe für die Literaturanalyse

Hierbei ist zu beachten, dass zunächst Informationssysteme gesucht wurden. Diese sollen die Eigenschaft der Dezentralisiertheit besitzen. Dies wird erreicht, indem nach verteilten (engl.: *distributed*) oder dezentralen (engl.: *decentral, decentralization*) Informationssystemen (engl.: *system, information*) gesucht wird. Es werden, ebenfalls die Suchbegriffe „dezentrale Informationen“, sowie „verteilte Systeme“ mit in die Untersuchung aufgenommen.

Anschließend erfolgt eine Aufteilung in die Bereiche Soziale Netzwerke/Datenschutz (SNS) und Dezentrale Technologien (DT). Beide Bereiche erweitern die Suche und grenzen somit die dezentralen Informationssysteme ein. Der Bereich SNS beinhaltet zum einen Datenschutz (engl.: *privacy*) und ermittelt grundsätzlich Informationssysteme mit einem Datenschutzaspekt. Der Bereich Soziale Netzwerke (engl.: *social network service, online social network*) verwendet hierbei die in der Forschung geläufigen Begriffe. DT fügt die Aspekte des Cloud Computings und Peer-to-Peer mit ein. Somit werden Informationssysteme ermittelt, welche als Peer-to-Peer bzw. mit Cloud-Technologien umgesetzt wurden. Diese Herangehensweise begründet sich darin, dass sich in einer früheren Recherche gezeigt hat, dass Peer-to-Peer einen häufig eingesetzten Ansatz darstellt, verteilte Systeme zu realisieren.

Für die Untersuchung werden Journals der Wirtschaftsinformatik ausgewählt. Die Auswahl richtet sich nach dem VHB-Ranking [2015] und bezieht Journals der Kategorie A+, A und B ein. Tabelle 8 gibt eine Übersicht der ausgewählten Journals. Insgesamt wurden vierzehn internationale Journals aus dem Bereich Wirtschaftsinformatik ausgewählt. Nachfolgend werden die in der Tabelle aufgeführten Abkürzungen für die Bezeichnung der Journals verwendet.

Journal	Abk.	Kategorie	VHB-Ranking
Information Systems Research	ISR	Wirtschaftsinformatik	A+
Management Information Systems Quarterly	MISQ	Wirtschaftsinformatik	A+
Journal of Management Information Systems	JMIS	Wirtschaftsinformatik	A
Journal of the Association for Information Systems	JAIS	Wirtschaftsinformatik	A
Proceedings of the International Conference on Information Systems	ICIS	Wirtschaftsinformatik	A
Information Systems Journal	ISJ	Wirtschaftsinformatik	A
European Journal of Information Systems	EJIS	Wirtschaftsinformatik	A
Journal of the ACM	JACM	Wirtschaftsinformatik	B
Business & Information Systems Engineering (früher: Wirtschaftsinformatik WI)	BISE	Wirtschaftsinformatik	B
ACM Transactions on Information Systems	ACM IS	Wirtschaftsinformatik	B
ACM Transactions on Management Information Systems	ACM MIS	Wirtschaftsinformatik	B

Proceedings of the European Conference on Information Systems	ECIS	Wirtschaftsinformatik	B
ACM Transactions on Computer-Human Interaction	ACM HCI	Wirtschaftsinformatik	B
Human-Computer Interaction	HCI	Wirtschaftsinformatik	C

Tabelle 8: Journalauswahl für die Literaturanalyse

Für die Abfrage der Suchbegriffe werden insgesamt sieben Datenbanken einbezogen. Tabelle 9 gibt eine Übersicht zu den Datenbanken sowie zu den jeweiligen Hyperlinks.

Datenbanken der Journals	
Datenbank	Journals
INFORMS PubsOnLine URL: *http://pubsonline.informs.org*	ISR
AIS Electronic Library (AISeL) URL: *http://aisel.aisnet.org*	MISQ, JAIS, ICIS, ECIS
Taylor & Francis Online URL: *http://www.tandfonline.com*	JMIS, HCI
Wiley Online Library URL: *http://onlinelibrary.wiley.com*	ISJ
Ingentaconnect URL: *http://www.ingentaconnect.com*	EJIS
ACM Digital Library URL: *http://dl.acm.org*	JACM, ACM IS, ACM MIS, ACM HCI
Springer Professionals URL: *http://www.springerprofessional.de*	BISE

Tabelle 9: Datenbanken der Journals

Anschließend erfolgt die eigentliche Durchführung der systematischen Literaturanalyse. Die Ergebnisse werden im nächsten Kapitel ausgewertet.

4.1.4 Voranalyse der Ergebnisse

Die Voranalyse gibt einen ersten Überblick über die gefunden Ergebnisse der systematischen Literaturanalyse. Insgesamt wurden 3.630 Papers gefunden, die im weitesten Sinne dezentrale Informationssysteme zum Inhalt haben. Hier zeigen sich bereits der große Umfang und die Themenvielfalt. Nach dem Durchführen der zweiten Suchebene ergaben sich 2.417 Papers, die entweder Aspekte des Datenschutzes bzw. soziale Netzwerke und dezentrale Systeme zum Inhalt haben. Dies sind im Durchschnitt rund 170 Papers pro Journal.

Die meisten Papers wurden in den Journals ICIS, ACM IS und ECIS ausgemacht. Die wenigsten Papers fanden sich in dem Journal EJIS.

Nach der Analyse des Titels blieben 72 Papers übrig. Tabelle 10 gibt eine detaillierte Übersicht zu der jeweiligen Anzahl an gefunden Papers pro Journal und Ebene.

	IS	IS & SNS IS & DT	∑ SNS+DT	Auswahl nach Titel
ISR	275	105 33	138	13
MISQ	157	81 19	100	8
JMIS	204	88 24	112	7
JAIS	176	80 9	89	5
ICIS	610	352 73	425	12
ISJ	266	85 16	101	2
EJIS	30	2 0	2	0
JACM	189	132 0	132	2
BISE	146	89 36	125	4
ACM IS	376	363 0	363	9
ACM MIS	110	109 0	109	3
ECIS	640	347 105	452	3
ACM HCI	173	171 0	171	1
HCI	278	82 16	98	3
∑	3630	2417	2417	72

Tabelle 10: Erste Ergebnisse der Literaturanalyse

Nach der Durchsicht der jeweiligen Abstracts der ermittelten Publikationen wurden 16 Papers identifiziert, welche einen inhaltlichen Mehrwert für diese Arbeit liefern. Abbildung 19 zeigt dies nochmals schematisch.

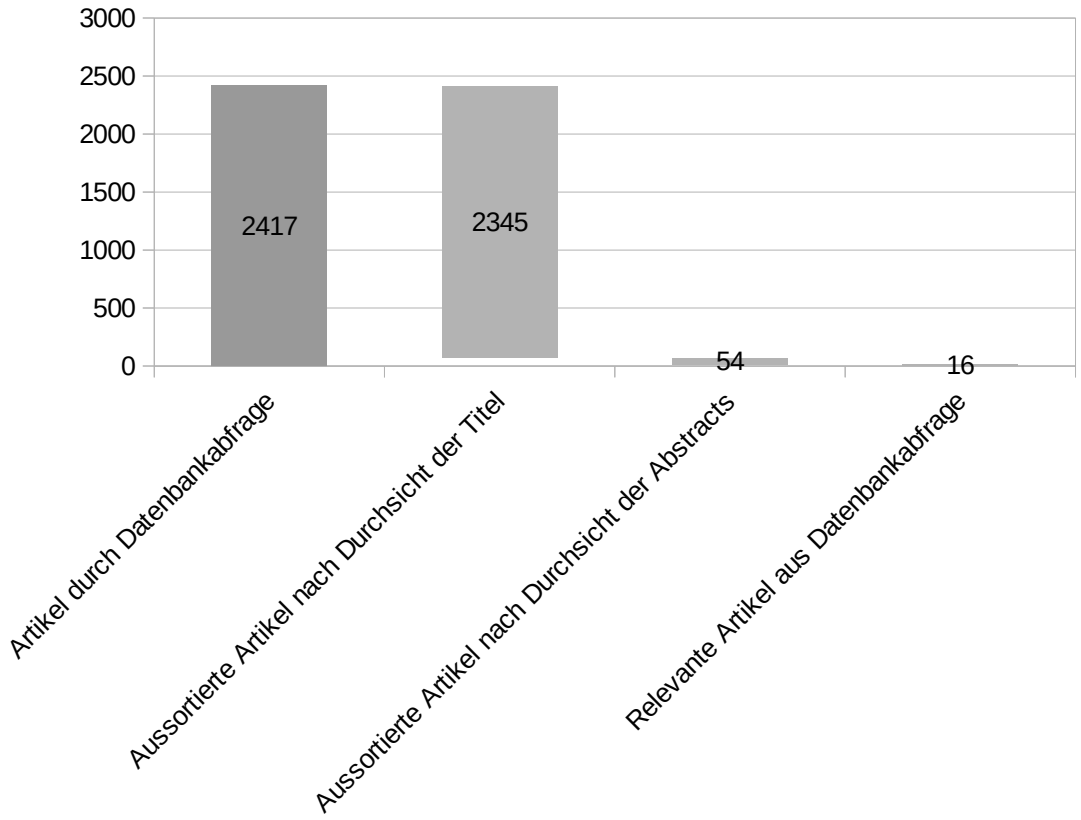


Abbildung 19: Erste Ergebnisse der Literaturanalyse

4.1.5 Erweiterung der Ergebnismenge

Die Ergebnisse der systematischen Literaturanalyse werden ergänzt mit Papers, welche zuvor in einer Recherche sowie durch Expertenfeedbacks als relevant eingestuft wurden. Im besonderen Maße sind dies Papers die sich grundsätzlich mit dem Themengebiet soziale Netzwerke beschäftigen. Dies begründet sich darin, dass hierbei ein großer Nutzerkreis in Bezug auf Datenschutz angesprochen wird. Dadurch hat dieses Themengebiet großes Interesse in der Forschergemeinschaft ausgelöst. Insgesamt werden 13 weitere Papers in die Ergebnismenge aufgenommen. Hierbei sind die Gestaltungen von Systemen und Architekturen ein wichtiger Fokus. Abbildung 20 gibt eine Übersicht über die Anzahl an Papers, welche zur Inhaltsanalyse herangezogen werden.

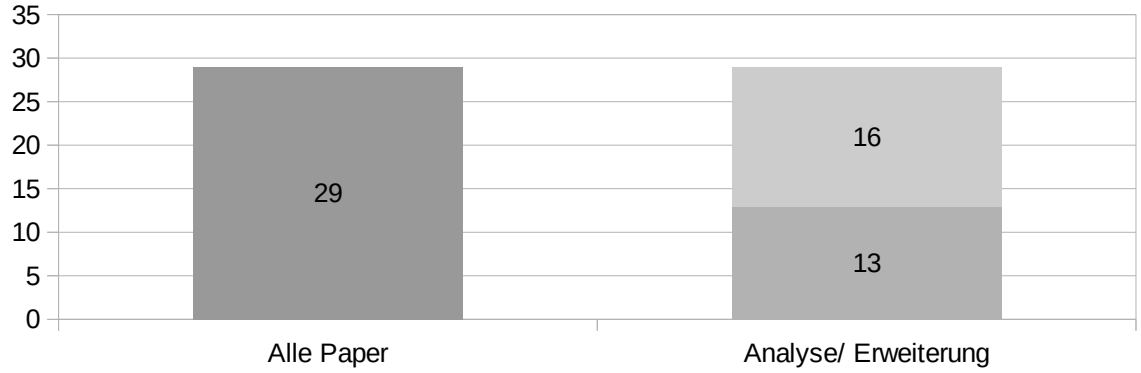


Abbildung 20: Ergebnis aller Papers für die Inhaltsanalyse

Mit insgesamt 29 Publikationen kann ein umfassender aktueller Stand des Themengebietes gewährleistet werden. Basierend auf ersten Ansätzen im Jahr 2009 ist eine leichte Abflachung in der Anzahl an Publikationen zu erkennen. In den Jahren 2011/2012 kam es zu einem Überdenken, ob die bisherigen Systeme entweder nicht funktionieren oder durch ihre Komplexität von den Anwendern nicht angenommen werden. Diaspora* hat dies nochmals unterstrichen, da dessen dezentrale Ausrichtung nicht im gewünschten Maß verwendet wurde. Daher müssen neue Ansätze gefunden werden, die die Anforderungen besser als bisher abdecken. Dies wird bestätigt in dem Paper von Narayanan et al. [2012]. Abbildung 21 gibt eine Übersicht zu den Publikationen und deren zeitliche Verteilung nach Jahren.

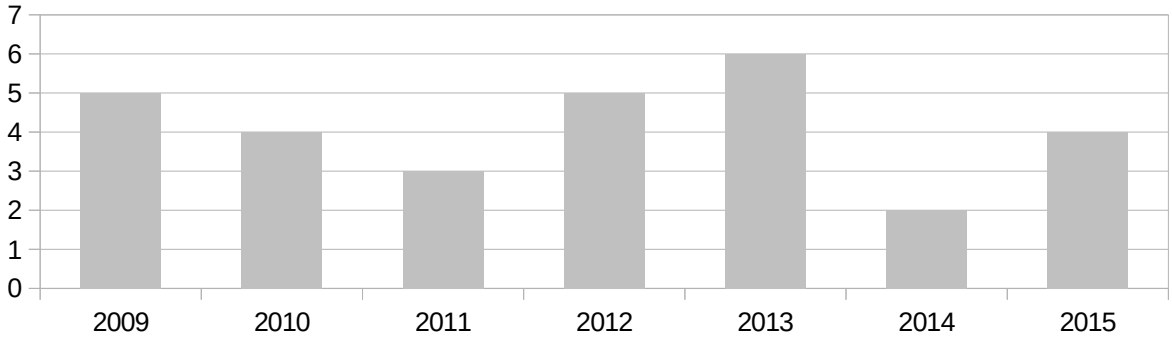


Abbildung 21: Jahresverteilung der Paper

Ein Vergleich der Verteilung der Papers nach Journals und Konferenzen zeigt, dass das Journal ACM Transactions on Information Systems (ACM IS) mit insgesamt vier Veröffentlichungen am stärksten vertreten ist. Positiv zu bewerten ist, dass das A+-Journal Information Systems Research (ISR) drei Publikationen stellt. Dadurch kann ein qualitativ hochwertige Ergebnisanalyse erfolgen. Abbildung 22 gibt eine Übersicht zu der Verteilung der Papers nach Journals und Konferenzen.

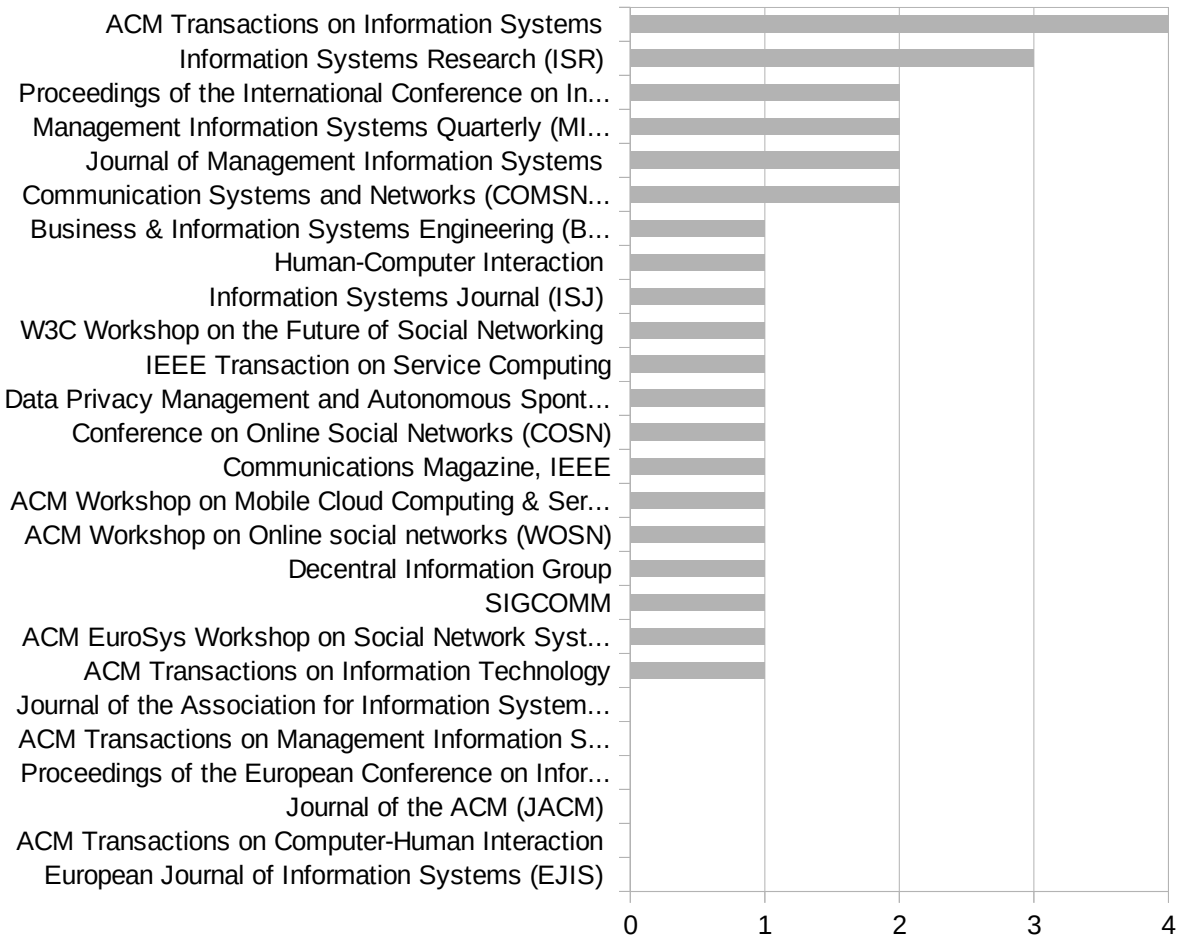


Abbildung 22: Anzahl der Publikationen pro Journal und Konferenz

Im nächsten Abschnitt erfolgt die Analyse und Synthese der Inhalte. Hierbei werden Erkenntnisse für die spätere Konzipierung eines Informationssystems gewonnen.

4.1.6 Analyse und Synthese der Inhalte

Softwaredienstleistungen besitzen eine zunehmend entscheidende Rolle bei der Abstraktion sowie Erweiterung der realen Welt. Im weiter gefassten Verständnis handelt es sich hierbei um virtuelle Welten. Somit finden grundsätzliche Prinzipien und Eigenschaften Anwendung, die jedem neu zu entwickelnden System zu Grunde gelegt werden müssen, um erfolgreich am Markt der zunehmenden Sozialisierung bestehen zu können. Ergänzt wird diese Perspektive durch sich dynamisch anpassende Systeme und Softwarebestandteile unterschiedlicher Hersteller, genannt Emergent Software[7]. Chaturvedi et al. [2011, 667 ff.] definieren hierbei fünf Emergent-Struktur-Design-Prinzipien für virtuelle Welten:

1. Beherbergen diverser Nutzer (Bürger)
2. Bürgerorientierter Blick auf die virtuelle Welt
3. Ermöglichen, Erhalten und Schützen der vom Benutzer erzeugten Inhalte
4. Mehrere Ebenen von computergestützten Experimenten
5. System muss reale und virtuelle Welten vereinen

Das Beherbergen diverser Nutzer kann in Bezug auf ein Informationssystem die Bereitstellung verschiedener Funktionalitäten bedeuten. Hierbei kann neben der kommunikativen Interaktion zum Beispiel auch die Möglichkeit eines integrierten Projektmanagements verstanden werden. Hierbei sind die zu erzeugenden Arten von Daten grundsätzlich anpassbar zu entwickeln und es sollte durch Zusammensetzung die Möglichkeit gegeben sein, neue Arten zu erschaffen. So kann aus logistischer Sicht eine Transportkette aus Akteuren, Transportgütern und Kartenpositionen bestehen. Der bürgerorientierte Blick auf die virtuelle Welt bezieht sich auf die Abstraktion von Daten, damit jedem Teilnehmer nur die Daten angezeigt werden, die für seine persönliche Arbeit notwendig sind. Das Ermöglichen, Erhalten und Schützen der vom Benutzer erzeugten Inhalte zielt zum einen auf die flexible Datenerzeugung und zum anderen auf den Schutz des geistigen Eigentums ab. Das Prinzip der mehreren Ebenen von computergestützten Experimenten ist weniger auf ein Informationssystem als im klassischen Sinne auf virtuelle Welten anwendbar. Die Vereinigung von realer und virtueller Welt bezieht sich auf die Unterstützung und Begleitung des Nutzer im realen Alltag. (vgl. [Chaturvedi et al. 2011, 680])

Ein wichtiger Bestandteil moderner Informationssysteme sind die sozialen Interaktions-

7 [Software Cluster 2015]: Emergente Software kombiniert dynamisch und flexibel eine Vielzahl von Komponenten unterschiedlicher Hersteller, um die hochkomplexen Anforderungen digitaler Unternehmen zu erfüllen.

möglichkeiten mit anderen Nutzern. Weiterhin ermöglichen sie eine Kooperation zum Erreichen gemeinsamer Zielstellungen sowie den Austausch von Knowhow. Hierfür benötigt es ein System zum Management der umfangreichen Kontakte (vgl. [Davison et al. 2013, 104]). Dies hat ebenfalls einen positiven Einfluss auf die Datenqualität an sich, da es wichtiger ist, von wem die Daten stammen als deren inhaltliche Aussagekraft (vgl. [Davison et al. 2013, 104]).

Die Realisierung von dezentral-orientierten Informationssystemen, in Verbindung mit der Steigerung des Datenschutzes für die Anwender mündet zumeist in der Idee von einer vollständigen Dezentralisierung des Systems. Hierfür wurden viele neue Prototypen von Peer-to-Peer-Systemen entwickelt, wobei keines von diesen einen weiten realen Einsatz hatte (vgl. [Tigelaar et al. 2012, 9:1]). Die am Netzwerk als Händler von Objekten Beteiligten können sowohl tangible als auch intangible Güter tauschen. Darauf aufbauend haben Andersson et al. [2013] eine Kategorisierung von Peer-to-Peer-Typen vorgenommen. (vgl. Tabelle 11.) Die ausgeprägten Eigenschaften der zeit-basierten Vergleichskategorien deuten darauf hin, dass Echtzeit nicht in jedem Fall als Anforderung gegeben ist.

	File sharing	Trading	Goods sharing	Service sharing
Object of exchange	Digital material	Tangible material	Tangible material	Intangible encounter
Timing requirement	No	Not necessarily	Not necessarily	Yes
Meeting requirement	No	No	Not necessarily	Yes
Example	Napster	eBay	AirBnb	Avego

Tabelle 11: Fünf Archetypen des Peer-to-Peer-Austausches (In Anlehnung an [Andersson et al. 2013, 3])

Bezugnehmend auf die Formen von Interaktionsmöglichkeiten ergibt sich hierbei eine Unterscheidung von Direktkommunikation, wie zum Beispiel Chat, und zeitversetzter Kommunikation, wie zum Beispiel das Kommentieren von Inhalten. Eine weitere Kategorisierung von datentransferorientierten Peer-to-Peer-Systemen kann anhand von Anwendungstypen erfolgen. (vgl. Tabelle 12) Hierbei zeigt sich die Schwierigkeit, ein System einzuordnen, welches mehrere Anwendungstypen vereint. Daraus kann geschlussfolgert werden, dass die Umsetzung eines solchen Informationssystems eine Herausforderung bezüglich der zu erfüllenden Anforderungen darstellt.

Anwendungstypen	Beispielanwendungen
Content Distribution	Usenet, Akamai, Steam
File Sharing	Napster, Kazaa, Gnutella, BitTorrent
Information Retrieval	Sixearch, YaCy
Instant Messaging	ICQ, MSN
Streaming Media	Tribler, Spotify
Telephony	Skype, SIP

Tabelle 12: Übersicht zu Arten und Beispielen von Peer-to-Peer-Anwendungen [Tigelaar et al. 2012, 9:3]

Der geringe Einsatz solcher Systeme im Ökosystem der Informationssysteme für Unternehmen und private Endanwender in Bezug auf die Übermacht zentralisierter Systeme, kann anhand der aktuellen Herausforderungen analysiert werden (vgl. [Tigelaar et al. 2012, 9:4]):

- Effiziente Nutzung von Ressourcen
- Bereitstellung von akzeptabler Servicequalität
- Garantie der Robustheit
- Sicherstellen, dass Daten verfügbar bleiben
- Anonymität

Diese werden um Herausforderungen erweitert, die an die Forschung gestellt werden: Simulation und standardisierte Tests (vgl. [Tigelaar et al. 2012, 9:4]). Ergänzt werden sie mit der Zielstellung einer effektiven und zeitsparenden Suche und dem Auffinden von Inhalten. Grundsätzlich gilt: Je mehr Knoten in einem P2P-Netzwerk involviert sind, desto länger dauert die Suche nach Inhalten (vgl. [Hosanagar et al. 2010, 12]). Daraus resultiert, ein System zu konzipieren, welches möglichst wenig Abhängigkeiten und Verknüpfungen der Entitäten untereinander aufweist und dennoch eine qualitativ hochwertige Bereitstellung von Informationen erlaubt.

Ein wichtiger Kernfokus bei Systemen mit einer hohen Verteilungsdichte von Daten ist Vertrauen. Hier hat sich gezeigt, dass die Zentralisierung von Vertrauen einen positiven Einfluss auf die Leistungsfähigkeit von verteilten Teams hat (vgl. [Sarker et al. 2011, 298]). Im weiter gefassten Sinne kann die Erstellung von digitalen Inhalten durch eine vertrauenswürdige Instanz (trusted party) bereichert werden und sorgt für eine unbelastete Zusammenarbeit. Gerade in der Kommunikation spielt Vertrauen eine vermittelnde Rolle (vgl. [Sarker et al. 2011, 298]). Ein Anwendungsbereich stellen hierbei soziale Netzwerke

dar. Freunden aus der realen Welt wird hierbei mehr Vertrauen entgegen gebracht als Onlinefreundschaften (vgl. [Liu et al. 2015, 21]). Wenn neben sozialen Interaktionen auch ökonomische Aktivitäten eine Rolle spielen, steigt die Wichtigkeit der Vertrauensbasis.

Es ist zielführend Realisierungs-Konzepte aus der Forschung zu betrachten, um Erkenntnisse für die zukünftige Entwicklung zu gewinnen. Nachfolgend werden zwei Konzepte von Vertrauen bezogen auf Datenbestände näher betrachtet. Williams und Sion [2013] setzten sich mit der Problemstellung auseinander, in einer nicht vertrauenswürdigen Umgebung Daten sicher zu speichern. Dazu entwickelten sie ein Datenzugriffsprotokoll, welches die Inhalte aufteilt und auf verschiedene Provider speichert (vgl. [Williams/Sion 2013, 12:1 / 12:27]). Ein Oblivious RAM speichert hierbei verteilte Daten auf der Storage Clouds und setzt diese bei Bedarf zusammen (vgl. [Williams/Sion 2013, 12:4]). Erway et al. [2015] lösen das Problem der nicht vertrauenswürdigen Server mit Dynamic Provable Data Possession (DPDP). Hierbei behält der Client eine kleine Anzahl an Metadaten und sendet den Rest an einen nicht vertrauenswürdigen Server (vgl. [Erway et al. 2015, 15:2]). Bei beiden Konzepten ist kritisch zu hinterfragen, ob die Infrastruktur bzw. das Management für Daten nicht eher eine Anpassung an neue Privacy-Aspekte benötigt. Dies stellt neue Herausforderungen an die Gesetzgebung (Datenschutz) und an die Unternehmenslandschaft. Weiterhin zeigt sich eine Schwierigkeit, die Daten zu jeder Zeit zur Verfügung zu stellen. Dadurch ergibt sich für diese Arbeit die Zielstellung, Vertrauen zentralisiert zu realisieren. Ein vielfach geäußerter Wunsch von Anwendern ist die Möglichkeit, Daten endgültig zu löschen. Dies wird unter dem Begriff „Recht auf Vergessenwerden“ in den Medien diskutiert. Karla et al. [2010] schlagen vor, dass ein Nutzer bei der Speicherung einer Information angeben können soll, wie lange diese gespeichert wird, bevor sie gelöscht wird (vgl. [Karla 2010, 105]). Hierbei ergeben sich Probleme bei der Umsetzung (vgl. [Karla 2010, 106]):

- Einsatz einer technischen Lösung
- Psychologisch: Kontraproduktiv, da einfaches Löschen die Hemmschwelle zum Veröffentlichen von persönlichen Daten senkt
- Bei sehr aktiven Nutzern werden viele Nachrichten zum Löschen geschickt
- Informationen könne einfach kopiert werden, Kontrolle schwierig bis unmöglich
- Erstellte Informationen von Dritten könne nicht kontrolliert werden

Somit muss ein Konzept entwickelt werden, welches den Nutzer intuitiv bei dieser Aufgabe unterstützt. Nur so kann ein Anreiz geschaffen werden, diese Funktionalität zu verwen-

den und implizit den Datenschutz zu verbessern.

In diesem Zusammenhang ist die Bedeutung des Datenschutzes (Privacy) für Informationssysteme eine wichtige Komponente, im besonderen Maße als Verkaufsargument heutiger Internetdienstleistungen. Tsai et al. [2011] führten eine experimentelle Studie durch, inwieweit die Anzeige von Datenschutz die Zahlungsbereitschaft der Probanden beeinflusst. Hierfür wurden Teilnehmer beim Einkauf auf Webseiten beobachtet, auf denen der Datenschutz symbolisch dargestellt wurde. Es konnte herausgefunden werden, dass ein höherer Preis gezahlt wurde, wenn der Datenschutz höher war. Das zeigt, dass Individuen eine Prämie zahlen würden, wenn die Datenschutzinformationen leicht zugänglich wären. Daraus wird geschlussfolgert, dass Datenschutz deutlich hervorgehoben werden muss, um einen positiven Einfluss auf das Verhalten der Nutzer zu nehmen. (vgl. [Tsai et al. 2011, 25]) Für die Visualisierung und Verdeutlichung von Datenschutz bieten sich zum Beispiel Icons an (vgl. [Tsai et al. 2011, 30]). Eine Möglichkeit Datenschutz zu realisieren, ist das sogenannte Beziehungsmanagement. Dieses ist ein im Umfeld von sozialen Netzwerken bisher wenig erschlossenes Themengebiet, wie Fogues et al. [2015] zeigten. Sie definierten fünf Anforderungen an ein Zugriffsmodell für SNS (vgl. [Fogues et al. 2015, 3 f.]):

- Beziehungsbasiert
- Feingranular
- Interoperabilität
- Angeheftete Richtlinien an Daten
- Co-Privacy (Co-Datenschutz)

Weiterhin legten sie vier Anforderungen für einen Datenschutzmechanismus fest (vgl. [Fogues et al. 2015, 4 f.]):

- Automatisches Beziehungsableiten
- Datenschutzeinstellungsempfehlung
- Datenschutzverständlichkeit
- Selbstdarstellungsmanagement

Hierbei zeigen sich aktuelle Herausforderungen, welche in ein zukünftig zu entwickelndes Informationssystem einfließen. Von besonderer Bedeutung sind hierbei an Daten angeheftete Richtlinien sowie Co-Privacy. Der Zugriff auf Daten erfolgt zumeist durch eine Zugriffskontrolle. Diese kann unter anderem rollen-basiert ausgestaltet werden. Problematisch an diesem Ansatz, wie bei jedem anderen Ansatz auch, ist die Komplexität und die

damit verbundene Konfliktwahrscheinlichkeit. Ni et al. [2010] stellten fest, je höher die Komplexität von Sicherheits- und Datenschutzrichtlinien, desto höher die Wahrscheinlichkeit von Konflikten [Ni et al. 2010, 13]. Dies hat zur Folge, dass der Datenschutz durch eine hohe Komplexität gefährdet wird. Daraus ergibt sich die Anforderung, die Zugriffskontrolle möglichst einfach, intuitiv und mit Hilfestellung für den Anwender umzusetzen.
Aus den zuvor besprochenen Themengebieten ergeben sich erste Anforderungen an ein zu entwickelndes Informationssystem. (vgl. Tabelle 13)

Anforderungen	
Beschreibung	**Quellenbezug**
Datenschutz	
Datenschutz muss für den Nutzer deutlich hervorgehoben werden	[Tsai et al. 2011]
Hohe Kontrolle über die Daten mit der Option der (automatischen) Löschung	[Karla 2010]
Einfaches Rechtemanagement, um Konflikte zu vermeiden	[Ni et al. 2010]
Anonymität der Nutzer	[Tigelaar et al. 2012]
Der Schutz der persönlichen Daten sollte durch den Nutzer bzw. dessen Datenverwalter erfolgen	[Williams/Sion 2013] [Erway et al. 2015]
Angeheftete Schutzrichtlinien	[Fogues et al. 2015]
Co-Datenschutz (Co-privacy)	[Fogues et al. 2015]
Verständlichkeit des Datenschutzes	[Fogues et al. 2015]
Empfehlung für Datenschutzeinstellung	[Fogues et al. 2015]
Vertrauen	
Zentralisierung des Vertrauens	[Sarker et al. 2011]
Es muss Vertrauen beim Nutzer erzeugt werden (z. B. mit Hilfe von anderen Nutzern)	[Liu et al. 2015]
Beziehungsmanagement	
Unterstützung beim Kontaktmanagement	[Davison et al. 2013]
Automatisch Beziehungen ableiten	[Fogues et al. 2015]
System	
Quellenvielfalt	[Tigelaar et al. 2012]
Verfügbarkeit von Daten	[Tigelaar et al. 2012]
Robustheit des Systems	[Tigelaar et al. 2012]
Feingranular	[Fogues et al. 2015]
Interoperabilität	[Fogues et al. 2015]
Auf Beziehungen basierend	[Fogues et al. 2015]

Performante Suche im Netzwerk	[Hosanagar et al. 2010]
Selbstdarstellungsmanagement (Monitoring/Feedback)	[Fogues et al. 2015]

Tabelle 13: Übersicht ermittelter Anforderungen aus systematischer Literaturanalyse

Damit die ersten extrahierten Anforderungen vervollständigt werden können, ist es zielführend, derzeitige Systeme und Ansätze andere Forscher zu analysieren und deren Lösungen gegenüberzustellen. Die nachfolgende Untersuchung alternativer Lösungsansätze von sozialen Netzwerken ergibt sich aus den hohen Datenschutzansprüchen und der Forschungslandschaft, die sich genau in diesem Bereich mit diesem Thema befasst. Im wirtschaftlichen Sinne ist es ein Vergleich des Marktes und der aktuellen Konkurrenzprodukte.

Auf abstrakter Ebene werden Akteure durch Technologie verbunden, um einen Mehrwert zu generieren. Dieses Modell kann auf soziale Netzwerke im Internet übertragen werden. Hierbei ist der Mehrwert die Informationserzeugung, die Akteure sind die Nutzer. Bezogen auf soziale Aktivitäten fand in den letzten Jahren eine Zentralisierung des Systems statt. Damit verbunden entstanden Abhängigkeiten und Probleme in Bezug auf Datenschutz.

Entgegen dieser Entwicklung waren es unter anderem Yeung et al. [2009], die die Zukunft der sozialen Netzwerke in der Dezentralisierung sehen. Hierfür geben sie drei Empfehlungen solcher Systeme an (vgl. [Yeung et al. 2009, 2]):

- **Datenschutz:** Kontrolle durch den Nutzer über Daten, welche gezeigt werden, und darüber, welche Restriktionen gelten
- **Besitz:** Nutzer haben den kompletten Besitz über ihre Daten, auch wenn ein Anbieter seine Dienstleistung abschaltet
- **Verbreitung:** Der Nutzer entscheidet selbst über die Verbreitung seiner Daten, welche anhand von Präferenzen definiert sind und auf Beziehungen und Freundschaften basieren

Weiterhin entwickelten sie ein Framework für dezentral organisierte Online Social Networks. (vgl. Abbildung 23)

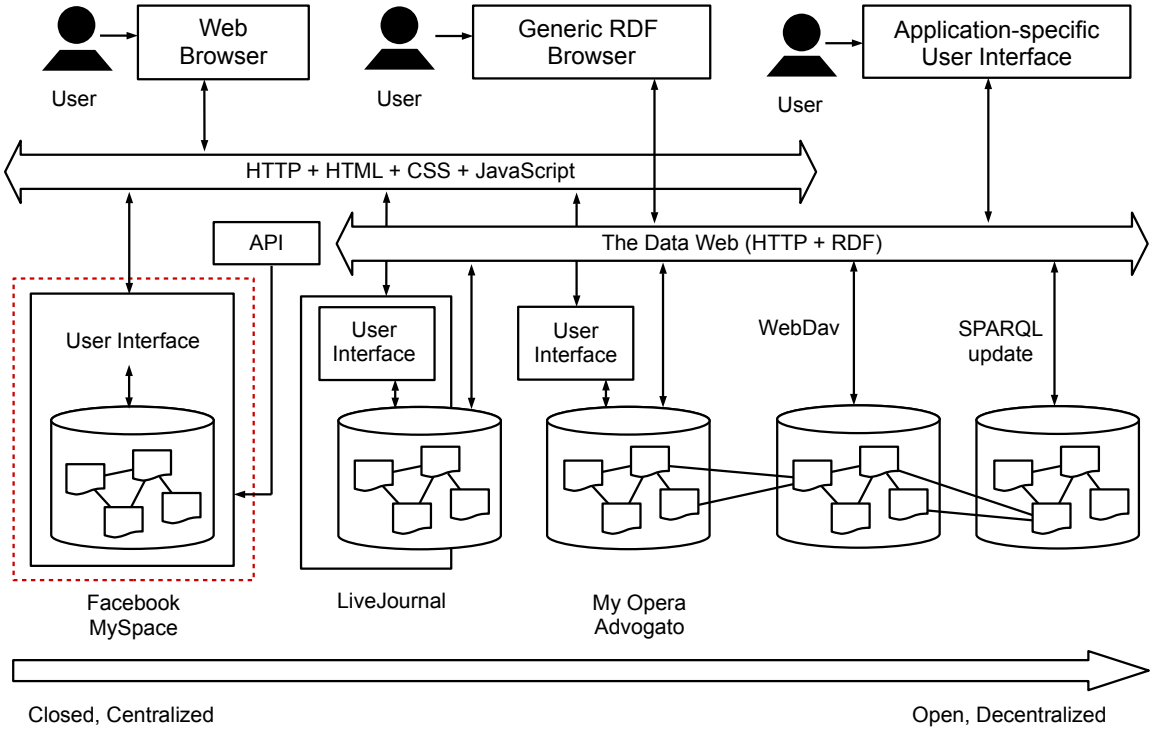


Abbildung 23: Framework für Online Social Networking [Yeung et al. 2009, 2]

Bei diesem Konzept ist zu erkennen, dass eine Entwicklung von geschlossenen, zentralen hin zu offenen, dezentralen Systemen vorausgesagt wird. Weiterhin werden verschiedene Datenquellen einbezogen. Dies hat zur Folge, dass der Nutzer einbezogen wird und er seine Daten selbst auf eigens konfigurierten Systemen zur Verfügung stellen muss. Dies wird nicht jedem Nutzer möglich sein. Ein weiteres Konzept welches zukünftig Einfluss nehmen wird, ist laut Yeung et al. das Semantic Web.

Heutzutage werden dezentrale Architekturen als natürliche Antwort auf zentrale Systeme gesehen (vgl. [Narayanan et al. 2012, 1]). Jedoch zeigt die Praxis, dass solche Systeme nur eine gering Einführung in den Markt haben. Nach langjähriger und intensiver Forschung in diesem Bereich wurden verschiedene Lösungsstrategien entwickelt. Diese können aus heutiger Sicht in die Bereiche vereinte (eng.: federated) und verteilte (eng.: distributed) Netzwerke eingeteilt werden. Vereinte Netzwerke zeichnen sich durch den Zusammenschluss von Nutzern in Gruppen aus, welche ein Ökosystem von Implementierungen bilden. Verteilte Systeme nutzen die Peer-to-Peer-Technologie. Vertrauensvolle Intermediäre könnten hierbei eine wichtige Rolle spielen, sind derzeit aber nicht daran interessiert sich zu beteiligen. (vgl. [Narayanan et al. 2012, 1 f.])

Narayanan et al. [2012, 3] setzten Charakteristiken fest, welche den Erfolg einer neuen Technologie in diesem Bereich positiv beeinflussen:

- Privatsphäre
- Nutzbarkeit
- Kosten
- Innovation

Gerade im Bereich Nutzbarkeit und Kosten haben die aktuellen Lösungsansätze noch Nachholbedarf, da die entwickelte Software meist ein hohes technisches Knowhow erfordert. Um zukünftige Entwicklungen positiv zu beeinflussen, geben Narayanan et al. [2012, 5] sieben Empfehlung für die Entwicklungen dezentraler Systeme:

1. Betrachten Sie die Wirtschaftlichkeit Ihrer Konstruktion
2. Achten Sie auf konzeptionelle Treue
3. Integrieren Sie andere Vorstellungen von Regulierbarkeit
4. Dem Nutzer mehr Vorteile als nur die Privatsphäre bieten
5. Gestalten Sie mit dem Fokus auf Standardisierung
6. Ziel ist ein beschränkter Funktionsumfang
7. Zusammenarbeit mit Aufsichtsbehörden

Grundsätzlich ist davon auszugehen, dass dezentrale soziale Netzwerke für den Konsumenten höhere Kosten bedeuten, da die Finanzierung der Plattform nicht mehr durch die gesammelten Daten der Nutzer geschieht (vgl. [Shakimov et al. 2009, 13]). Nach Shakimov et al. gibt es drei Realisierungsformen basierend auf P2P: Cloud, Desktop, Hybrid. (vgl. Abbildung 24)

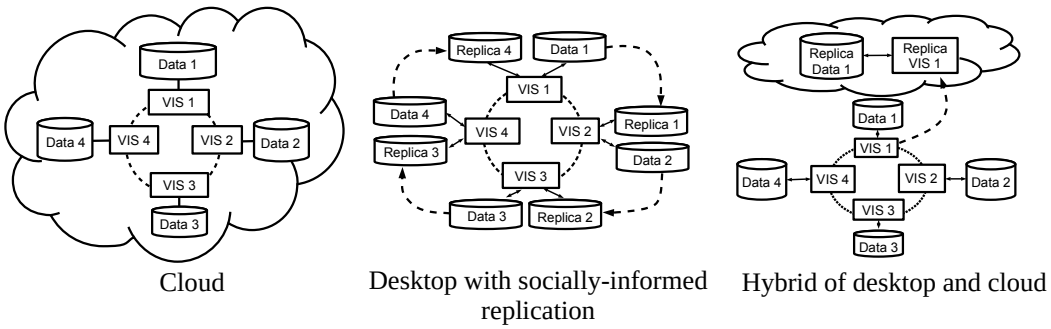


Abbildung 24: Realisierungsformen von sozialen Netzwerken basierend auf P2P [Shakimov et al. 2009, 13]

Ein weiteres Konzept dem Nutzer mehr Datenhoheit zu übertragen, ist die Integration einer Storage Cloud in ein bestehendes soziales Netzwerk. Chard et al. [2012] nutzen hierfür Facebook, um Daten nach dessen Rechtemanagement mit anderen Nutzern zu tauschen. Ergänzt wird dieser Ansatz mit einer konzeptionellen Erweiterung zu einem sozialen Marktplatz für Daten (vgl. [Chard et al. 2012, 553]).

Nachfolgend werden acht Systeme, die sich der Problematik Dezentralisierung in sozialen Netzwerken angenommen haben, kurz vorgestellt und auf ihre derzeitigen offenen Probleme und Fragestellungen hin betrachtet. Abschließend erfolgt ein tabellarischer Vergleich basierend auf unterschiedlichen Ausprägungen von Eigenschaften der Systeme.

PeerSoN von Buchegger et al. [2009] ist ein Peer-to-Peer-Ansatz, welcher einen Fokus auf Privacy legt. Um die Daten der Nutzer zu schützen, wird eine Verschlüsselung nach dem Public-Key-Verfahren durchgeführt. Der Datenzugriff kann so nur bei vorhandenem Schlüssel erfolgen. Grundsätzlich sind alle Daten auf den jeweiligen lokalen Computern der Anwender gespeichert. Ein Look-up-Service hilft beim Finden von Nutzern und bei der Interaktion. Ist ein Nutzer nicht online, können keine Daten aktualisiert werden. Das Problem der Verfügbarkeit von Daten kann gelöst werden, indem Freunde die eigenen Daten zwischenspeichern. Dies hat jedoch negative Auswirkungen auf den Datenschutz. Die Direktkommunikation findet über externe Applikationen statt.

Persona von Baden et al. [2009] ist ein weiterer Lösungsansatz, welcher einen zentralen Storage Service verwendet. Weiterhin wird auf ein attributbasiertes Verschlüsseln mit feingranularen Regeln gesetzt. Mit Hilfe einer Browser-Erweiterung kann es in einen bestehenden SNS integriert werden. Erste Performanzmessungen ergaben jedoch, dass das Laden sehr vieler Daten relativ lange dauert (bis zu 10 Sekunden).

Priv.io von Zhang und Mislove ist ein cloud-basierter Ansatz. Hierzu wurden zwei Komponenten, priv.io core und priv.io applications, entwickelt. Priv.io core ist eine Java-Anwendung, welche es erlaubt, auf Daten der Nutzer zuzugreifen und diese zu manipulieren. Ergänzend wird sie zur Kommunikation mit anderen Nutzern verwendet. Priv.io applications erlaubt die Verwendung weiterer Anwendungen in diesem Ökosystem. Grundsätzlich verwendet Priv.io eine attributbasierte Verschlüsselung. Auf jeder Cloud muss die priv.io-Anwendung als Webservice laufen. Es werden alle Daten von dem Cloud Provider gespeichert. Dadurch wird die Verfügbarkeit sicher gestellt und es steigen die Kosten.

PrPl von Seong et al. [2010] führen eine Softwarekomponente namens Personal Cloud Butler ein. Diese wird entweder vom Nutzer selbst betrieben oder durch einen Anbieter bereit gestellt. Dadurch ergeben sich unterschiedliche Datenschutzstufen, je nach dem, beim wem die Software betrieben wird. Für die Bildung eines Netzwerkes, kommunizieren die verschiedenen Instanzen der Butler untereinander. Weiterhin ist es möglich, Daten aus anderen Systemen anzubinden (z. B. aus Facebook). Dieses Konzept ist überwiegend dezentral organisiert, da jede Instanz von einem Anwender aufgesetzt werden muss, ohne dass es eine zentrale Einheit gibt.

Safebook von Cutillo et al. [2009] ist ein Dezentralisierungsansatz mit Real-life trust. Hierbei ist versucht worden, die Problematik des Vertrauens unter den Anwendern und dem System und deren Betreibern als solches zu lösen. Wie viele andere Lösungskonzepte wird für die Kommunikation und die Bildung des Netzwerkes P2P-Technologie eingesetzt. Realisiert wird die Verbindung über eine Matryoska-Architektur, die das Vertrauen der Nutzer untereinander prüft. Der Kommunikationsaufbau erfolgt über einen Social Network Server.

SlopPy von Gambs und Lolive [2013] ist ein Ansatz, verschlüsselte Daten auf sogenannten semi-trusted Instanzen zu speichern. Hierbei werden Daten an Freunde übertragen, können aber nur mit dem richtigen Schlüssel angesehen werden. Die Kommunikation läuft über ein anonymes Kommunikations-Netzwerk. Hier wird dem Problem der geringen Verfügbarkeit entgegnet.

SuperNova von Sharma und Datta [2012] ist ein P2P-Lösungsansatz mit Super-Peers. Hierbei übernehmen Freunde die Speicherung der eigenen Daten, um eine hohe Verfügbarkeit zu gewährleisten. Sogenannte Storekeepers haben Schlüsselaufgaben inne und halten das Netzwerk in Betrieb.

Vis-à-Vis von Shakimov et al. [2011] ist ein umfangreiches Konzept eines dezentral cloudbasierten sozialen Netzwerkes. Virtual Individual Server (VIS) werden von Nutzern entweder selbst betrieben oder bei einem Cloud-Provider angemietet. Diese VIS bestehen aus einer Speicherschicht und einer Verarbeitungsschicht und kommunizieren untereinander. Auf diesem Weg werden die Daten der Nutzer ausgetauscht. Das System ist lokal- und gruppenbasiert und vergleichbar mit Diaspora*.

Nachdem bisherige Lösungsansätze kurz vorgestellt wurden, findet im nächsten Schritt ein Vergleich basierend auf vier Eigenschaftskategorien statt: Architektur, Performanz, Sicherheit/Datenschutz und Nutzen.

Der Bereich Architektur vergleicht zunächst die Systeme im Hinblick auf ihren Einsatz von P2P und auf ihre Dezentralität im Allgemeinen. Darauf aufbauend wird die Möglichkeit das System in einer Cloud-Umgebung zu betreiben, untersucht. Von besonderem Interesse ist die Verteilung von Daten auf andere Teilnehmer, da hier ein potentielles Datenschutzrisiko besteht. Die Umsetzung eines Messaging-Systems (Chat) wird tangiert, da Direktkommunikation für Interaktionen in einem sozialen Netzwerk sehr wichtig ist. Die Kategorie Performanz bewertet die Verfügbarkeit sowie Übertragungsqualität der Daten. Sicherheit und Performanz betrachten die Verschlüsselung und die Umsetzung des Datenschutzes. Die Kategorie Nutzen zielt auf die eigentliche Verwendung in der Praxis ab und stellt dar, wie hoch für den Anwender die Hürden sind, das vorgestellte System einzusetzen. Tabelle 14 zeigt den Vergleich der Systeme tabellarisch.

Name	Autor(en)	Jahr	Architektur					Performanz			Sicherheit/ Datenschutz			Nutzen		
			P2P	Dezentral	Cloudbasiert	Datenverteilung auf Teilnehmer	Messaging	Verfügbarkeit der Daten	Datentransfer	Load and Process Delay	Verschlüsselung	Daten-Caching	Beziehungsmanagement	Kosten	Komplexität	Prototyp
PeerSoN	Buchegger et al.	[2009]	●	◕	○	◑	○	◑	◕	◌	●	◑	○	◌	●	◑
Persona	Baden et al.	[2009]	○	○	○	○	◕	●	◑	◕	◕	○	○	◌	●	◑
Priv.io	Zhang & Mislove	[2013]	●	●	●	○	○	◕	◕	●	●	○	○	◕	●	◑
PrPl	Seong et al.	[2010]	○	◕	◑	◕	◌	◕	●	◑	◑	◕	◕	◕	◕	●
Safebook	Cutillo et al.	[2009]	◕	◕	○	◑	◌	◑	●	◌	◕	○	◑	◕	●	◑
SlopPy	Gambs & Lolive	[2013]	○	◕	○	●	○	●	◕	◑	●	◑	○	◑	●	◑
SuperNova	Sharma & Datta	[2012]	●	●	○	◕	○	◕	●	◑	◕	●	◑	◌	●	◑
Vis-à-Vis	Shakimov et al.	[2011]	○	●	◕	●	○	◑	◑	◑	○	◌	○	◕	◕	◑

◌ Keine Angabe

○ Nicht / Keine

◑ Teilweise / Mittel

◕ Überwiegend / Hoch

● Vollständig / Sehr Hoch

Tabelle 14: Systemvergleich dezentraler Informationssysteme (Soziale Netzwerke)

Es ist zu erkennen, dass Architekturen entweder komplett auf Peer-to-Peer-Technologien aufbauen oder komplett darauf verzichten. Das Verhältnis beträgt hierbei 50 %. Dezentralisierung ist der Hauptlösungsansatz bei diesen Systemen. Dieses Ergebnis überrascht nicht, da die Suchanfrage darauf abzielte, solche Umsetzungen zu finden. Lediglich Persona, welches als Teilkonzept gedacht ist, präferiert die Zentralisierung. Weiterhin ist zu erkennen, dass nur wenige Ansätze cloud-basiert sind. Die Umsetzung sicherer Cloud-Lösungen ist sehr aufwändig und stellt grundsätzlich ein höheres Sicherheitsrisiko dar. Eines der größten Probleme dezentraler Systeme ist die Sicherstellung der Verfügbarkeit. Ein überwiegender Anteil an Systemen versucht dieses zu lösen, indem Daten auf anderer Teilnehmer teilweise oder komplett ausgelagert werden. Dies geschieht auf Kosten des Datenschutzes und der Datensicherheit. Ein sehr wichtiges Element der Interaktion zwischen den Nutzern in Form von Direktkommunikation (Messaging) wird von fast allen Systemen nicht betrachtet.

Die Auswertung der Kategorie Performanz zeigt, dass die Verfügbarkeit von Daten bei allen Umsetzungen relativ hoch ist. Dies stellt beim Entwerfen von Konzepten eine der Hauptanforderungen dar und ermöglicht erst ein funktionsfähiges System. Weiterhin zeigt sich, dass sehr hohe Datentransferraten entstehen, wenn ein System nicht zentralisiert gestaltet ist. Mit den heutigen Internettechnologien ist dieser Overhead (deutsch.: *allgemeine Unkosten*) an Übertragung jedoch zunehmend handhabbar. Es kann angemerkt werden, dass diese Systeme erst durch den technischen Fortschritt in diesem Bereich realisierbar sind. Durch den Einsatz von Dezentralisierung zeigt sich, dass es zwangsläufig zu Lade- und Prozessverzögerungen kommt. Dies ist in diesem speziellen Anwendungsbereich jedoch vernachlässigbar, da keine Echtzeit gefordert wird. Ein Maximum an Verzögerungszeit ist jedoch dringend geboten, damit die Nutzererfahrung mit der Anwendung nicht negativ beeinflusst wird.

Der zentrale Aspekt dieser Arbeit und die darauf aufbauende systematische Literaturanalyse ist die Steigerung des Datenschutzes dezentraler Systeme, einhergehend mit Sicherheitsaspekten. Verschlüsselung wird sehr häufig und intensiv eingesetzt. Oft basieren Konzepte lediglich auf einer ausgeklügelten Sicherheitstechnologie. Beispiele sind die Matryoska-Architektur oder das Real-Life-Trust. Für die Steigerung der Verfügbarkeit von Daten wird teilweise Caching eingesetzt, welches aber ein Sicherheitsrisiko darstellt. Weiterhin ist festzustellen, dass Beziehungsmanagement, welches dem Nutzer hilft, nur vereinzelt als Ele-

ment umgesetzt ist.

Basierend auf den Zielstellungen der Wirtschaftsinformatik muss ein Konzept sich der Realität und damit der Realisierung stellen. Daher wurden die Systeme auf ihre Nutzbarkeit hin verglichen. Ein wesentlicher Aspekt in diesem Bereich sind die entstehenden Kosten für die Nutzung. Es zeigt sich, dass Systeme, die Kosten angeben, sehr hohe Kosten auszeichnen. Weiterhin ist die Komplexität bei allen Systemen enorm. Dies liegt meist daran, dass der Nutzer eine eigene Anwendung selbst aufsetzten und betreiben muss. Dies ist für den „normalen“ Nutzer nicht möglich. Erschwert wird dies weiterhin dadurch, dass die Systeme sich zumeist in einer Betaphase befinden und die Konfiguration sehr aufwändig ist. Auch der Umgang mit den fertigen Systemen gestaltet sich schwierig. Wenn ein System nicht verfügbar ist, findet keine Interaktion mit anderen Nutzern statt. Es zeigt sich weiterhin, dass alle Entwicklungen Prototypen verwenden, um die Funktionsweise oder Teilaspekte zu verdeutlichen.

Nachfolgend werden aus der Analyse Zusammenhänge dargestellt und Implikationen für ein zu entwickelndes Informationssystem abgeleitet. Eine zentrale Anforderung dezentraler Systeme ist die Gewährleistung der Verfügbarkeit von Daten. Dies kann entweder durch eine Datenverteilung auf die Instanzen anderer Anwender erfolgen oder durch das Bereitstellen einer durchgehend lauffähigen Instanz. Dies kann zum Beispiel mit Hilfe eines Cloud-Providers erfolgen. Die eigenen Daten liegen somit entweder bei anderen Teilnehmern, bei einer Gruppe von Teilnehmern, bei einem Unternehmen oder im Cache des Systems. Prinzipiell ist der Betrieb eines eigenen Servers kompliziert und teuer. Es zeigte sich weiterhin, dass der Datentransfer bei allen Systemen sehr hoch ist, da sie nicht zentralisiert verwaltete Daten einsetzen und diese ständig versendet werden müssen. Dadurch entstehen Wartezeiten und Verzögerungen. Bei sozialen Netzwerken ist dies nicht zwangsläufig ein Problem, da keine Echtzeit gefordert ist. Durch Verschlüsselung versuchen die meisten Ansätze, den Datenschutz zu erhöhen. Dies verdeutlicht, dass die Konzepte Lücken für Angriffe aufweisen, zum Beispiel bei der Herangehensweise Daten bei „Freunden“ zu speichern. Eine weitere Erkenntnis ist, dass das Rechtemanagement, realisiert durch ein Beziehungsmanagement, positiv beeinflusst werden kann. Die Kosten für alle Systeme sind sehr hoch und die Anwendung dieser sehr komplex. Alle Systeme sind eher für technisch versierte Anwender geeignet, nicht für einen normalen Nutzer, der vielleicht eine Alternative sucht. Alle Lösungen bieten einen Prototypen an, der aber nicht

alle Anforderungen umsetzt. Dadurch kann kein gesamtes Ökosystem getestet werden, sondern nur Teilaspekte. Aus den analysierten Implikationen werden nachfolgend Konzeptanforderungen extrahiert. (vgl. Tabelle 15)

Konzeptanforderungen (B) Beschreibung, (A) Abgeleitete Anforderung	
B	Zentralisierung durch eine vertrauenswürdige Instanz, kann viele Probleme des absoluten Nicht-Trauens der Teilnehmer untereinander lösen
A	Zentrale Instanz als „Trusted Party“
B	Geringe Komplexität ist erforderlich, um eine breite Masse an Nutzern anzusprechen
A	Einsatz bestehender Dienstleistungen, ohne aufwändige Konfiguration
B	Nur kostengünstige Ansätze können Nutzer zum Wechsel bewegen
A	System ohne Kosten/variable Kosten
B	Die Wahl der Datenschutzhöhe muss vom Nutzer selbst bestimmt werden
A	Alternative Auswahl an Datenschutzhöhen
B	Verschlüsselungen sind komplex und nur sinnvoll, wenn notwendig
A	Möglichkeit der Verschlüsselung anbieten
B	Beziehungsmanagement hilft dem Nutzer bei Rechtemanagement
A	Integration Beziehungsmanagement
B	Ladezeiten können verzögert sein bis zu einem gewissen Maximum
A	Ladezeiten minimieren bis zu einem Punkt n
B	Die Verfügbarkeit der Daten muss gewährleistet sein
A	Vollständige Verfügbarkeit der Ressourcen
B	Ein Messaging-System (Chat) ist für eine Gesamtlösung notwendig
A	Integration Chat-System

Tabelle 15: Extrahierte Konzeptanforderungen aus Systemvergleich

Nachdem in diesem Abschnitte eine Analyse und Synthese der Inhalte erfolgte, wird im nächsten Kapitel die Einordnung der Ergebnisse in die gesamte Arbeit vollzogen.

4.1.7 Bewertung und Implikation der Ergebnisse

Mit der systematischen Literaturanalyse konnte gezeigt werden, dass bisher nicht gelöste Herausforderungen in der Wissenschaft bezüglich dezentraler Informationssysteme und Datenschutz bestehen. Daraus abgeleitet wurden erste Anforderungen generiert. In dieser Untersuchung ergaben sich insgesamt 30 Anforderungen in fünf Bereichen (Konzept, Datenschutz, Vertrauen, Beziehungsmanagement und System). Grundsätzlich ist fraglich, ob ein Konzept eines Informationssystems alle angegeben Anforderungen erfüllen kann. Eine eventuelle Teilerfüllung ist dennoch ein Fortschritt. Im nächsten Abschnitt werden die extrahierten Anforderungen konsolidiert, kategorisiert und konkretisiert.

Andersson et al. [2013] unterscheiden vier Archetypen des Peer-to-Peer-Austausches. Durch die Analyse verschiedener Peer-to-Peer-Lösungsansätze zeigte sich, dass diese Einteilung um ein weiteres Element, dem Peer-to-Peer „Information sharing“ (deutsch: *Informationsaustausch*) erweitert werden kann. (vgl. Tabelle 16) Hierbei muss festgestellt werden, dass es sich bei dem Dateiaustausch (engl.: *File sharing*) um eine Unterkategorie des Informationsaustausches handelt. An das information sharing wird keine zeitliche Anforderung gestellt. Es muss dennoch beachtet werden, dass Informationen im Laufe der Zeit an Wert verlieren können.

	File sharing	Trading	Goods sharing	Service sharing	Information sharing
Object of exchange	Digital material	Tangible m aterial	Tangible material	Intangible encounter	Digital material
Timing requirement	No	Not necessarily	Not necessarily	Yes	No
Meeting requirement	No	No	Not necessarily	Yes	No
Example	Napster	eBay	AirBnb	Avego	Diaspora*

Tabelle 16: Fünf Archetypen des Peer-to-Peer-Austausches (in Anlehnung an [Andersson et al. 2013, 3])

Insgesamt wird festgestellt, dass Dezentralisierung zumeist direkt mit P2P in Verbindung gebracht wird. Daher ist eine Einordnung von P2P in eine grundlegende Kategorisierung zielführend. Dies eröffnet die Möglichkeit, weitere Lösungsansätze zu konzipieren. Einhergehend mit der Beantwortung der Unterforschungsfrage 1 (UFF 1) findet eine Einordnung von Dezentralisierung auf technischer und organisatorischer Ebene statt. Die systematische

Literaturanalyse wird hierbei als Methodik verwand, um ein Artefakt der Kategorie Modell zu erzeugen.

Die Arten der Dezentralisierung unter technischen Gesichtspunkten können im Kontext verteilter Systeme untersucht werden. Aktuelle Realisierungen und Techniken wurden bereits umfangreich katalogisiert und beschrieben. Nachfolgend werden die Ausführungen von Schill [2012] verwandt. Die aktuelle Forschung ergänzt diese Sammlung um eine weitere Kategorie, das Fog Computing. Hierbei handelt es sich um eine riesige Menge an heterogenen, dezentralen, überall vorkommenden Endgeräten für Speicher- und Bearbeitungsaufgaben ohne Beteiligung Dritter, die Kommunizieren und Kooperieren. (vgl. [Vaquero/ Rodero-Merino 2014, 30]) Tabelle 17 gibt eine Übersicht zu verschiedenen Modellen und Ansätzen der technischen Dezentralisierung.

Arten der technischen Dezentralisierung
Client/Server-Modell *„ein Client [ruft] eine bestimmte Funktionalität bzw. Dienstleistung eines Servers über ein Rechnernetz hinweg auf, der diese Funktionalität zur Verfügung stellt.“* [Schill 2012, 14]
Objektorientiertes Modell *„Das objektorientierte Modell strukturiert Verteilte Systeme ebenfalls nach Art des Client/Server-Modells, die Einheiten der Kommunikation und Verteilung, die nun die Rolle des Dienstterbringers bzw. Dienstnutzers einnehmen, sind dabei jedoch Objekte beliebiger Granularität.“* [Schill 2012, 15]
Komponentenbasiertes Modell *„Die Grundidee dabei ist, dass die eigentliche Anwendungsfunktionalität weitgehend von den Eigenschaften der Verteilten Systeme getrennt wird.“* [Schill 2012, 18]
Dienstorientiertes Modell *„Dienste stellen grobgranulare Bausteine von Softwaresystemen dar, die in loser Kopplung zu komplexen Geschäftsprozessen und Abläufen in Unternehmen ebenso wie über Unternehmensgrenzen hinweg integriert werden können. Ähnlich wie Komponenten kapseln Dienste Funktionalität und Daten, die sie über eine wohldefinierte Schnittstelle zugreifbar machen, die Granularität ist jedoch für Dienste in der Regel höher.“* [Schill 2012, 20]

Mehrstufige Architekturen *„Die wesentlichen Stufen einer dreistufigen Architektur sind dabei die Benutzerschnittstelle und ggf. einige Vorverarbeitungsfunktionen auf dem Client (Präsentationsschicht), die Ebene der Anwendungslogik (Verarbeitungsschicht) mit der eigentlichen serverseitigen Verarbeitung sowie die Datenebene mit der Verwaltung persistenter Datenbestände, auf die serverseitig zugegriffen wird (Persistenzschicht).“* [Schill 2012, 23]
Grid Computing *„Grid Computing bezeichnet ein Konzept zur Aggregation und gemeinsamen Nutzung von heterogenen, vernetzten Ressourcen wie Rechnern, Datenbanken, Sensoren und wissenschaftlichen Instrumenten.“* [Schill 2012, 26]
Cloud Computing *„Ähnlich wie beim Grid Computing sollen Nutzern in der Cloud Ressourcen je nach deren Anforderungen bereitgestellt und wieder entzogen werden können.“* [Schill 2012, 31]
Fog Computing *Riesige Menge an heterogenen, dezentralen, überall vorkommenden Endgeräten für Speicher- und Bearbeitungsaufgaben ohne Beteiligung Dritter, die Kommunizieren und Kooperieren.* (vgl. [Vaquero/Rodero-Merino 2014, 30])
Peer-to-Peer-Architekturen *„Sogenannte Peers kommunizieren direkt miteinander und nutzen dabei gegenseitig Dienste bzw. stellen Dienste zur Verfügung. Während bei mehrstufigen Architekturen die Rollen von Client und Server getrennt und Systembestandteilen fest zugeordnet werden, erfolgt diese Trennung in Peer-to-Peer-Architekturen nicht.“* [Schill 2012, 36]

Tabelle 17: Arten der technischen Dezentralisierung

Nachdem die technischen Möglichkeiten der Dezentralisierung aufbereitet wurden, werden jetzt die organisatorischen Möglichkeiten der Dezentralisierung näher beleuchtet. Hierbei können Beteiligte verschiedene Rollen als Teilnehmer einnehmen. Von besonderer Bedeutung ist hierbei das Vertrauen gerade bei hoher Zentralisierung. Es zeigte sich bei der Systemanalyse, dass Lösungsansätze Peer-to-Peer favorisierten, da hierbei das Vertrauen gegenüber einer zentralen Instanz nicht notwendig war. Dies geschah auf Kosten der Verfügbarkeit von Daten. Nachfolgend werden die vier Stufen der organisatorischen Dezentralisierung aufgelistet. (vgl. Tabelle 18)

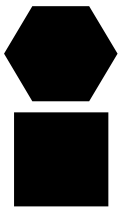
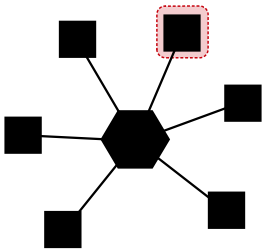
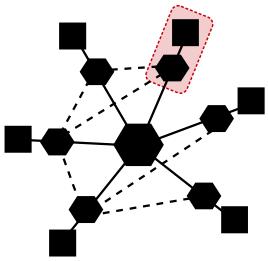
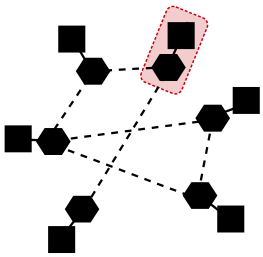
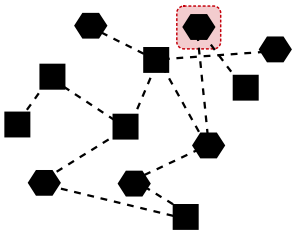
Modell der organisatorischen Dezentralisierung	
Abbildung	Stufe
	**Stufe 0: Keine (Zentral)** Vollständig zentral organisierte Anwendung. Beispiel: Facebook
	**Stufe 1: Dezentrale Daten** Daten der Anwender werden dezentral gespeichert. Beispiel: -
	**Stufe 2: Zentrales Management** Anwendungen mit Daten werden dezentral betrieben mit zentralem Management. Beispiele: Peer-to-Peer mit Super-Peer(s), WWW
	**Stufe 3: Dezentrale Knoten** Anwendungen mit Daten werden dezentral betrieben. Beispiele: Peer-to-Peer, Diaspora*
	**Stufe 4: Vollständig Dezentral** Alle Anwendungen und Daten werden dezentral gespeichert und betrieben. Beispiele: Fog Computing, Microservices
Legende: Softwareinstanz — Notwendige Verbindung — Muss Vertrauen Daten --- Mögliche Verbindung	

Tabelle 18: Arten der organisatorischen Dezentralisierung

Insgesamt wird das Modell der organisatorischen Dezentralisierung in fünf Stufen eingeteilt, wobei die erste Stufe (Stufe 0) keine Dezentralisierungsaspekte beinhaltet. Der Übergang zu jeder Stufe ist fließend, dennoch werden bestehende Systemumsetzungen genau einer Stufe zugeordnet. Mit steigender Stufe steigt auch die Höhe der Dezentralisierung. Nachfolgend werden alle Stufen näher erläutert.

Stufe 0 zeichnet sich durch vollständig zentral organisierte Anwendungen und Dienstleistungsangebote aus. Dieser Ebene können die meisten aktuellen Plattformen des Internets zugeordnet werden. Dies liegt vor allem daran, dass die Leistungsangebotserstellung unabhängig von den Teilnehmern erfolgen kann. Damit ist es möglich, ein System einfach und ohne Restriktionen umzusetzen. Weiterhin besteht der Vorteil für Unternehmen, Daten als Wirtschaftsgut innerhalb der Anwendung zu halten. Einhergehend mit diesem Konzept gibt es keine Abgabe von Verantwortlichkeiten, Rechten und Pflichten an Dritte.

In **Stufe 1** erfolgt die erste Dezentralisierung in Form von Externalisierung von Nutzerdaten. Für eine konzeptionelle Umsetzung eignet sich die Cloud Computing-Technologie Storage Cloud, die eine einfache Integration der Datenspeicher von Nutzern in das System ermöglicht. Der zentral agierende Anbieter übernimmt hierbei sowohl die Rolle einer vertraulichen Instanz als auch das Management der Daten, behält selbst aber keine Daten der Nutzer.

Stufe 2 zeichnet sich aus durch die Auslagerung von Anwendungen und Daten hin zu den Teilnehmern mit einem zentralen Management. Bei sehr vielen Systemen wird für die Verbindung der einzelnen Knoten eine zentrale Registry verwendet, um die Teilnehmer miteinander zu verbinden. Das serviceorientierte Paradigma ist ein typischer Vertreter dieser Organisationsform für die technische Realisierung. Weiterhin ist dieses Konzept im World Wide Web umgesetzt. Mit sogenannten Domain Name Servern werden zentral verwaltete Internetadressen auf Server umgeleitet. Hierbei ist zu beachten, dass dem zentralen Management ein hohes Vertrauen entgegengebracht werden muss.

In **Stufe 3** entfällt das zentrale Management des Netzwerkes und die Teilnehmer müssen sich für eine Zusammenarbeit und Interaktion selbst organisieren. Im Bereich der sozialen Netzwerke ist Diaspora* ein Vertreter, welcher dieses Konzept einsetzt. Hierbei ist zu beachten, dass die eigene Verwaltung der Knoten, nicht-technikaffine Nutzer vor eine schwierige Aufgabe stellt. Grundsätzlich folgt das Prinzip dem Peer-to-Peer-Ansatz.

Die letzte Ebene, **Stufe 4**, beschreibt eine vollständige Dezentralisierung aller Komponen-

ten, dass heißt sowohl Daten als auch Anwendungen werden getrennt von einander gespeichert bzw. betrieben. Dieses Konzept ist zu finden im Bereich Internet der Dinge und beim Fog Computing-Paradigma. Die Umsetzung dieser Stufe ist derzeit noch Gegenstand aktueller Forschung. Interessant ist hierbei, dass keinem zentralen Anbieter Vertrauen entgegengebracht werden muss, sondern dem gesamten Ökosystem an sich.

Jede der beschriebenen Stufen kann mit unterschiedlichen Arten technischer Dezentralisierung umgesetzt und betrieben werden. Es handelt sich somit um ein rein organisatorisches Modell. Diese Auflistung verschiedener Stufen der organisatorischen Dezentralisierung hilft bei der späteren Konzipierung eines Architekturmodells für ein dezentrales Informationssystem.

Aus den Erkenntnissen der systematischen Literaturanalyse, den Systemvergleich dezentraler Systeme aus dem Bereich sozialer Netzwerke und dem Modell der organisatorischen Dezentralisierung werden im nächsten Kapitel die Anforderungen an ein zukünftiges Informationssystem abgeleitet.

4.2 Anforderungsanalyse in dezentralen Cloud Networks

Die Anforderungsanalyse dient dem Aufstellen von Anforderungen für die Konzeption eines Informationssystems, welches dezentral ausgerichtet ist. Hindel et al. [2006, 41] geben eine umfassende Definition für eine Anforderungsanalyse:

Eine Anforderungsanalyse ist ein systematischer Prozess, um durch eine iterative und kooperative Problemanalyse Anforderungen zu finden. Die gefundenen Ergebnisse werden mit Hilfe von verschiedenen Notationen festgehalten, so dass eine Überprüfung des gewonnenen Problemverständnisses möglich ist.

Iterativ bedeutet in diesem Fall, dass es sich um einen sich wiederholenden Prozess der Anforderungsspezifikation handelt, kooperativ, dass mehrere Beteiligte involviert sind (vgl. [Hindel et al. 2006, 41]). Beide Eigenschaften treffen in dieser Arbeit nicht zu, da die Anforderungen nicht durch die Zusammenarbeit mit einem Kunden definiert werden. Vielmehr wurden sie aus der systematischen Literaturanalyse und dem Systemvergleich gewonnen. Bei der Anforderungsdefinition existieren im wesentlichen drei Problemfelder: vollständige Erfassung aller Anforderungen, eindeutige Formulierung der Anforderungen und konsistente bzw. widerspruchsfreie Formulierung (vgl. [Hindel et al. 2006, 40/41]). Die vollständige Erfassung ist abhängig von der Güte der Literaturanalyse und des Systemvergleiches. Hierbei wird auf die hohe Zahl an ausgewählten Journals für die Untersuchung hingewiesen. Durch den Vergleich von acht Systemen, welche dezentral organisiert sind, kann ein hoher Abdeckungsgrad festgehalten werden. Die Eindeutigkeit der Formulierung wird garantiert durch die Anwendung von Qualitätskriterien ebenso wie die Konsistenz und die Widerspruchsfreiheit.
Bei dieser Anforderungsanalyse ist darauf hinzuweisen, dass es sich im Sinne einer Machbarkeitsstudie um einen Proof of Concept handelt. Dieser kann daher nicht vollständig spezifiziert werden, wie es bei einem Kundenprojekt der Fall wäre. Vielmehr werden technische Vorkenntnisse sowie Erkenntnisse aus der Literatur verwendet.
Die Anforderungen sind zunächst in zwei Oberkategorien zu unterteilen: Funktionale und Nichtfunktionale Anforderungen. Funktionale Anforderungen sind definiert als „vom System bzw. von seinen Systemkomponenten bereitzustellende Funktionen oder ein bereitzustellender Service. Als Benutzeranforderung kann eine funktionale Anforderung sehr allgemein beschrieben sein. Als Bestandteil einer Spezifikation beschreibt eine funktionale An-

forderung detailliert die Eingaben und Ausgaben sowie bekannte Annahmen“ [Pohl 2008, 15] Der Autor Pohl [2008, 16] weißt eindringlich darauf hin, den Terminus nichtfunktionale Anforderungen nicht zu verwenden, da diese entweder unterspezifizierte funktionale Anforderungen oder Qualitätsanforderungen an das System darstellen.

Da es sich in dieser Arbeit um ein noch zu entwickelndes System, genauer um ein Architekturmodell handelt, werden nur Qualitätsanforderungen festgelegt. Qualitätsanforderungen beschreiben „qualitative Eigenschaften des Gesamtsystems, einzelner Funktionen oder Funktionsgruppen“ [Pohl 2008, 16]. Das Vorgehen für eine qualitativ hochwertige Anforderungsanalyse besteht in dieser Arbeit aus fünf Schritten:

1. Vergeben von eindeutigen Identifikationsnummern (ID) für jede Anforderung
2. Vergeben von eindeutigen ID für das jeweilige Qualitätskriterium
3. Prüfen der Anforderungen nach Qualitätskriterien
4. Eventuelle Anpassung der Anforderung
5. Erstellen eines gut spezifizierten Anforderungskataloges

Abbildung 25 zeigt eine schematische Darstellung der Anforderungsanalyse.

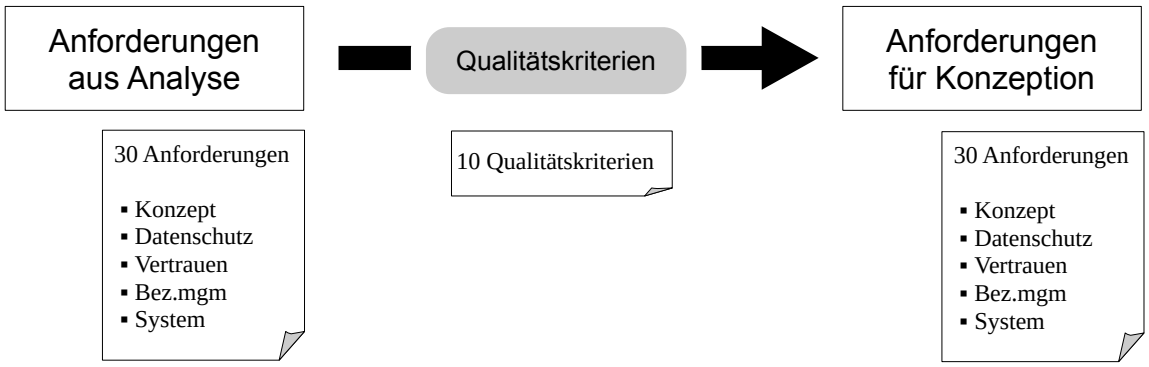


Abbildung 25: Vorgehen bei der Anforderungsanalyse.

Nachfolgend werden die Qualitätskriterien nach Pohl [2008] vorgestellt.

4.2.1 Qualitätskriterien

Qualitätskriterien dienen der hochwertigen Anforderungsdefinition. Jede Anforderung kann anhand dieser Kriterien analysiert und auf dessen Richtigkeit geprüft werden. In dieser Arbeit werden 30 Anforderungen definiert und jeweils mit den zehn Qualitätskriterien analysiert. Tabelle 19 gibt eine Übersicht sowie Beschreibung zu jedem Qualitätskriterium.

Qualitätskriterien für Anforderungen	
ID	Kriterium
RE1	**Vollständigkeit** Eine Anforderung ist vollständig, wenn die Anforderung gemäß den festgelegten Kriterien dokumentiert ist und keine inhaltlichen Lücken aufweist.
RE2	**Nachvollziehbarkeit** Eine Anforderung ist nachvollziehbar, wenn sowohl der Ursprung der Anforderung, die Evaluation der Anforderung als auch deren Einfluss auf die Realisierung des Systems nachverfolgbar sind.
RE3	**Korrektheit** Eine Anforderung ist korrekt, wenn die relevanten Stakeholder deren Korrektheit bestätigen und die Anforderung vollständig im zukünftigen System umgesetzt werden muss.
RE4	**Eindeutigkeit** Eine Anforderung ist eindeutig, wenn die Anforderung so formuliert ist, dass sie nur eine gültige Interpretation zulässt.
RE5	**Verständlichkeit** Eine Anforderung ist verständlich, wenn der Inhalt der Anforderung möglichst einfach erfasst werden kann.
RE6	**Konsistenz** Eine Anforderung ist konsistent, wenn die Aussagen einer Anforderung sich nicht widersprechen und nicht konfliktionär zueinander sind.
RE7	**Überprüfbarkeit** Eine Anforderung ist überprüfbar, wenn das erstellte System daraufhin überprüft werden kann, ob es die Anforderung erfüllt.
RE8	**Bewertet** Eine Anforderung ist bewertet, wenn die Bedeutung der Anforderung und/oder deren Stabilität ermittelt und dokumentiert ist.
RE9	**Aktualität** Eine Anforderung ist aktuell, wenn diese Anforderung die aktuellen Gegebenheiten des zu entwickelnden Systems und des zugehörigen Kontexts widerspiegelt […].
RE10	**Atomarität** Eine Anforderung ist atomar, wenn diese Anforderung einen isolierten Sachverhalt beschreibt.

Tabelle 19: Die zehn Qualitätskriterien für Anforderungen [Pohl 2008, 222/223]

Für die Überprüfung von Anforderungsgruppen wird eine tabellarische Übersicht erstellt, welche für jede Anforderung die Ausprägung der Qualität darstellt. Ein Qualitätskriterium für eine Anforderung kann hierbei erfüllt (●), teilweise erfüllt (◐) oder nicht erfüllt (○) sein. Tabelle 20 zeigt das Überprüfungsmuster von Qualitätskriterien für Gruppen.

ID	RE1	RE2	RE3	RE4	RE5	RE6	RE7	RE8	RE9	RE10
ID	○	○	○	○	○	○	○	○	○	○

Tabelle 20: Überprüfungsmuster von Qualitätskriterien für Gruppen

Die Ergebnisse der Qualitätserfüllung jeder Anforderung resultieren aus einer Überprüfung nach einem Überprüfungsprotokoll. Hierbei wird jedes Qualitätskriterium einzeln geprüft und bei einem Nichterfüllen beschrieben, welche Änderungen notwendig sind. Abschließend wird die Anforderung neu formuliert, um alle Kriterien zu erfüllen. Tabelle 21 zeigt das Überprüfungsmuster für eine Anforderung.

ID Beschreibung der Anforderung vor der Überprüfung		
ID		Erfüllung
RE1	○	*Beschreibung der Nicht-Erfüllung*
RE2	○	*Beschreibung der Nicht-Erfüllung*
RE3	○	*Beschreibung der Nicht-Erfüllung*
RE4	○	*Beschreibung der Nicht-Erfüllung*
RE5	○	*Beschreibung der Nicht-Erfüllung*
RE6	○	*Beschreibung der Nicht-Erfüllung*
RE7	○	*Beschreibung der Nicht-Erfüllung*
RE8	○	*Beschreibung der Nicht-Erfüllung*
RE9	○	*Beschreibung der Nicht-Erfüllung*
RE10	○	*Beschreibung der Nicht-Erfüllung*
Nach Beschreibung der Anforderung nach der Anpassung		

Tabelle 21: Überprüfungsmuster für eine spezifische Anforderung

Unter dem Kapitel A Anwendung der Qualitätskriterien sind alle Anforderungen detailliert überprüft worden und deren Ergebnisse dargestellt. Anschließend erfolgt die Darstellung der Transformation der Anforderung aus der systematischen Literaturanalyse und dem Systemvergleich, hin zu einem gut spezifizierten Anforderungskatalog. Hierbei wird unterteilt in: Konzeptanforderungen (4.2.2), Datenschutzanforderungen (4.2.3) und Systemanforderungen (4.2.4).

4.2.2 Konzeptanforderungen

Konzeptanforderungen dienen der Erstellung eines Grobkonzeptes sowie einer anschließend detaillierten Konzeption eines zukünftigen dezentralen Informationssystems mit dem Fokus Datenschutz. Sie beschreiben das Einhalten von ganz bestimmten Anforderungen für ein erfolgreiches Design eines Ansatzes. Für die flexible Gestaltung sind nicht alle An-

forderungen spezifiziert, dadurch kann ein Konzept entwickelt werden, welches dynamisch nicht spezifizierte Qualitätskriterien selbst ausfüllt.

Die Anforderungen sind mit einer eindeutigen Identifikationsnummer kategorisiert, welche das Kürzel CO (**Co**ncept) tragen, jeweils gefolgt von einer Nummer. Insgesamt wurden neun Konzeptanforderungen festgelegt. Tabelle 22 zeigt die vorläufigen Anforderungen, die es zu überprüfen gilt. Diese wurden aus der systematischen Literaturanalyse und dem Systemvergleich gewonnen.

Konzeptanforderungen (vorläufig)	
ID	Beschreibung
CO	**Konzept**
CO1	Zentrale Instanz als „trusted party“
CO2	Einsatz bestehender DL ohne aufwändige Konfiguration
CO3	System ohne Kosten/variable Kosten
CO4	Alternative Auswahl an Datenschutzhöhe
CO5	Möglichkeit der Verschlüsselung anbieten
CO6	Beziehungsmanagement integrieren
CO7	Ladezeiten minimieren bis zu einem Punkt x
CO8	Vollständige Verfügbarkeit der Ressourcen
CO9	Chat System Integration

Tabelle 22: Konzeptanforderungen vor Qualitätskontrolle

Tabelle 23 zeigt die Bewertung aller Anforderungen nach den 10 Qualitätskriterien von [Pohl 2008, 222/223].

Bewertete Konzeptanforderungen										
ID	RE1	RE2	RE3	RE4	RE5	RE6	RE7	RE8	RE9	RE10
CO1	●	●	●	◐	●	●	●	●	●	●
CO2	●	●	●	◐	◐	●	●	●	●	●
CO3	●	●	●	●	●	◐	●	●	●	●
CO4	●	●	●	●	●	●	●	●	●	●
CO5	◐	●	●	◐	●	●	●	●	●	●
CO6	●	●	●	●	●	●	●	●	●	●
CO7	○	●	●	◐	●	●	●	●	●	●
CO8	●	●	●	◐	◐	●	●	●	●	●
CO9	●	●	●	●	●	●	●	●	●	●

Tabelle 23: Bewertung der Konzeptanforderungen

Bei der Bewertung der Konzeptanforderungen ist festzustellen, dass es in den Bereichen Vollständigkeit (RE1), Eindeutigkeit (RE4), Verständlichkeit (RE5) und Konsistenz (RE6) Defizite gibt.

Die Vollständigkeit ist bei den Anforderungen CO5 und CO7 nicht gegeben. Diese müssen detaillierter beschrieben werden. Eine mögliche Nichteinhaltung wird später zu evaluieren sein, kann aber nur durch eine gute Spezifikation gelingen. Mehr als die Hälfte der Anforderungen war nicht eindeutig formuliert und wurden im Zuge der Überprüfung besser beschrieben. Im Bereich Verständlichkeit gab es bei den Anforderungen CO2 und CO8 Schwächen, welche behoben wurden. Die Konsistenz war bei der Anforderung CO3 nicht gegeben. Diese wurde geändert in „Grundsätzliche Verwendung des Systems ohne Kosten für den Nutzer“. Dadurch können versierte Nutzer, die einen höheren Datenschutz bevorzugen, eigene Investitionen (z. B. Für Server/Dienstleistung) tätigen. Weiterhin wurde die Variabilität entfernt, da diese sich aus der Konzeption ergibt.

4.2.3 Datenschutzanforderungen

Datenschutzanforderungen stellen einen bedeutenden Faktor bei der Konzeption des Informationssystems dieser Arbeit dar. Daher werden speziell hierfür Anforderungen in den Bereichen Datenschutz, Vertrauen und Rechtemanagement festgelegt. Primär stehen hierbei Anforderungen im Mittelpunkt, die dem Nutzer bzw. dessen Daten schützen. Weiterhin geht es um Anforderungen, die einen sicheren und geschützten Umgang mit diesen Daten erlauben.

Es existieren drei Kategorien von Anforderungen im Bereich Datenschutz:

- **Datenschutz** mit Identifikationsnummer DS (**D**aten**s**chutz) und jeweils einer Nummer
- **Vertrauen** mit Identifikationsnummer TR (**Tr**ust) und jeweils einer Nummer
- **Beziehungsmanagement** mit Identifikationsnummer RM (**R**ole **M**anagement) und jeweils einer Nummer

Datenschutz adressiert allgemeine Anforderungen im Bereich Schutz von benutzerbezogenen Daten. Vertrauen thematisiert das besondere Verhältnis von Nutzer und Anbieter einer Dienstleistung. Das Beziehungsmanagement ergibt sich aus der durchgeführten Literaturanalyse und dessen Implikation, das Rechtemanagement auf Beziehungen aufzubauen.

Alle vorläufig festgelegten Anforderungen sind in Tabelle 24 aufgelistet.

Datenschutzanforderungen (vorläufig)	
ID	Beschreibung
DS	**Datenschutz**
DS1	Datenschutz muss für den Nutzer deutlich hervorgehoben werden
DS2	Hohe Kontrolle über die Daten mit der Option der (automatischen) Löschung
DS3	Einfaches Rechtemanagement, um Konflikte zu vermeiden
DS4	Anonymität der Nutzer
DS5	Der Schutz der persönlichen Daten sollte durch den Nutzer bzw. dessen Datenverwalter erfolgen
DS6	Angeheftete Schutzrichtlinien
DS7	Co-Datenschutz (Co-Privacy)
DS8	Verständlichkeit des Datenschutzes
DS9	Empfehlung für Datenschutzeinstellung
TR	**Vertrauen**
TR1	Zentralisierung des Vertrauens
TR2	Es muss Vertrauen beim Nutzer erzeugt werden (z. B. mit Hilfe anderer Nutzer)
RM	**Beziehungsmanagement**
RM1	Unterstützung beim Kontaktmanagement
RM2	Automatisch Beziehungen ableiten

Tabelle 24: Datenschutzanforderungen vor Qualitätskontrolle

Tabelle 25 zeigt die Bewertung der Datenschutzanforderungen nach Qualitätskriterien. Anforderung DS4 besaß Schwächen im Bereich der Verständlichkeit, da nur über die allgemeine Anonymität der Nutzer gesprochen wurde. Dies wurde angepasst, und jetzt wird von Nutzern innerhalb des Systems gesprochen. DS7 (Co-Privacy) wurde nicht vollständig beschrieben (RE1). Co-Datenschutz bezieht sich nun auf gemeinsam erstellte Daten der Nutzer. DS8 wies Schwächen in der Vollständigkeit (RE1) und Eindeutigkeit (RE4) auf und wurde entsprechend umformuliert. In den Bereichen TR und RM sind die Anforderungen bereits hochwertig spezifiziert, da diese primär aus der Literatur stammen. Daher sind in diesem Bereich keine Anpassungen notwendig.

Die Bereiche Nachvollziehbarkeit (RE2), Korrektheit (RE3), Konsistenz (RE6), Überprüfbarkeit (RE7), Bewertet (RE8), Aktualität (RE9) und Atomarität (RE10) wurden von allen Anforderungen bereits umfassend erfüllt.

Bewertete Datenschutzanforderungen										
ID	RE1	RE2	RE3	RE4	RE5	RE6	RE7	RE8	RE9	RE10
DS1	●	●	●	●	●	●	●	●	●	●
DS2	●	●	●	◐	●	●	●	●	●	●
DS3	●	●	●	●	●	●	●	●	●	●
DS4	●	●	●	●	○	●	●	●	●	●
DS5	●	●	●	●	●	●	●	●	●	●
DS6	●	●	●	●	●	●	●	●	●	●
DS7	○	●	●	◐	●	●	●	●	●	●
DS8	◐	●	●	◐	●	●	●	●	●	●
DS9	●	●	●	●	●	●	●	●	●	●
TR1	●	●	●	●	●	●	●	●	●	●
TR2	●	●	●	●	●	●	●	●	●	●
RM1	●	●	●	●	●	●	●	●	●	●
RM2	●	●	●	●	●	●	●	●	●	●

Tabelle 25: Bewertung der Datenschutzanforderungen

4.2.4 Systemanforderungen

Systemanforderungen beschreiben allgemein das zu entwickelnde System. Die Anforderungen sind hierbei eher technisch orientiert und sollen den Grundcharakter der zukünftigen Konzeption verdeutlichen. Die Anforderungen werden kategorisiert mit einer Identifikationsnummer SY (**Sy**stem) und einer jeweilig eindeutigen Nummer. Tabelle 26 schlüsselt die vorläufigen Systemanforderungen auf.

Systemanforderungen (vorläufig)	
ID	Beschreibung
SY	**System**
SY1	Quellenvielfalt
SY2	Verfügbarkeit von Daten
SY3	Robustheit des Systems
SY4	Feingranular
SY5	Interoperabilität
SY6	Auf Beziehungen basierend
SY7	Performante Suche im Netzwerk
SY8	Selbstdarstellungsmanagement (Monitoring/Feedback)

Tabelle 26: Systemanforderungen vor Qualitätskontrolle

Tabelle 27 zeigt die Bewertung der vorläufigen Systemanforderungen nach Qualitätskriterien. Da gerade die ersten Anforderungen (SY1, SY2 und SY3) relativ kurz und allgemein gehalten wurden, mussten diese in ihrer Vollständigkeit angepasst werden. Weiterhin besaß Anforderung SY6 ein Defizit in der Eindeutigkeit, bezogen auf ein Beziehungsmanagement innerhalb der Konzeption. Dies wurde behoben und angepasst.

Bewertete Systemanforderungen										
ID	RE1	RE2	RE3	RE4	RE5	RE6	RE7	RE8	RE9	RE10
SY1	◐	●	●	◐	●	●	●	●	●	●
SY2	◐	●	●	◐	●	●	●	●	●	●
SY3	◐	●	●	●	●	●	●	●	●	●
SY4	●	●	●	●	●	●	●	●	●	●
SY5	●	●	●	●	●	●	●	●	●	●
SY6	●	●	●	◐	●	●	●	●	●	●
SY7	●	●	●	●	●	●	●	●	●	●
SY8	●	●	●	●	●	●	●	●	●	●

Tabelle 27: Bewertung der Systemanforderungen

Im nächsten Kapitel erfolgt der Zusammenschluss der Unterkategorien der Anforderungen in einen Anforderungskatalog. Hierbei werden die gewonnen Erkenntnisse durch die Qualitätsprüfung nach Qualitätskriterien einbezogen und die umformulierten und angepassten Anforderungen dargestellt.

4.2.5 Anforderungskatalog

Der Anforderungskatalog fasst alle gewonnenen, gesammelten und erstellten Anforderungen nach Kategorien tabellarisch zusammen. Hierbei werden fünf Kategorien gebildet: Konzept, Datenschutz, Vertrauen, Beziehungsmanagement und System.

Tabelle 28 zeigt den Anforderungskatalog mit jeweils einer ID und der dazugehörigen Formulierung. Auf jede Anforderung kann in Form einer Evaluation in späteren Teilen dieser Dissertation zurückgegriffen und deren Erfüllungsgrad bestimmt werden. Der Anforderungskatalog erlaubt die passgenaue Konzeption eines dezentralisierten Informationssystems. Er ist somit Bestandteil der Analysephase des Design Science Frameworks. Nachfolgend wird ein Konzept vorgestellt, welches der Konstruktion entspricht.

Anforderungskatalog	
ID	Beschreibung
CO	**Konzept**
CO1	System als zentrale Instanz des Vertrauens („trusted party“)
CO2	Am Markt angebotene Dienstleistungen verwenden, welche eine geringe Konfiguration für den Nutzer benötigen
CO3	Grundsätzliche Verwendung des Systems ohne Kosten für den Nutzer
CO4	Alternative Auswahl an Datenschutzhöhe
CO5	Möglichkeit der Datenverschlüsselung auf den Speichermedien anbieten
CO6	Beziehungsmanagement integrieren
CO7	Ladezeiten bis zum Anzeigen des Inhaltes minimieren bis unter 1 Minute
CO8	Vollständige Verfügbarkeit der notwendigen Ressourcen in Form von Daten
CO9	Chat System Integration
DS	**Datenschutz**
DS1	Datenschutz muss für den Nutzer deutlich hervorgehoben werden
DS2	Hohe Kontrolle über die Datenverwendung mit der Option der (automatischen) Löschung
DS3	Einfaches Rechtemanagement, um Konflikte zu vermeiden
DS4	Anonymität der Nutzer innerhalb des Systems
DS5	Der Schutz der persönlichen Daten sollte durch den Nutzer bzw. dessen Datenverwalter erfolgen
DS6	Angeheftete Schutzrichtlinien
DS7	Co-Datenschutz (Co-Privacy) bei gemeinsam erstellten Daten der Nutzer
DS8	Für den Nutzer einfach nachvollziehbar dargestellter Datenschutz
DS9	Empfehlung für Datenschutzeinstellung
TR	**Vertrauen**
TR1	Zentralisierung des Vertrauens
TR2	Es muss Vertrauen beim Nutzer erzeugt werden (z. B. mit Hilfe anderer Nutzer)
RM	**Beziehungsmanagement**
RM1	Unterstützung beim Kontaktmanagement
RM2	Automatisch Beziehungen ableiten
SY	**System**
SY1	Quellenvielfalt an Daten durch hohe Abstraktion der Zugriffsschicht
SY2	Konzept für die Verfügbarkeit und Nichtverfügbarkeit von Daten der Nutzer
SY3	Robustheit des Systems gegenüber Angriffen und fehlerhaften Daten
SY4	Feingranular
SY5	Interoperabilität
SY6	Rechte- und Interaktionsmanagement basierend auf Beziehungen

SY7	Performante Suche im Netzwerk
SY8	Selbstdarstellungsmanagement (Monitoring/Feedback)

Tabelle 28: Übersicht zu den qualitätsgesicherten Anforderungen

4.3 Zusammenfassung

In diesem Kapitel wurde eine Systematische Literaturanalyse durchgeführt. Hierbei lag der Fokus auf dezentralen Informationssystemen. Dies ermöglichte eine umfassende Auswertung des aktuellen Stands der Forschung. Trotz der hohen Auswahl an Journals gab es nur wenige Publikationen, die sich auf die Dezentralisierung konzentrierten. Erweitert wurde diese Betrachtung durch einen Systemvergleich von acht Anwendungen. Dadurch konnten aktuelle Lösungsstrategien im Bereich Systeme mit erhöhten Datenschutz aufgezeigt und verglichen werden. Im Fokus waren hierbei dezentral organisierte soziale Netzwerke.

Das primäre Ziel dieses Kapitels war die Erstellung eines Anforderungskataloges für die Beantwortung der Unterforschungsfrage 2 (UFF 2). Durch eine Anforderungsanalyse in dezentralen Cloud Networks gelang es, die zunächst einfach formulierten Anforderungen aus der Literatur in gut spezifizierte Qualitätsmerkmale umzuwandeln. Hierfür wurden Qualitätskriterien für jede Anforderung angewendet und gegebenenfalls Änderungen vorgenommen. Dies stellt im Sinne des Design Science Frameworks eine erste Evaluation dar. Es zeigte sich, dass eine direkte Übernahme von Anforderungen aus der Literatur nicht zielführend ist, da viele Anforderungen eine spezifischere Beschreibung benötigten, um diese zunächst umzusetzen und im zweiten Schritt an einem Konzept zu evaluieren

Neben der grundlegenden Beschreibung eines zu entwickelnden Informationssystems in Form von Anforderungen, gelang es, ein Modell für die organisatorische Dezentralisierung zu entwerfen. Dadurch konnte die Unterforschungsfrage 1 (UFF 1) beantwortet werden. Im nächsten Kapitel wird ein Konzept entwickelt, welches ausgehend von den aufgestellten Anforderungen für die Beantwortung der Unterforschungsfrage 3 (UFF 3) dient.

Teil II: Konzeptionelles Modell

Das konzeptionelle Modell als eines der drei Teile dieser Dissertation beschreibt den Übergang von der Analysephase hin zur Entwurfsphase der Design Science. In letzterer liegt der Fokus auf der Entwicklung von Lösungsstrategien für die Erfüllung zuvor aufgestellter Anforderungen. Die Abstraktion der Betrachtung einhergehend mit der Konzipierung des Lösungsansatzes sinkt im Laufe dieses Abschnittes kontinuierlich. So erfolgt zunächst die Entwicklung eines grundlegenden Konzeptes, welches die zentralen und richtungsweisenden Anforderungen erfüllt. Aus dem Konzept wird eine Grobarchitektur mit Grundkomponenten und deren Beziehungen untereinander entworfen. Anschließend erfolgt, basierend auf diesen Erkenntnissen, die modellgetriebene Konstruktion eines Architekturmodells. Eine detaillierte Beschreibung jeder Komponente ist, ebenso wie die Beschreibung der Interaktionen aller involvierten Mensch-Maschine-Elementen, Bestandteil der umfangreichen Ausführungen. Abschließend wird die modellierte Architektur erweitert durch neue technologische Methoden, um Performanz- und Sicherheitsaspekten gerecht zu werden.

5 Decentral Cloud Network

Ausgehend von den aufgestellten Anforderungen des vorhergehenden Kapitels wird in diesem Abschnitt ein Konzept für ein Informationssystem entwickelt. Die Konzeptanforderungen besitzen hierbei einen direkten Einfluss auf die Konzeptentwicklung.

Die Anforderung CO1, ein System als zentrale Instanz des Vertrauens zu konzipieren, schließt die Verwendung eines vollständig dezentralen Ansatzes aus. Als am Markt befindliche Dienstleistungen, die eine geringe Konfiguration für den Nutzer benötigen (vgl. CO2), bieten sich Storage Clouds als Datenspeicher an. Hierbei übernimmt der Anbieter die Verwaltung und den Betrieb des externen Festplattenspeichers. In Verbindung mit der Anforderung CO8, der vollständigen Verfügbarkeit der notwendigen Ressourcen, kann ein Zugriff auf die Daten der Nutzer jederzeit gewährleistet werden. Bezugnehmend auf die organisatorische Dezentralisierung und deren Stufen bieten die Rahmenbedingungen der Konzeptidee die Möglichkeit der Einordnung in das Modell. Stufe 1 positioniert einen Anbieter aus organisatorischer Sicht in die Mitte der Interaktionen. Dieser stellt Daten und Informationen für die Interaktionen zwischen den Beteiligten bereit. Hierbei werden alle Daten aus externen Quellen integriert, die der Anwender extern speichert und verwaltet. Diese Aufgabe kann sowohl der Nutzer erfüllen, sie kann aber auch an einen Dritten abgeben werden. (vgl. TR1) Abbildung 26 zeigt die Ausprägung der Stufe 1 der organisatorischen Dezentralisierung für das in dieser Dissertation vorgestellte Konzept.

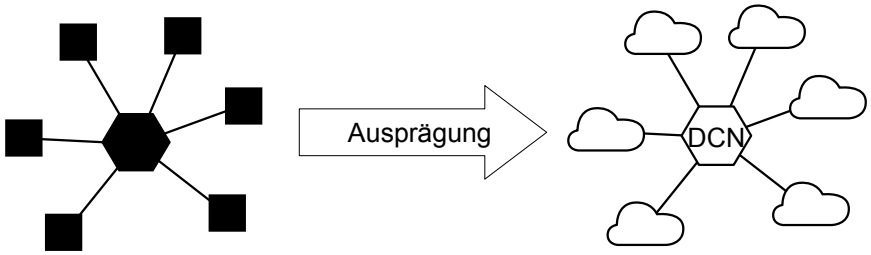


Abbildung 26: Ausprägung Stufe 1 der organisatorischen Dezentralisierung

In dieser Dissertation wird für dieses Lösungskonzept der Begriff des Decentral Cloud Networks (DCN) eingeführt. Die Wortwahl deutet auf die Verteilung von Systembestandteilen sowie den Einsatz einer Cloud Computing-Infrastruktur hin.

Der Begriff Decentral Cloud Network wird in dieser Dissertation wie folgt definiert:

Ein Decentral Cloud Network ist ein Informationssystem, welches seine gesamten Daten und Informationen von außerhalb aus den Storage Clouds seiner Teilnehmer bezieht und intern, außer Verwaltungsdaten, keine eigenen Informationen speichert. Die Kontrolle und Steuerung ist Aufgabe des Informationssystems, die Datenhoheit liegt bei den Nutzern.

Durch dieses Konzept werden ebenfalls die Anforderungen CO3, CO4 und CO5 erfüllt. Dies wird nachfolgend kurz durch die Darstellung des Storage Cloud-Marktes verdeutlicht. Anschließend helfen die gewonnenen Erkenntnisse bei der Einordnung der Anforderungen. Die umfassende Integration von Storage Clouds in das DCN-System als Speicherort für die Daten der Nutzer, ermöglicht eine vielfältige Auswahl an Storage Cloud-Varianten. Diese können zum einen von externen Anbietern und zum anderen von den Teilnehmern selbst bereitgestellt werden.
Externe Anbieter können unterschieden werden in große kommerzielle Unternehmen, wie etwa Google, Dropbox oder Microsoft, und in kleinere Anbieter, deren Geschäftsmodelle primär den Fokus auf den Schutz und die Integrität der Daten ihrer Kunden legen, dafür aber einen finanziellen Gegenwert erwarten (vgl. [Welt 2014]). Das höhere Sicherheitsniveau wird dabei erreicht, indem zum einen eine Verschlüsselungen der Daten angeboten wird und da der Firmensitz sich zum anderen in Europa befindet, was die Einhaltung des europäischen Datenschutzes erzwingt. (vgl. DS5)
Die meiste Kontrolle über seine Daten hat ein Nutzer, wenn er selbst eine Storage Cloud betreibt. Dies ist realisierbar durch den Betrieb eines eigenen Servers und den Einsatz einer geeigneten Software. Ein Teilnehmer kann diese Anwendung installieren und besitzt somit die absolute Kontrolle über seine gespeicherten Daten. Denkbar ist ebenfalls der Einbezug mehrerer Personen (z. B. Familienmitglieder) oder die Übertragung in den betriebswirtschaftlichen Bereich (private Unternehmens-Clouds). Gleich, welche Lösung ein Anwender bevorzugt, in jedem Fall hat er vollen Zugriff auf seine gespeicherten Informationen und Daten, die zuvor vom Decentral Cloud Network erzeugt wurden. Dadurch besteht zu jeder Zeit die Möglichkeit, die eigenen Daten zu bearbeiten. (vgl. DS2) Dies kann unabhängig vom Netzwerk geschehen und erlaubt zusätzlich den Einsatz systemunabhängiger Software. Dadurch ist eine dynamische Bearbeitung der gespeicherten Informationen erreichbar, was im besonderen Maße für Unternehmen interessant ist, die sich an dem sozia-

len Netzwerk beteiligen.

Die Darstellung des Storage Cloud-Marktes zeigte, dass ein Anwender flexibel in der Auswahl an geeigneten Speichervarianten ist. Neben kostenlosen Angeboten, meist für private Endanwender, bieten kostenpflichtige Dienstleistungen vornehmlich einen gesteigerten Datenschutz. Kosten entstehen ebenfalls bei der Erweiterung des Speicherplatzes. Somit kann von variablen Kosten hinsichtlich der Nutzung des DCN gesprochen werden. (CO3)

Durch die Auswahl eines Storage Cloud-Anbieters aus Deutschland kann der Datenschutz, ausgehend vom deutschen Datenschutzgesetz, sehr hoch gewählt werden: im Vergleich zu US-amerikanischen Anbietern bei denen er geringer ist. Weiterhin kann der Nutzer die Auswahl treffen zwischen einem großen Unternehmen mit sehr umfangreichen Service Level Agreements (SLA) oder einem kleinen Unternehmen, das tendenziell geringere Kosten verursacht, dadurch aber nicht so einen umfassenden Datenschutz bietet. (CO4)

Moderne Storage Cloud-Anbieter, die zumeist spezialisiert sind auf KMU, bieten die Möglichkeit der Verschlüsselung von hinterlegten Kundendaten an. Dies kann in das Konzept integriert werden und erfüllt Anforderung CO5.

Im nachfolgenden Abschnitt wird das Grundkonzept nochmals detailliert vorgestellt. Anschließend erfolgt die Beschreibung des Architekturmodells sowie die interne Kommunikation der einzelnen Komponenten untereinander.

5.1 Grundkonzept

Das Grundkonzept ist eine Detaillierung der 1. Abstraktionsebene aus dem vorhergehenden Kapitel. Es erfolgt eine Beschreibung des Lösungsansatzes, ohne dass bei den beteiligten Komponenten ins Detail gegangen wird. Weiterhin werden die Kommunikationsstrukturen der Elemente untereinander aufgezeigt. Für die Konzipierung ist die Verwendung von bekannten Architekturmustern zielführend, da sich diese aus den langjährigen Erfahrungen des Software Engineerings und dessen Erstellungsprozess ableiten. Das in dieser Dissertation entwickelte Konzept folgt der Serviceorientierten Architektur (SOA) im Sinne eines Architekturmusters. Für eine genau Definition bietet es sich an, den Vergleich zu einem Entwurfsmuster zu vollziehen. Goll [2014, 288] definiert ein Architekturmuster wie folgt:

Entwurfsmuster stellen feinkörnige Muster dar, während Architekturmuster grobkörnig sind. Architekturmuster lösen nicht ein Teilproblem, sondern beeinflussen die Grundzüge der Architektur eines Systems.

SOA ist eine Ausprägung, welche Services verwendet, um die an das System gestellten Aufgaben zu lösen. Dostal et al. [2005, 11] geben hierfür eine genau Definition:

Unter SOA versteht man eine Systemarchitektur, die vielfältige, verschiedene und eventuell inkompatible Methoden oder Applikationen als wiederverwendbare und offen zugreifbare Dienste repräsentiert und dadurch eine plattform- und sprachenunabhängige Nutzung und Wiederverwendung ermöglicht.

Abbildung 27 zeigt die grafische Darstellung einer SOA-Architektur.

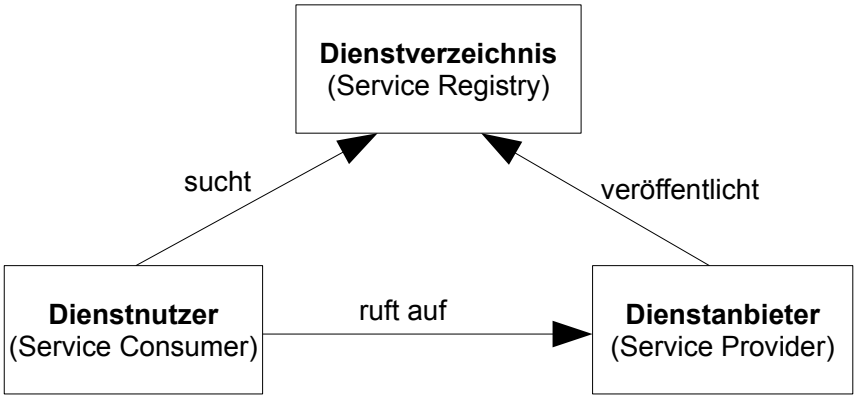


Abbildung 27: Dreieck der SOA-Architektur (in Anlehnung an [Melzer 2010, 14])

Im Unterschied zu einer klassischen SOA-Architektur werden nicht die Dienste in den Mittelpunkt der Betrachtung gestellt, sondern die Daten, die vom Informationssystem verarbeitet werden. Die Daten sind im Unterschied zu den Diensten vorher nicht bekannt, lediglich die Datenquellen, aus denen sie bezogen werden. Durch die Differenz beider Betrachtungsperspektiven wird in dieser Dissertation eine Abstraktion der SOA-Architektur durchgeführt und danach spezialisiert auf das vorgestellte Problemfeld. Es kann dadurch im Sinne einer Referenzarchitektur von einer Umwandlung in ein allgemeingültiges Konzept gesprochen werden, welches eine Ausprägung für einen Spezialfall ermöglicht. Hierbei wurde, bezogen auf Brocke [2007, 51] eine Wiederverwendungsvariante, die Analogie, gewählt. Für eine Analogie, welche einem Ausgangsmodell einer Spezialisierung entspricht, benötigt es zunächst eine Abstraktion des SOA-Modells. Tabelle 29 zeigt die verschiedenen Varianten der Wiederverwendung eines Referenzmodells. Hierbei wurde Bezug genommen auf die Umwandlung eines servicegetriebenen Ansatzes hin zu einem datengetriebenen Ansatz. Das Konzept ist somit datengetrieben statt servicegetrieben.

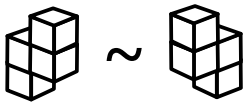
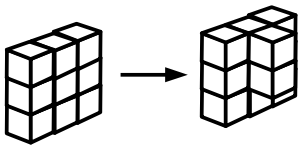
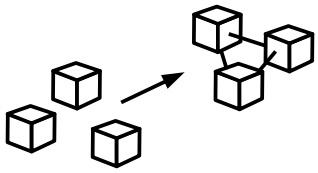
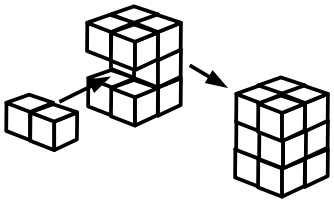
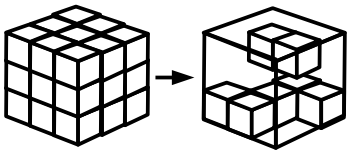
Fünf Wiederverwendungsarten von Referenzmodellen		
Analogie 	Spezialisierung 	Aggregation 
Instanziierung 	Konfiguration 	

Tabelle 29: Wiederverwendungsarten von Referenzmodellen (in Anlehnung an [Brocke 2007, 51])

Der Service Consumer wird zum Data Consumer und stellt für die Nutzer eine Darstellung der Daten bereit. Der Service Provider wird zum Data Provider und ist als Komponente zuständig für die Ermittlung und Weiterleitung der Storage Cloud-Daten der Nutzer. Die Service Registry wird zur Data Registry und verwaltet das Verzeichnis an Nutzern bzw. deren Datenquellen. Das soeben Beschriebene wird in Abbildung 28 nochmals grafisch aufbereitet.

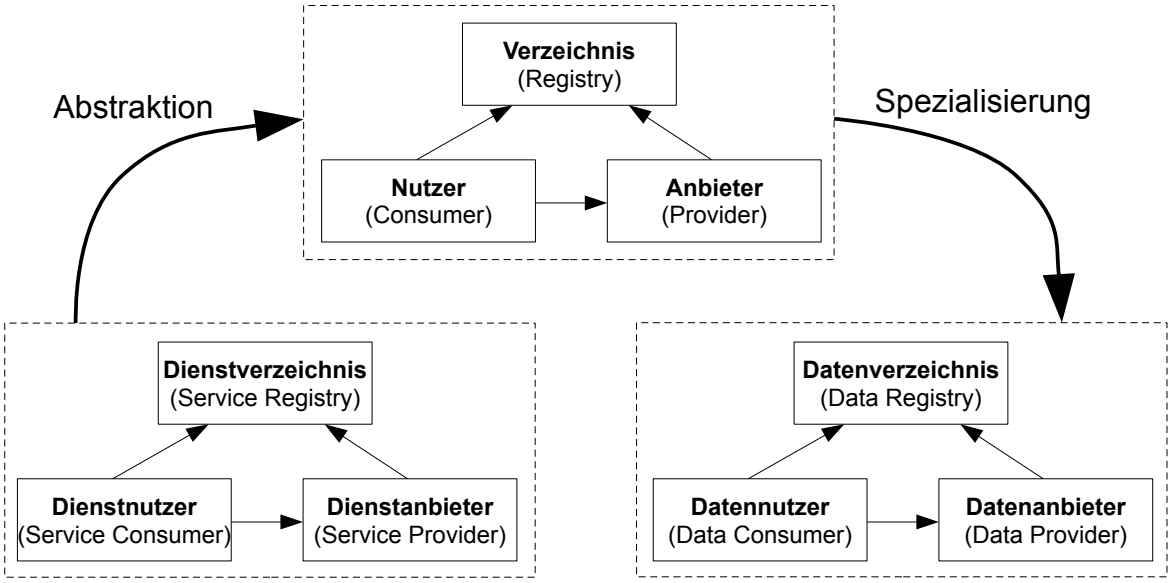


Abbildung 28: Von der Abstraktion des SOA-Modells zur Spezialisierung

Die Registry steht bei dieser Betrachtung weniger im Mittelpunkt des Findens von Infor-

mationen. Die Anfragen und Antworten werden primär über den Provider abgewickelt. Weiterhin ist zu beachten, dass die Nutzer nicht frei zur Verfügung stehen, wie etwa Services, sondern geschützt werden sollen. Das grundsätzliche Ziel dieser Konzeption sind die Minimierung der Kopplung der Komponenten untereinander, sowie der Schutz der Storage Cloud-Zugriffsdaten der Teilnehmer. Damit dieses Ziel erreicht werden kann, ist die interne Kommunikation der Komponenten abzugrenzen von den Zugriffsdaten der Nutzer. Dies wird erreicht durch die Implementierung eines Authentifizierungsverfahrens mit Hilfe eines Authentifizierungsschlüssels. Dieser dient bei jeder internen Kommunikation als Nachweis der Zugriffsberechtigung der Funktionsnutzung.

Das Decentral Cloud Network besteht aus drei Komponenten: Portal, Verbinder und Nutzerverzeichnis. Das Portal ist die primäre Anlaufstelle der Teilnehmer und dient als zentraler Zugriff zum Informationssystem. Mit Hilfe einer grafischen Oberfläche, basierend auf HTML, wird die Darstellung der Daten und Informationen ermöglicht. Weiterhin findet in diesem Bereich die Konfiguration und Integration der Storage Clouds der Nutzer statt.

Der Verbinder ist zuständig für den Zugriff auf die Storage Clouds. Dieser kommuniziert mit den externen Anwendungen und führt eine Verifizierung und Aggregation der Daten durch. Nach der Anwendung von Regeln des Rechtemanagements werden die gewonnenen Daten an das Portal weitergeleitet. Das Nutzerverzeichnis ist verantwortlich für die Nutzerverwaltung und -speicherung. Primär dient es der Speicherung der Zugriffsdaten auf den Storage Clouds.

In dieser Konzeption ist der Nutzer der wichtigste Bestandteil des Informationssystems. Es handelt sich hierbei um eine Umsetzung des Mensch/Maschine/Technik-Systems. Der Mensch als wichtigster Faktor steht im Mittelpunkt mit seinen Interaktionen und mit seinen Daten. Für eine umfassend hohe Verwendbarkeit (engl. *usability*) des Systems wird ihm der Zugriff auf das Portal mit verschieden Endgeräten ermöglicht. Es ist ihm möglich, seine Daten mit Hilfe des Systems abzulegen und zu verwalten. Dies schließt die Datenschutzkomponente durch ein Rechtemanagementsystem mit ein. Realisiert wird dies durch die Integration seines eigenen Systems oder des Anbietersystems in das Gesamtsystem. Somit hat der Teilnehmer die Möglichkeit seine Daten auch über seine Storage Cloud direkt zu bearbeiten. Abbildung 29 zeigt das Gesamtsystem mit den dazugehörigen Komponenten.

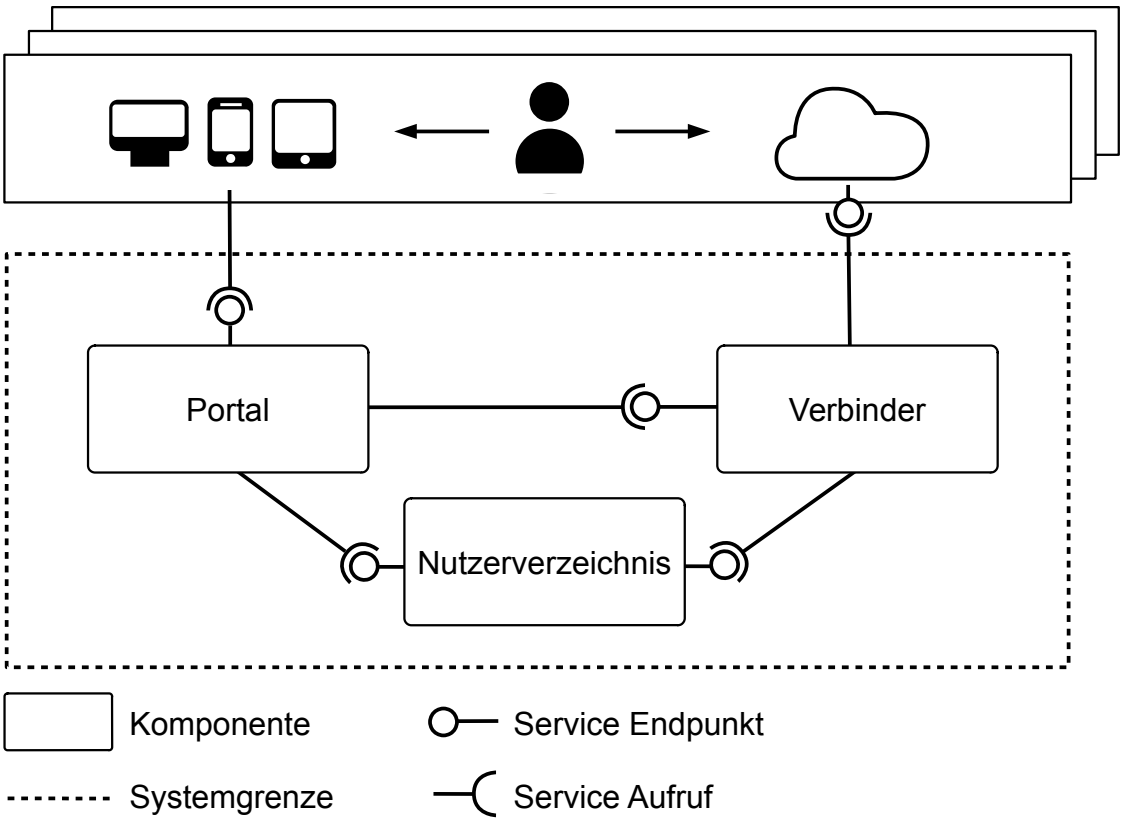


Abbildung 29: Decentral Cloud Network-Grundkonzept

Das nächste Kapitel beinhaltet die dritte Stufe der Detaillierung und eine noch genauere Beschreibung der Komponenten.

5.2 Decentral Cloud Network-Architektur

In diesem Kapitel wird die Architektur des Decentral Cloud Networks detailliert vorgestellt. Diese basiert auf dem Grobkonzept und realisiert die an ein dezentrales Informationssystem gestellten Anforderungen. Die Architektur ist in Softwarekomponenten unterteilt, welche wiederum Unterkomponenten besitzen. (vgl. SY4) Jede Komponente ist hierbei autark und besitzt eigene Eigenschaften im Bezug auf Verfügbarkeit und Transaktionssicherheit. Die lose Koppelung der einzelnen Komponenten erlaubt eine eventuelle Erweiterung der Architektur um externe Dienste, wie etwa ein Suche nach anderen Teilnehmern. Hierbei werden Daten für das Auffinden der Nutzer in einer zusätzlichen Datenbank gespeichert. Dies würde dem Konzept der externen Speicherung und damit verbunden, dem Einhalten des Datenschutzes widersprechen und müsste daher ausgelagert werden. Abbildung 30 zeigt die Architektur des DCN mit den Komponenten, den Kommunikationsverbindungen und den externen Quellen.

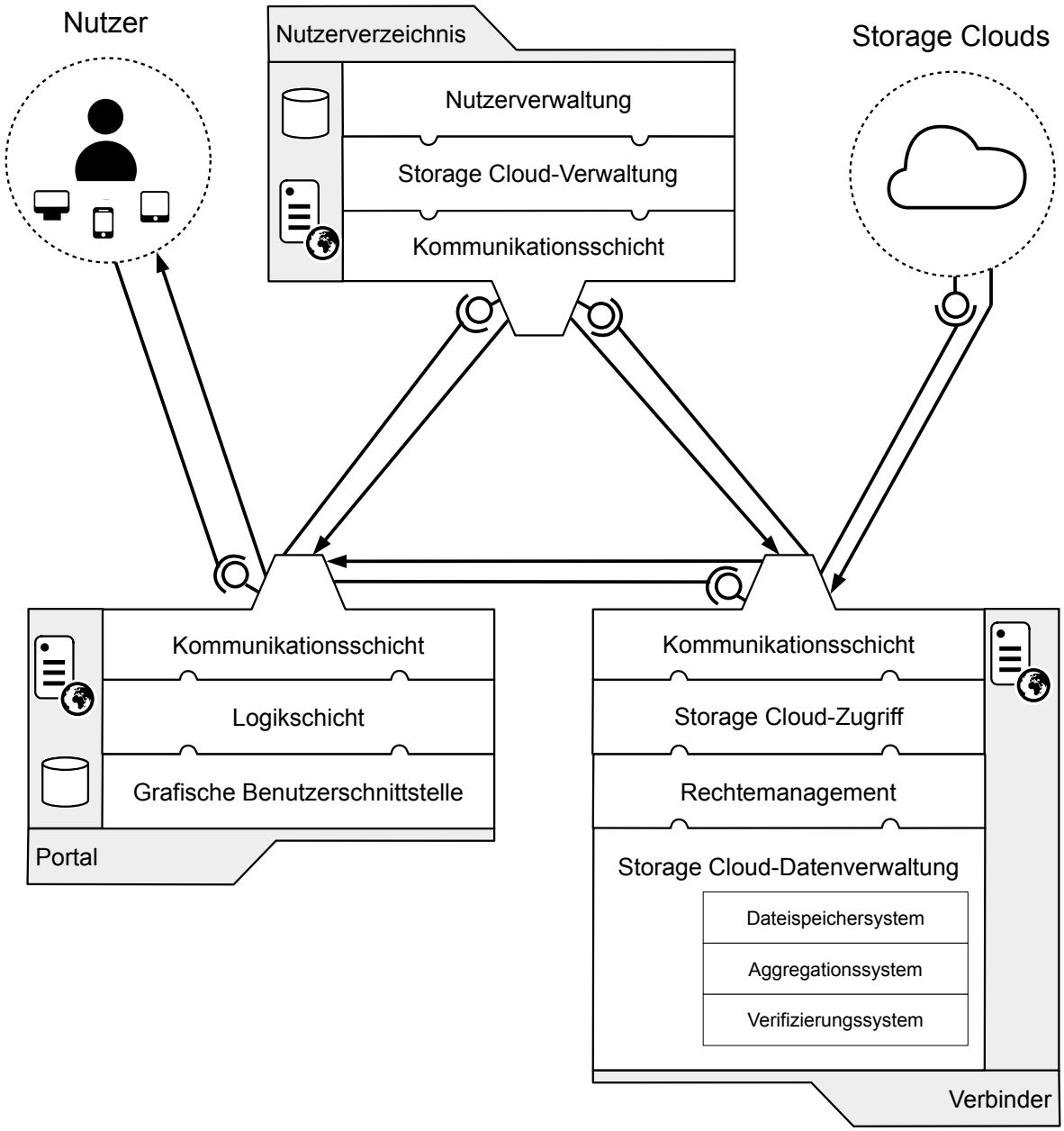


Abbildung 30: Decentral Cloud Network-Architektur

Das Nutzerverzeichnis, das Portal, der Verbinder, die Nutzer und die Storage Clouds werden kurz mit deren jeweiligen Bestandteilen aufgeführt. Eine ausführliche Betrachtung der jeweiligen Komponenten geschieht in den benannten Kapiteln.

Das Nutzerverzeichnis verwendet, für die Erfüllung der spezifischen Aufgaben, als Technologie einen Webserver und eine Datenbank. Es beinhaltet die Unterkomponenten Nutzerverwaltung, Storage Cloud-Verwaltung und Kommunikationsschicht. Das Portal setzt als Technologie ebenfalls einen Webserver und eine Datenbank ein. Die Datenbank ist für die

Speicherung von Konfigurations- und Administrationsdaten der Nutzer verantwortlich und somit ein Element, den Komfort des Portals für die Anwender zu steigern. Die Unterkomponenten sind: Grafische Benutzerschnittstelle, Logikschicht und Kommunikationsschicht. Der Verbinder ist eine statische Komponente ohne die Notwendigkeit Daten zu speichern. Daher wird als einzige Technologie ein Webserver benötigt. Der Verbinder verfügt über eine umfangreiche Anzahl an Unterkomponenten: eine Storage Cloud-Datenverwaltung, welche sich wiederum in die Unterkomponenten Dateispeichersystem, Aggregationssystem und Verifizierungssystem aufteilt, ein Rechtemanagement, einen Storage Cloud-Zugriff und eine Kommunikationsschicht. Das sich außerhalb der Systemgrenze befindliche Element Nutzer beschreibt den Zugriff von außen durch verschiedene Endgeräte, wie etwa einen Desktop, ein Smartphone oder ein Tablet. Das Element Storage Clouds, welches sich ebenfalls außerhalb der Systemgrenze befindet, stellt die externen Quellen für das Informationssystem dar. Hierunter befinden sich neben klassischen Storage Clouds (verwendete Software oder Anbieter), auch NAS-Systeme. Der Zugriff erfolgt hierbei üblicherweise durch REST-API-Schnittstellen. Der nächste Abschnitt beschreibt die interne Kommunikation der Komponenten untereinander.

5.3 Interne Kommunikation zwischen Komponenten

Das Konzept für die Kommunikation prägt in entscheidendem Maße die zu entwickelnde Architektur. Eine gesonderte, umfangreiche Betrachtung des Themenfeldes ist daher förderlich für das Grundverständnis. Hierbei werden der Einsatz von aktuellen Standards ebenso berücksichtigt wie Best Practice-Ansätze.

Grundsätzlich kommunizieren die lose gekoppelten Komponenten untereinander mit Hilfe von HTTP-Aufrufen. Das Ziel ist die Auslagerung von In-Memory Function Calls zu Out-of-Process Calls, was einen positiven Einfluss auf die Steigerung der losen Kopplung hat. Als In-Memory Function Calls werden Aufrufe bezeichnet, welche Softwarebibliotheken als Komponenten mit dem Programm verknüpfen. Im Gegensatz dazu werden Out-of-Process Calls umgesetzt mit Diensten bzw. Services, die mit einem Mechanismus, wie einer Webservice-Anfrage oder einem Remote Procedure Call, untereinander kommunizieren. (vgl. [Fowler/Lewis 2014])

Für eine wissenschaftlich fundierte Betrachtung wird das Kommunikationsmodell nach Shannon und Weaver [1949] auf den vorliegenden Anwendungsfall übertragen. Zunächst erfolgt die Beschreibung des Kommunikationsmodells in seiner grundsätzlichen Form. Die

Datenübertragung erfolgt hierbei von einer Quelle hin zu einer Senke. Der Sender sorgt für die Vorbereitung der Übertragung. Daten werden anschließend mit Hilfe eines gemeinsamen Zeichenvorrates über einen Kanal übertragen. Abschließend nimmt der Empfänger die Nachricht entgegen und übergibt sie der Senke für die weitere Verarbeitung. (vgl. [Shannon/Weaver 1949, 50ff.]) Abbildung 31 zeigt eine schematische Darstellung des Kommunikationsmodells.

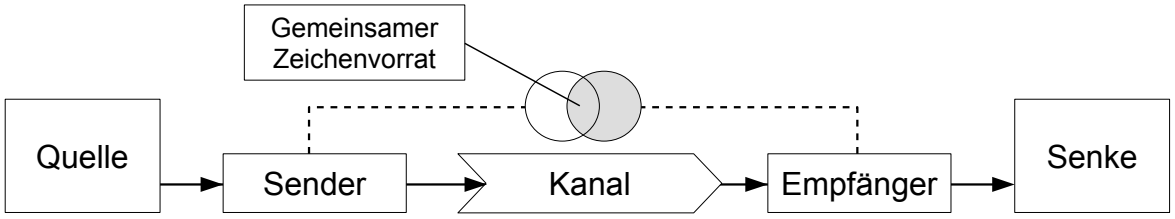


Abbildung 31: Kommunikationsmodell nach Shannon und Weaver (in Anlehnung an [Shannon/Weaver 1949], übersetzt von [Goldammer 2013, 50])

Dieses Kommunikationsmodell wird übertragen auf das in dieser Dissertation vorgestellte Konzept. Die Quelle bzw. Senke entspricht hierbei den jeweiligen Komponenten. Sender und Empfänger werden durch HTTP-Aufrufe und Webservices (Webservice-Endpunkte) realisiert. Der Kanal wird mit Hilfe von HTTP und mit den Kommunikationsformen GET und POST umgesetzt. Die verwendete Kommunikationssprache ist JSON mit festgelegten Bezeichnern und einem gemeinsamen Zeichenvorrat in UTF-8.

Das in dieser Dissertation vorgestellte Konzept nutzt REST-Webservices in Form einer Service-Architektur. Ein Webservice ist ein Softwaresystem, welches entworfen, wurde um interoperable Maschine-zu-Maschine-Interaktionen über ein Netzwerk zu unterstützen [W3C 2004]. Weiterhin kann durch ein Interface der Zugriff in einem maschinenauswertbaren Format beschrieben werden. Dadurch ist es anderen Systemen möglich, mit diesem zu interagieren. Typischerweise werden hierfür web-bezogene Standards verwendet (HTTP, XML, JSON, etc.). (vgl. [W3C 2004])

Das Konzept des Representational State Transfer (REST) ist ein vergleichbares Verfahren wie SOAP und WSDL für die vereinfachte Maschine-zu-Maschine-Kommunikation. REST besitzt nach Tilkov et al. [2015, 11 ff.] fünf Kernprinzipien: Ressourcen mit eindeutiger Identifikation, Verknüpfungen/Hypermedia, Standardmethoden, unterschiedliche Repräsentationen und statuslose Kommunikation.

Das Prinzip Ressourcen mit eindeutiger Identifikation bezieht sich auf die Eindeutigkeit des Zugriffs. Dies wird mit Hilfe von IDs, im speziellen mit URIs, realisiert. Diese bilden

einen globalen Namensraum. Dadurch wird sichergestellt, dass Elemente weltweit eindeutig identifiziert werden können. Verknüpfungen und Hypermedia enthalten beim Zugriff weitere Informationen über das Objekt. Mit sogenannten Links werden verschiedene Dokumente und Informationsquellen miteinander verknüpft. Standardmethoden erlauben eine einheitliche Kommunikation über Systemgrenzen hinweg. Hierbei ist die HTTP-Spezifikation ein zentraler Bestandteil. Mit den Befehlen GET und POST (Kontextabhängig auch: PUT, DELETE, HEAD und OPTIONS) werden Software- und Systeminstanzen angesprochen. Das Prinzip der unterschiedlichen Repräsentationen fokussiert eine Trennung der Verantwortlichkeiten für Daten und Operationen. Die statuslose Kommunikation legt fest, dass der Zustand des Systems entweder vom Client gehalten oder vom Server in einen Ressourcenstatus umgewandelt wird. Somit ist jede Anfrage unabhängig von vorhergehenden Interaktionen. Ein serverseitig abgelegter, transienter, clientspezifischer Status ist konzeptionell ausgeschlossen. (vgl. [Tilkov et al. 2015, 11 ff.])

Aus der Verbindung von Webservices und REST ergibt sich der REST Webservice. Dieser vereint beide zuvor dargestellten Elemente. In besonderem Maße ist die Eigenschaft des Zustandes ein prägendes Kriterium. Es sind zwei Arten des Zustandes zu unterscheiden: Ressourcenzustand und Anwendungszustand. Der Ressourcenzustand beschreibt den aktuellen auf dem Server befindlichen Zustand. Dieser wird dem Client in Form einer Repräsentation übergeben. Der Anwendungszustand verbleibt beim Client und beschreibt den Pfad und dessen Informationen, welcher durch die Anwendung erstellt wurde. (vgl. [Richardson/Ruby 2007, 246]) Werden alle Anforderungen eingehalten, so wird von einem REST-konformen Webservice gesprochen. Konformität wird erreicht durch die Zustandslosigkeit des Webservices, das heißt, dass der Server zu keinem Zeitpunkt den Zustand der Anwendung speichert. Weiterhin ist jede Anfrage durch den Client an den Server isoliert. Die Manipulation des Ressourcenzustandes wird ausschließlich durch Anfragen an den Server erreicht. (vgl. [Richardson/Ruby 2007, 246])

Durch die umfassende Betrachtung des Themenfeldes ist es möglich, die Ausgestaltung des Kommunikationsmodells für dieses Konzept vorzunehmen. Die Anfrage (engl.: *Request*) an eine Komponente des Netzwerkes erfolgt in Form von HTTP. Diese stellt einen REST Webservice bereit. Die Argumentenübertragung wird realisiert durch GET und POST. Bei GET sind Daten als Parameter-Wertepaare Teil der URL. POST eignet sich bei großen Datenmengen, zum Beispiel bei Bildern, und überträgt die Argumente in einem HTTP-Nach-

richtenrumpf. Die Argumente sind somit nicht in der URL sichtbar. Die Antwort (engl.: *Response*) erfolgt als HTTP-Übertragung.

Die Übertragung der Daten erfolgt in Form des Standards JavaScript Object Notation (JSON). Hierbei handelt es sich um ein Datenformat, welches von Maschinen und Menschen einfach lesbar ist. Dieses ist zunächst Bestandteil von JavaScript gewesen und hat neben der internen Datenverwaltung den Zweck, den Datenaustausch zwischen Anwendungen einfach zu gestalten. (vgl. [ECMA 2013] und [IETF/Bray 2014]) Es kann neben XML als de facto-Standard für den HTTP-basierten Datenaustausch zwischen Anwendungen gesehen werden.

Aus der Beschreibung der einzelnen Bestandteile der Kommunikation kann das REST-Webservice-Kommunikationsmodell abgeleitet werden. Abbildung 32 zeigt das bei diesem Konzept verwendete zweistufige Modell der REST-Kommunikation.

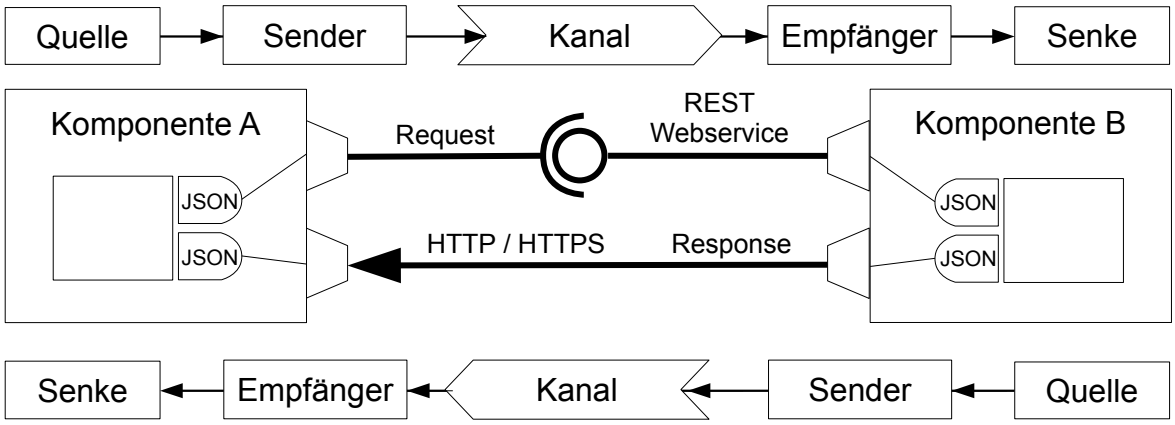


Abbildung 32: Zweistufiges Modell der REST-Kommunikation

5.4 Zusammenfassung

In diesem Kapitel wurde das Konzept des Decentral Cloud Networks sowie dessen Architektur vorgestellt. Dadurch war es möglich, bereits eine Vielzahl an Anforderungen zu erfüllen. Durch eine schrittweise wissenschafts-getriebene Konzipierung wurde der Konstruktionsprozess innerhalb der Entwurfsphase nachvollziehbar und begründbar dargelegt. Die interne Kommunikation wurde realisiert mit REST-Webservices und bezieht sich auf das Kommunikationsmodell von Shannon und Weaver. In den folgenden Kapiteln werden die einzelnen Komponenten und deren Kommunikation untereinander näher beschrieben.

6 DCN: Portal

Das Portal ist die erste Anlaufstelle für die Benutzung des Decentral Cloud Networks. Es ist erreichbar über das Internet durch eine Webadresse. Somit ist eine direkte Kommunikation mit den Anwendern und deren verschiedenen Endgeräten möglich. Die Aufgaben dieser Komponente sind sehr umfangreich und dienen primär der Interaktion mit den Nutzern. Für die Teilnahme ist zunächst eine Registrierung am Portal notwendig. Dies schließt ebenfalls eine Registrierung im Netzwerk mit ein. Das Portal übernimmt die grafische Aufbereitung sämtlicher aus externen Quellen bezogener Daten und Informationen der Nutzer und visualisiert sie durch eine HTML-Webseite. Weiterhin zeigt es dem Anwender verschiedene Möglichkeiten der Interaktion auf. Es erkennt logische Zusammenhänge und führt den Nutzer durch die Anwendung. Neben der zeitverzögerten Interaktion mit Teilnehmern, in Form von sozialen Feedbacks auf Daten, ist die Direktkommunikation durch ein Chat-System realisiert.

Das Portal kommuniziert innerhalb des Netzwerkes mit den Komponenten Nutzerverzeichnis und Verbinder. Hierfür werden bereitgestellte REST-Webservices verwendet. Es überträgt die hinterlegte ID und das Passwort des Nutzers an das Nutzerverzeichnis, um den aktuellen Authentifizierungsschlüssel zu erhalten. Die Kommunikation zum Verbinder dient der Übertragung von Daten an das Portal. Dieser sendet auf Anfrage alle relevanten Informationen und Daten eines Nutzers und seiner Beziehungen, mit denen er in Kontakt steht.

Für die Bereitstellung sämtlicher Funktionalitäten werden auf technologischer Seite ein Webserver und eine Datenbank verwendet. Der Webserver dient der Kommunikation zu den Komponenten sowie zu den Teilnehmern. Die Datenbank speichert alle individuellen Daten der Nutzer für die Nutzung des Portals. Diese dienen der Administration, Individualisierung und Komfortsteigerung. Diese sind zu unterscheiden von Daten, die für die Interaktion vom Nutzer selbst angelegt werden. Diese stehen unter einem besonderem Schutz für die Einhaltung von Datenschutzaspekten und werden zu keiner Zeit vom Netzwerk gespeichert. Abbildung 33 verdeutlicht mit Hilfe einer schematischen Darstellung die Architektur der Portal-Komponente. Nachfolgend werden alle Bestandteile des Portals näher vorgestellt.

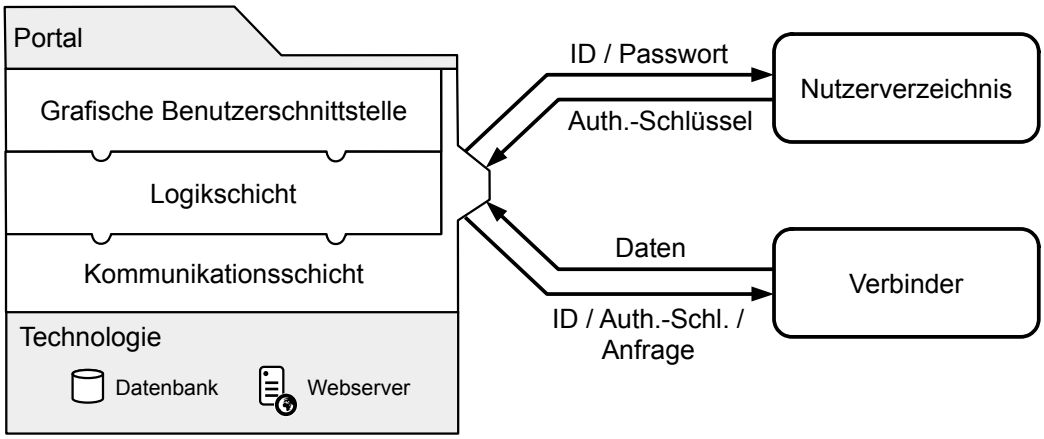


Abbildung 33: Portal-Architektur

6.1 Portal-Komponenten

Das Portal besteht aus drei Unterkomponenten: Grafische Benutzerschnittstelle (6.1.1), Logikschicht (6.1.2) und Kommunikationsschicht (6.1.3). Die Grafische Benutzerschnittstelle realisiert die HTML-basierte Darstellung der Daten und Informationen aus dem Netzwerk und dient der Interaktion mit anderen Nutzern. Die Logikschicht übernimmt die Verarbeitung und Aufbereitung der darzustellenden Daten und Informationen. Die Kommunikationsschicht dient dem Datenaustausch mit dem Nutzerverzeichnis, dem Verbinder und den Anwendern. Nachfolgend werden alle Unterkomponenten detailliert vorgestellt.

6.1.1 Grafische Benutzerschnittstelle

Die Grafische Benutzerschnittstelle (GUI) ist verantwortlich für die Interaktion der Anwender mit dem Netzwerk. Hierbei wird eine HTML-basierte Visualisierung, ergänzt durch JavaScript, eingesetzt. Ziel ist die Darstellung der Daten und Informationen für verschiedene Endgeräte. Hierfür können zwei Strategien verfolgt werden: entweder der Einsatz von Responsive Design oder eine parallele Entwicklung für jedes Ausgabegerät.

Bei Responsive Design handelt es sich um ein Bündel an Maßnahmen, im Wesentlichen auf der Clientseite, welche das Anlegen von Webseiten vereinfacht. Diese lassen sich, basierend auf der Bildschirmgröße, optimal an unterschiedliche visuelle Ausgabegeräte anpassen. Drei Hauptmerkmale sind dem Responsive Design zuzuordnen: fluides Layoutraster, anpassungsfähige Inhalte und Layoutumbrüche durch Media Queries[8]. (vgl.

8 Media Queries: Überprüfung von Bedingungen für bestimmte Medienfunktionen (Weite, Höhe, Farbe). Mit Media Queries können Präsentationen auf einen bestimmten Bereich der Ausgabegeräte angepasst werden ohne Änderung des Inhaltes selbst. (vgl. [W3C 2012])

[Laborenz/Ertel 2014, 21])

Im Sinne eines optimalen Mensch/Aufgabe/Technik-Systems ist es denkbar, Ausprägungen basierend auf verschiedenen Nutzergruppen zu realisieren. So ist nicht nur das Ausgabegerät ein Faktor der Visualisierung, sondern auch der Mensch, der das entsprechende Gerät nutzt. Eine Parallelentwicklung die die Bedürfnisse der Anwender umfassend befriedigt, findet sich unter anderem in den Bereichen Schriftgröße, vereinfachte Sprache, deutliche Kontraste, große Symbole und eingeschränkter Funktionsumfang.

Neben der Darstellung von Inhalten besitzt die grafische Benutzerschnittstelle eine Vielzahl an weiteren Funktionen. Zunächst ist sie verantwortlich für den Anmeldeprozess des Nutzers am Portal. Weiterhin ermöglicht sie die Erstellung von Inhalten, einhergehend mit sozialen Interaktionen. Ein weiterer wichtiger Aspekt ist das Festlegen von Rechten sowie die Rechteverwaltung, ergänzt durch die Darstellung von Datenschutz und Sicherheit. Sie ermöglicht die Konfiguration des Portals sowie die Konfiguration des Storage Cloud-Zugriffs. Für die Direktkommunikation ist innerhalb des GUI die Integration eines Chat-Systems eine zu erfüllende Anforderung. (vgl. CO9) Dies wird realisiert durch die Verwendung eines externen Dienstes. Hierfür eignet sich in besonderem Maße das freie und standardisierte XMPP-Protokoll.

Ein Ziel moderner Informationssysteme ist die visuelle Rückmeldung (engl.: *Feedback*) über Aktionen und Zustände der durch den Nutzer initiierten Prozesse. In diesem Kontext ist der Bereich des Darstellens von Sicherheit und der Höhe des Datenschutzes von besonderem Interesse. (vgl. DS1) Dieser muss deutlich hervorgehoben werden. Diese Anforderungen werden durch die Visualisierung der Verschlüsselung (z. B. SSH) und des Schutzes mithilfe von Rechten und Regeln erfüllt. Die Höhe an Schutz wird primär durch das Beziehungsmanagement und dessen Kategorien dargestellt. Die Lösungsstrategie, des in dieser Dissertation vorgestellten Konzeptes, basiert auf symbolischen Verdeutlichungen. Die Symbolik ist geeignet Informationen zu übermitteln, da die Nutzer sowohl aus anderen Anwendungen, als auch aus der realen Welt Symbole kennen, die eine ganz bestimmte Bedeutung besitzen. Bezugnehmend auf die Gestaltpsychologie und das Gesetz der Vertrautheit bilden Objekte Gruppen, die als Gruppe vertraut erscheinen (vgl. [Goldstein 2002, 195]). Für die Darstellung von Schutz eignen sich Abbildungen wie Schlüssel, Schloss, Schild und Maske. Eine zentrale Frage, die sich daraus ergibt, lautet: Ist es zielführender, das Einhalten oder das Nichteinhalten von Datenschutz darzustellen? Wird davon ausgegangen,

dass der Anwender möglichst wenig durch visuelle Reize abgelenkt werden soll, ist die Darstellung des Nichteinhaltens zu bevorzugen. Dennoch müssen wichtige Informationen über die Einhaltung visualisiert werden. Im Fokus steht hierbei die Eignung für das Kommunikationsziel als Gestaltungsgrundsatz nach DIN EN ISO 9241-10 (vgl. [Sarodnick/ Brau 2006, 41]).

Ein weiteres Ziel ist die nachvollziehbare Darstellung des Datenschutzes (vgl. DS8). Der Nutzer muss die Begründung sowie den Prozess des Datenschutzes sehen können. Neben der Einteilung von Kategorien an Nutzern muss aufgezeigt werden, wer die aktuellen Daten sehen kann. Durch die verschiedenen Ebenen der Beziehung gibt es eine Vielzahl an Ausprägungen für das Recht Daten zu erhalten. Die Funktionalität „Daten aus der Sicht von anzeigen“ ist hierbei zielführend. Weiterhin ist ein umfangreiches Monitoring sowie Feedback über angezeigte Daten sinnvoll. Hierbei können ebenfalls Symbole die Informationsübertragung zum Nutzer unterstützen.

Schlussendlich hat das GUI das Potential, Vertrauen bei den Nutzern zu erzeugen (vgl. TR2). Dies wird unter anderem durch andere Teilnehmer erreicht. Die Darstellung anderer Nutzer, die das Informationssystem erfolgreich einsetzen, hat positiven Einfluss auf den Netzwerkeffekt und damit auf das Vertrauen. Weiterhin unabkömmlich ist der deutliche Hinweis auf die Erfüllung datenschutzrechtlicher Anforderungen. Ergänzt wird dies durch Anzeigen externer Bewertungen, wie etwa durch ein TÜV-Zertifikat.

6.1.2 Logikschicht

Die Unterkomponente Logikschicht verwaltet die Datenströme innerhalb des Systems, welche von dem Verbinder gesendet werden. Sie berechnet intern Informationen, um für den Nutzer einen Mehrwert zu generieren. Alle Anfragen der Nutzer werden durch die Kommunikationsschicht an den Verbinder gesendet. Die gewonnenen Ergebnisse werden mit zusätzlichen Informationen angereichert. Weiterhin unterstützt es den Nutzer beim Umgang mit dem Netzwerk, gerade im Bezug auf das Rechtemanagement. Es erkennt Abhängigkeiten, gibt Empfehlungen für Einstellungen, fasst Inhalte zusammen und verweist auf weitere Informationsquellen und Inhalte. Darüber hinaus verwaltet es durch die Verwendung eines Authentifizierungsschlüssels das interne Session-Management.

Die Logikschicht des Portals berechnet automatisch Empfehlungen für den Nutzer, basierend auf der Kommunikationsintensität, unter Einhaltung des Datenschutzes. Dadurch kann es dem Nutzer Empfehlungen für die Datenschutzeinstellungen geben (vgl. DS9).

Grundsätzlich wird hierbei das Prinzip des Privacy by Default befolgt. Weiterhin werden ausgehend vom Beziehungsmanagement des Systems Empfehlungen für Einstellungen angegeben (vgl. RM2). Diese basieren auf der Anzahl an Kommunikation und Interaktion. Die sogenannte Interaktionsintensität erlaubt die Angabe von Vorschlägen. Die Logikschicht unterstützt den Nutzer des Weiteren beim Kontaktmanagement (vgl. RM1). Das Portal bietet die Möglichkeit, Kontaktanfragen an Nutzer des Netzwerkes zu senden. Diese werden in den jeweiligen Storage Clouds gespeichert. Hierbei werden neue Kontakte stets als Bekannte deklariert (Privacy by Default).

Die Logikschicht übernimmt weiterhin das Selbstdarstellungsmanagement in Form von Monitoring und Feedback. (vgl. SY8) Zugriffsdaten können beim externen Anbieter bezogen und innerhalb des Systems in grafischer Form (Diagramme) abgerufen werden. Weiterhin wird innerhalb des Netzwerkes mit sozialen Interaktionen, wie „Gefällt mir“, eine direkte Rückmeldung integriert.

6.1.3 Kommunikationsschicht

Die Kommunikationsschicht innerhalb des Portals verwaltet die Kommunikation zu externen Komponenten sowie zu den Teilnehmern des Netzwerkes. Weiterhin stellt sie die Verbindung zu dem Nutzerverzeichnis und dem Verbinder her. An das Nutzerverzeichnis wird die Anfrage nach der Authentifizierung des Nutzers gestellt. Zusätzlich werden die Konfiguration bzw. Änderungen der Storage Cloud-Zugriffsdaten übermittelt. Die Kommunikation zum Verbinder dient primär der Anfrage nach Inhalten, Daten und Informationen. Dies können einzelne Nutzer-relevante Daten, Daten anderer Nutzer und Mediadaten, wie z. B. Bilder, sein. Darüber hinaus werden Änderungen und neue Daten an den Verbinder gesendet, damit diese auf den Storage Clouds der Nutzer ablegt werden können.

6.2 Zusammenfassung

Dieses Kapitel stellte die Komponente Portal, dessen Architektur sowie deren Unterkomponenten detailliert vor. Es wurde gezeigt, welche Aufgabe das Portal besitzt und welche Technologien zur Erfüllung eingesetzt werden. Weiterhin wurde verdeutlicht, welche Schnittstellen zur Kommunikation zu den anderen Komponenten und der Umwelt bestehen.

7 DCN: Nutzerverzeichnis

Das Nutzerverzeichnis ist das zentrale Element des entwickelten Konzeptes dieser Arbeit. Durch die Speicherung von hochsensiblen Administrationsdaten der Nutzer ist ein besonderer Fokus auf die Sicherheit der Komponente gelegt. Grundsätzlich ist keine Kommunikation des Elementes mit der Systemaußenwelt möglich. Sie ist ein isolierter Systembestandteil, welcher nur von den Komponenten des Netzwerkes aufgerufen werden kann. Dies ergibt sich aus der Zielstellung, dass der Schutz von Nutzerdaten die höchste Priorität besitzt.

Die primäre Aufgabe ist die Verwaltung der Nutzerdaten für das Netzwerk an sich und der Storage Cloud-Zugriffsdaten. Zusätzlich können mit Hilfe des Portals die gespeicherten Informationen für die Storage Cloud-Verwaltung geändert oder neu angelegt werden.

Für die Steigerung der Sicherheit der internen Kommunikation erzeugt es einen Authentifizierungsschlüssel, der bei jeder Transaktion übergeben und geprüft wird. Dadurch wird die Benutzung des Passwortes eines Teilnehmers überflüssig und die Sicherheit in Bezug auf die Datenübertragung des Passwortes erhöht. Bei Angabe einer gültigen ID sowie des Passwortes eines Nutzers wird der aktuelle individuelle Authentifizierungsschlüssel zurück gesendet.

Durch die Vergabe einer Identifikationsnummer wird die Anonymität der Nutzer gewahrt. (vgl. DS4) Es werden keine weiteren Daten über die Nutzer innerhalb des Systems gespeichert. Nur Daten für die interne Administration und die individuelle Anpassung des Portals zur Komfortsteigerung. Die absolute Sicherheit für die Nutzerzugriffsdaten wird durch die Abtrennung der Komponente positiv beeinflusst.

Die technische Realisierung erfolgt durch den Einsatz eines Webservers und einer Datenbank. Der Webserver dient dem Zugriff auf das Nutzerverzeichnis. Die Kommunikationsschnittstellen stehen hierbei ausschließlich dem Portal und dem Verbinder zur Verfügung. Weiterhin stellt es keine eigenen Anfragen an das Netzwerk. Die Datenbank speichert alle Nutzerzugriffsdaten für den Zugriff auf das Netzwerk und die Zugriffsdaten für die Kommunikation mit den Storage Clouds.

Abbildung 34 verdeutlicht mit Hilfe einer schematischen Darstellung die Architektur der Nutzerverzeichnis-Komponente. Nachfolgend werden alle Unterkomponenten näher vorgestellt.

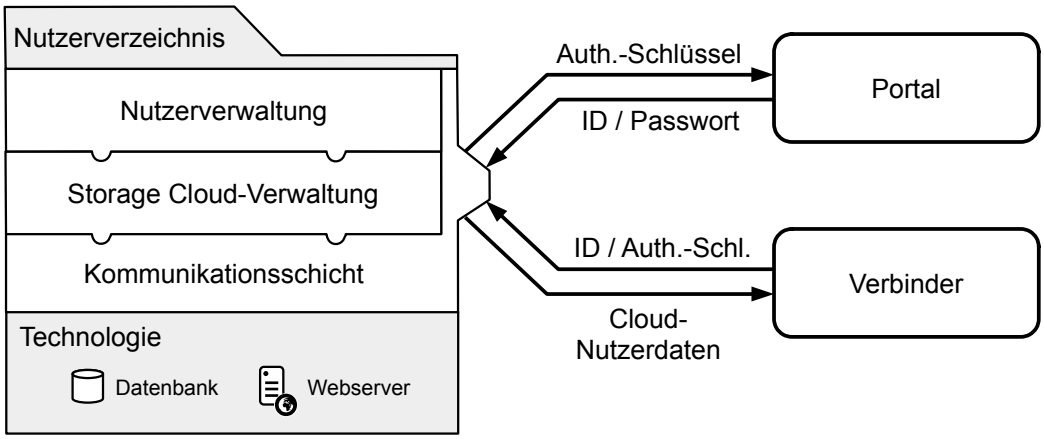


Abbildung 34: Nutzerverzeichnis-Architektur

7.1 Nutzerverzeichnis-Komponenten

Das Nutzerverzeichnis besitzt drei Unterkomponenten: Nutzerverwaltung, Storage Cloud-Verwaltung und Kommunikationsschicht. Die Nutzerverwaltung verwaltet die Speicherung der Zugriffsdaten der Nutzer für das Netzwerk. Die Storage Cloud-Verwaltung dient der Speicherung der Zugriffsdaten der Nutzer für die Storage Clouds. Die Kommunikationsschicht dient der Kommunikation mit dem Portal und dem Verbinder. Die passive Ausrichtung sorgt dafür, dass das Nutzerverzeichnis keine eigenen Anfragen an das Netzwerk stellt.

7.1.1 Nutzerverwaltung

Die Nutzerverwaltung verwaltet alle Nutzer, die am Netzwerk teilnehmen und speichert die Administrationsdaten innerhalb einer Datenbank ab. Jeder Nutzer erhält bei der Registrierung eine eindeutige Identifikationsnummer (ID). Durch diese Eindeutigkeit ist sie in besonderem Maße als Primärschlüssel geeignet. Weiterhin gibt es ein Passwort für den sicheren Zugang. Für jeden Benutzer wird nach Bedarf ein Authentifizierungsschlüssel generiert, welcher ebenfalls in der Datenbank gespeichert wird. Dieser Schlüssel wird entweder nutzungsbasiert oder zeitbasiert neu generiert. Nutzungsbasiert bedeuted bei jeder neuen Sitzung und zeitbasiert zum Beispiel zu jeder Stunde.

7.1.2 Storage Cloud-Verwaltung

Die Storage Cloud-Verwaltung speichert den Zugriff auf die Storage Clouds in einer Datenbank. Hierfür werden die jeweiligen Zugriffsdaten die einen Zugriff ermöglichen, detailliert festgehalten. Dies sind: Nutzer(-name), Passwort, Cloud-Typ, Cloud-Zugriffsart und eine eventuelle Verschlüsselung. Je nach Storage Cloud-Typ werden später innerhalb

des Systems andere Prozesse für den Zugriff verwendet.

7.1.3 Kommunikationsschicht

Die Kommunikationsschicht des Nutzerverzeichnisses dient der Kommunikation mit dem Portal und dem Verbinder. Hierbei sendet das Nutzerverzeichnis keine eigenen Anfragen. Das Portal empfängt nach der Anfrage und der Authentifizierung einen Authentifizierungsschlüssel. Weiterhin werden Schnittstellen angeboten, die es erlauben, die Zugriffsdaten zu konfigurieren. Für Aktionen auf den Storage Clouds übernimmt das Nutzerverzeichnis für den Verbinder die Überprüfung der Richtigkeit des Authentifizierungsschlüssels. Weiterhin übergibt es die Zugriffsdaten für die Storage Clouds.

7.2 Zusammenfassung

In diesem Kapitel wurden das Nutzerverzeichnis, dessen Unterkomponenten und die Architektur desselben näher vorgestellt. Es wurde gezeigt, welche Aufgaben das Nutzerverzeichnis besitzt und wie diese realisiert werden. Weiterhin wurde die verwendete Technologie aufgezeigt. Ergänzend dazu wurde vorgestellt, welche Schnittstellen zu anderen Komponenten bestehen. Das nachfolgende Kapitel stellt die Komponente Verbinder detailliert vor.

8 DCN: Verbinder

Der Verbinder hat die Aufgabe, den Zugriff zu externen Medien, wie etwa zu den Storage Clouds, zu realisieren. Neben dem reinen Zugriff überwacht er die Richtigkeit der Übertragung und das Zusammenführen von Informationen. Intern speichert er keine Daten der Nutzer und ist darüber hinaus statuslos in Bezug auf Langzeitdaten. Dadurch eignet er sich im besonderen Maße für eine Skalierung.

Die Unterkomponenten des Verbinders sind: Storage Cloud-Datenverwaltung, Rechtemanagement, Storage Cloud-Zugriff und Kommunikationsschicht. Die technische Realisierung erfolgt durch den Einsatz eines Webservers. Auf eine Datenbank wird verzichtet. Abbildung 35 gibt eine schematische Darstellung für die Architektur der Verbinder-Komponente wieder. Nachfolgend werden die Unterkomponenten detailliert vorgestellt.

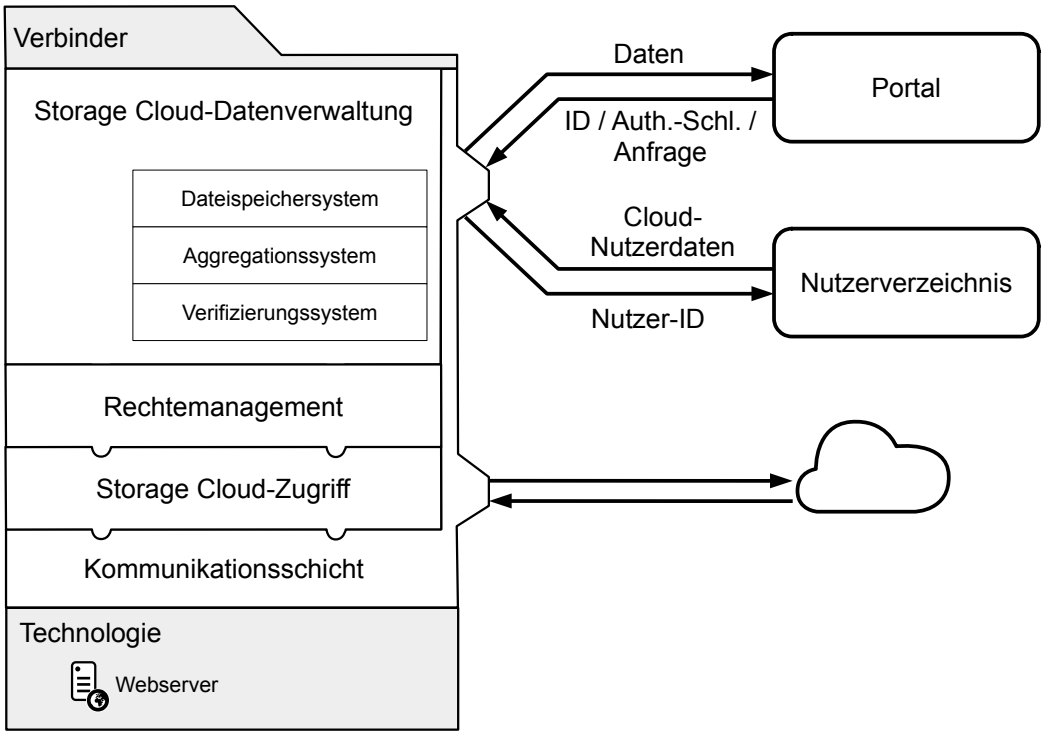


Abbildung 35: Verbinder-Architektur

8.1 Verbinder-Komponenten

Der Verbinder besteht aus den vier Unterkomponenten: Storage Cloud-Datenverwaltung, Rechtemanagement, Storage Cloud-Zugriff und Kommunikationsschicht. Die Storage Cloud-Datenverwaltung sorgt für die korrekte Speicherung der Daten auf den externen Storage Clouds. Hierfür werden festgelegte Standards verwendet. Daten werden darüber hin-

aus erweitert um Metainformationen, wie soziale Interaktionen (Kommentare und Likes), Zeit und Standortinformationen. Für die umfangreichen Aufgaben besteht die Datenverwaltung aus weiteren Unterkomponenten: Dateispeichersystem, Aggregationssystem und Verifizierungssystem.

Das Rechtemanagement umfasst die Absicherung sämtlicher Transaktionen an das Portal. Hierbei wird das Konzept des Beziehungsmanagements verwendet. Der Storage Cloud-Zugriff sorgt für eine Verbindung zu den Storage Clouds. Die Kommunikationsschicht stellt für die interne Kommunikation Schnittstellen in Form von REST Webservices bereit.

8.1.1 Storage Cloud-Datenverwaltung

Die Storage Cloud-Datenverwaltung ist verantwortlich für die komplette Datenverwaltung nach innen für das Netzwerk und nach außen auf die Storage Clouds. Da Daten einen zentralen Bestandteil des Konzeptes darstellen, ist diese Komponente sehr umfangreich in seiner Ausprägung. Grundsätzlich werden der Zugriff, die Speicherung und die Modifizierung der Nutzerdaten in dieser Komponente durchgeführt. Das Datenmanagement stellt in Bezug auf Aggregation und Verifizierung eine große Herausforderung dar. Besonders der Faktor Zeit ist hier von Interesse. Eine umfassende Beschreibung der Unterkomponenten Aggregationssystem (8.3) und Verifizierungssystem (8.4) erfolgt in den gesonderten Kapiteln. Darüber hinaus wird das Dateispeichersystem (8.2) in einem eigenen Kapitel abgehandelt.

Eine weitere Aufgabe ist die Bereitstellung einer performanten Suche von Daten im Netzwerk. (vgl. SY7) Dies geschieht unter Berücksichtigung sämtlicher Datenschutzaspekte. Daher werden Suchergebnisse, wenn überhaupt, nur auf den Storage Clouds der Nutzer gespeichert. Der Suchbereich umfasst hierbei freie Seiten und freigegebene Daten von bekannten Nutzern. Erweiterte Funktionalitäten müssen durch externe Dienste angeboten werden, bei denen der Nutzer explizit zustimmt, Teile seiner Daten frei zugeben. Wenn der Nutzer am Netzwerk angemeldet ist, werden Suchen automatisiert im Hintergrund durchgeführt, um dem Nutzer zusätzliche Informationen anzubieten.

8.1.2 Rechtemanagement

Das Rechtemanagement ist verantwortlich für die Absicherung der Daten, die im Netzwerk verwendet werden. Primär ist es die Aufgabe des Nutzers, die Kategorisierung seiner Kontakte sowie die Datenschutzeinstellungen zu seinen Daten vorzunehmen. Für eine Steigerung der Sozialisierung des Informationssystems werden bestehende Konzepte der Interak-

tionen und Verhaltensweisen zwischen Menschen in die virtuelle Welt übernommen.
Aus diesem Grund ist das Rechtemanagement als Beziehungsmanagement angelegt. (vgl. CO6) Die Beziehungen zu anderen Nutzern werden mit speziellen Regeln abgedeckt. Abhängig von der Intensität der Beziehung besteht das Recht Daten zu sehen. (vgl. SY6) Es besteht ein enger Zusammenhang zwischen der Beziehungsintensität und dem Vertrauen. Daraus ergeben sich folgende grundsätzliche Annahmen:

- Mit steigendem Vertrauen steigt auch die Intensität der Beziehung
- Mit sinkender Nähe zu einem anderen Teilnehmer sinkt auch das Vertrauen
- Je enger eine Beziehung ist, desto mehr Rechte auf den Zugriff

Die Interaktionsmöglichkeiten basieren somit auf den vorhandenen bzw. vergebenen Rechten der Nutzer. Eine überschaubare Anzahl an Abstufungen von Beziehungstypen senkt die Komplexität und beeinflusst die Fehlervermeidung positiv. So ist es das Ziel, mit einer möglichst geringen Anzahl an Regeln das Repertoire an gewünschten Einstellungen abzudecken. Für eine schnelle Einstufung ist eine mittlere Auswahl an Beziehungsintensität zielführend. Dadurch kann der Nutzer eine schnelle Entscheidung treffen, ohne sich zwischen zwei Einstellungen entscheiden zu müssen. Im Sinne der Privacy by Default wird bei neuen Kontakten die geringste Beziehungsintensität voreingestellt. Damit die Rechtevergabe die zuvor aufgezeigten Ziele erfüllen kann, erfolgt die Ausbildung einer Gruppenhierarchie. Die Gruppierungen von Individuen basieren hierbei auf sogenannten Klassen des Vertrauens. Dieses stufen-basierte Rechtemanagement gruppiert die Teilnehmer, mit denen man in Kontakt steht, in festgelegte Typen. Die Benennung der Stufen erfolgt in jedem Informationssystem kontextabhängig (z. B. Unternehmen und Privatpersonen). Tabelle 30 listet die Stufen mit Beispielausprägungen auf.

Stufen-basiertes Rechtemanagement			
St.	Vertrauen	Beschreibung	Ausprägung
1	kein	Unbekannte Individuen, keinerlei Beziehung	Umwelt, Alle
2	gering	Wenig Rechte Daten zu sehen, nur ausgewählte Daten können gesehen werden	Bekannte, Neuer Kontakt
3	teilweise	Viele Daten werden freigegeben	Freunde, (Geschäfts-) Partner
4	hoch	Daten mit hoher Relevanz werden freigegeben	Gute Freunde, Kollaboration
5	Sehr hoch	So gut wie alles wird freigegeben	Enge Freunde, Verbundenes Unternehmen
6	Absolut	Eigene Sicht	Ich, Wir

Tabelle 30: Stufen-basiertes Rechtemanagement mit Ausprägungen

Daraus lässt sich ein schematisches Kreisdiagramm erzeugen, in dessen Mittelpunkt der Nutzer steht. Je weiter entfernt ein Kontakt vom Mittelpunkt ist, desto weniger Rechte besitzt er und desto schwächer ist die Beziehung und damit das Vertrauen. Abbildung 36 zeigt zwei mögliche Ausprägungen des Stufensystems in Kreisdiagrammform.

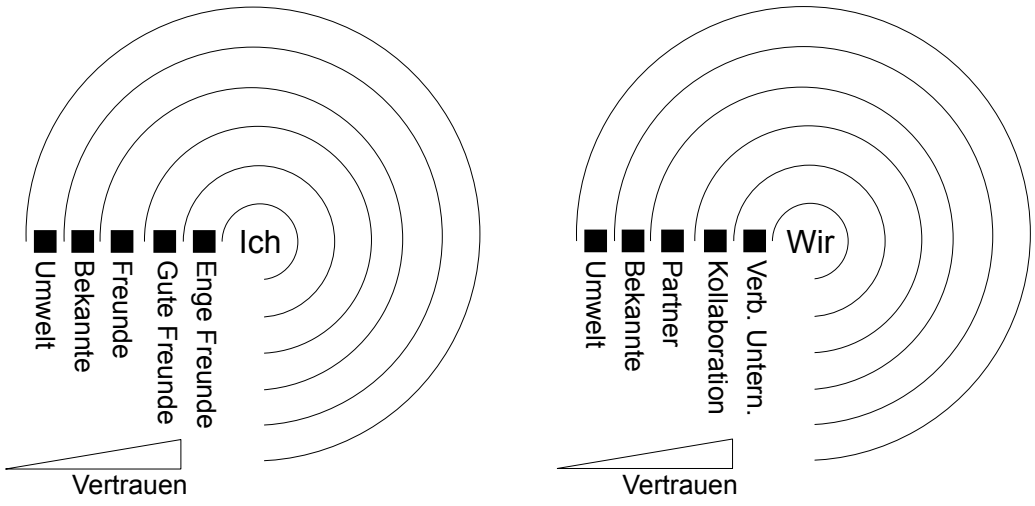


Abbildung 36: Ausprägung der Beziehungen für Personen und Unternehmen

Neben der Vergabe von Zugehörigkeiten zu einzelnen Stufen ist es ebenfalls möglich Gruppen anzulegen. Für die Umsetzung des Rechtekonzeptes ist eine technische Realisierung notwendig. Zunächst kann eine Unterscheidung zwischen zentralen und dezentralen Rechtevergaben getroffen werden. Bei einem zentralisierten Rechtemanagement werden Rechte zentral bei einem Server hinterlegt. Dies hat den Vorteil, dass alle Rechte an einem logischen Ort hinterlegt sind und auf einem Server, dem vertraut werden kann, gespeichert

sind. Angesichts des Konzeptes der dezentralen Datenspeicherung ist diese Vorgehensweise hier jedoch nicht geeignet. Die dezentrale Speicherung von Rechten ist allerdings durch angeheftete Schutzrichtlinien möglich. (vgl. DS6) Hierbei werden zu allen erzeugten Daten Metainformationen angeheftet, welche die Richtlinien/Regeln enthalten. Diese werden in einer speziellen Schicht an dem Datenobjekt gespeichert. Dies wird realisiert mit Hilfe von Metadaten, entweder als Teil der Daten oder angeheftet an den Daten.

Die softwaretechnische Umsetzung ist XML-basiert. Die Extensible Markup Language (XML) ist eine Mensch-Maschinen-lesbare Auszeichnungssprache. Sie dient unter anderem dem Austausch von Daten zwischen Computersystemen. (vgl. [W3C 2008]) Durch die einfache Lesbarkeit von Anwendern eignet sie sich in besonderem Maße als Speicherform von Rechten. Es gibt drei Arten, wie Rechte festgelegt werden können: per Stufen, durch Gruppen und mittels Nutzern. Grundsätzlich können zwei Regeln angewendet werden: Einschließen (engl.: *Include*) und Ausschließen (engl.: *Exclude*). Stufen werden durch Ziffern von 0−5 kategorisiert. Fünf ist die höchste Stufe, was für den Nutzer einfach nachvollziehbar ist. Listing 2 zeigt eine beispielhafte Ausprägung einer Schutzrichtlinie.

```
<rights>
	<!--levels:   0: all, 1: acquaintance, 2: friends, 3: good friends
	4: close friends, 5: myself, (empty): none  -->
	<level>0</level>
	<include>
		<user></user>
	</include>
	<exclude>
		<user></user>
	</exclude>
</rights>
```

Listing 2: Schutzrechteregeln XML-Metadatei

Eine weitere Herausforderung stellt das Konfliktmanagement dar. Bei einer Vielzahl an Regeln und vergebenen Rechten kann es zu sich gegenseitig beeinflussenden Einstellungen kommen. Mit steigender Anzahl an Regeln steigt auch die Wahrscheinlichkeit für Überschneidungen. Daher ist es das Ziel des in dieser Dissertation vorgestellten Konzeptes ein möglichst einfaches Rechtemanagement zu integrieren. Das gewählte Beziehungsmanagement besitzt daher wenige einfache Regeln, um Konflikte zu vermeiden. (vgl. DS3) Somit kann eine einfache Verwendung, einhergehend mit einer geringen Konfliktwahrscheinlichkeit, garantiert werden. Das Konfliktmanagement erfolgt hierbei anhand festgelegter Regeln. Die Befolgung der Regeln geschieht anhand eines Algorithmus. Dabei gilt, dass Beziehungen Vorrang vor Gruppen haben. Zusätzlich können Gruppen und Personen per Aus-

schluss (engl.: *Exclude*) und Einschluss (engl.: *Include*) gezielt verwaltet werden.
Für die Steigerung des Komforts für den Nutzer bei der Verwendung des Netzwerkes werden Empfehlungen für die Einstellungen von Rechten sowie neuen Kontakten angeboten. Diese basieren auf der Häufigkeit, mit der andere Nutzer, welche nicht zu den Kontakten gehören, im Wirkungskreis der eigenen Kontaktliste stehen. Im Sinne des Privacy by Default besitzt ein Erstkontakt immer den Status „Bekannter“, außer der Nutzer nimmt gleich eine Änderung vor. So können im Unternehmenskontext neue Partnerschaften angebahnt werden. Weiterhin werden Empfehlungen für die Höhe an Schutz aus der Anzahl an Kommunikation und Interaktion abgeleitet.

8.1.3 Storage Cloud-Zugriff

Der Storage Cloud-Zugriff ist verantwortlich für den Zugriff auf die Storage Clouds. Beim Einlesen von Daten ist eine hohe Fehlertoleranz sowie Robustheit des Systems gefordert. Durch diese Komponente ist es möglich, Daten zu erstellen, anzupassen und zu löschen. Hierfür werden Softwaretechniken wie REST-API, WebDAV und OAuth für den Zugriff und XML, JSON und HTML für die Antwort eingesetzt. Beim Zugriff auf externe Dienste muss eine jeweilige Anpassung an die Sicherheitsstrategie des Anbieters erfolgen.
Das Ziel ist die Integration einer möglichst großen Anzahl von Anbietern bzw. Techniken. (vgl. SY1) Dies wird zum einen erreicht durch eine zunehmende Standardisierung der Kommunikationstechniken und durch die Einrichtung einer Abstraktionsschicht des Zugriffes innerhalb des Systems. Durch die Entwicklung einer Abstraktionsschicht ist es möglich, eine große Anzahl an externen Quellen für die Datenintegration abzufragen. Da die grundlegenden Funktionen über alle Storage Clouds hinweg gleich sind und nur die technischen Lösungen voneinander abweichen, ist es zielführend, die Zugriffsfunktion zentral zu standardisieren. Anschließend wird für jede Art von externer Quelle eine eigene Ausprägung in Form von Algorithmen vorgenommen. Die in engem Zusammenhang stehenden Konzepte der Aggregation (8.3) sowie der Verifizierung (8.4) werden in gesonderten Kapiteln betrachtet. Abbildung 37 zeigt die grafische Darstellung der Abstraktionsschicht in Verbindung mit den externen Storage Clouds.

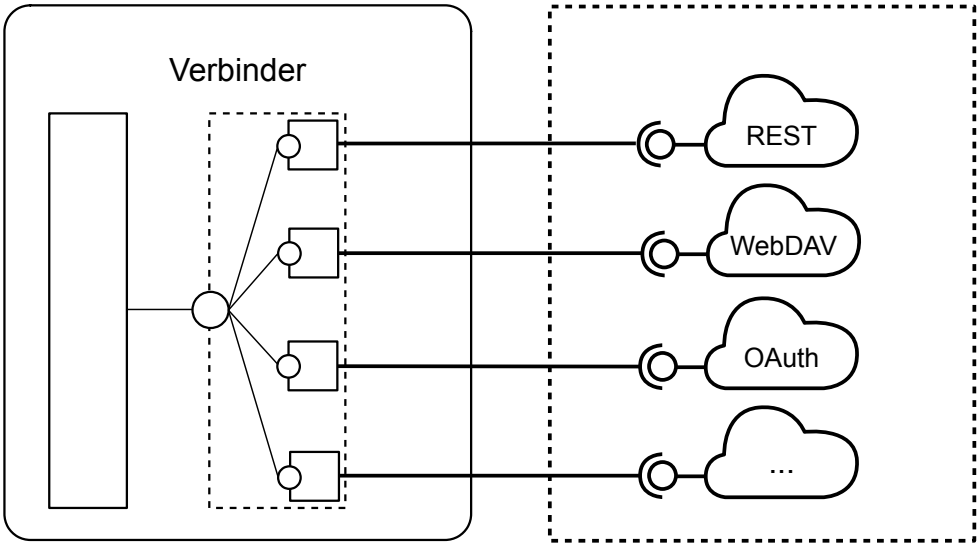


Abbildung 37: Abstraktionsschicht des Storage Cloud-Zugriffs

8.1.4 Kommunikationsschicht

Die Kommunikationsschicht des Verbinders ist durch die an sie gestellten Anforderungen sehr umfangreich. Das Portal stellt Anfragen und erwartet konsolidierte und überprüfte Ergebnisse. Das Nutzerverzeichnis dient der Authentifizierung des Zugriffes auf die Clouds sowie der Übermittlung der Zugriffsdaten. Die Kommunikation zu den Storage Clouds zeichnet sich durch sehr viele unterschiedliche (heterogene) Kommunikationspartner und durch hochfrequente Zugriffe aus.

8.2 Datenspeichersystem

Die primäre Aufgabe des Datenspeichersystems ist die Speicherung der vom Nutzer erzeugten Daten. Es sorgt nicht für die Übertragung der Daten, sondern für die Art und Weise der Speicherung. Für die Verwaltung der Daten enthält jedes Datum für die Identifikation einen eindeutigen Zeitstempel. Unterordner dienen der Strukturierung und Kategorisierung. Für die Einhaltung der Interoperabilität (vgl. SY5) wird zum einen das Speicherformat XML verwendet und zum anderen eine standardisierte Speicherung eingesetzt. XML ist ein im Internet weit verbreitetes und anerkanntes Format für die Speicherung und den Datenaustausch. Dadurch ist es dem Nutzer sowie anderen Systemen leicht möglich, unabhängig vom DCN Daten zu verwenden und zu bearbeiten. Die Standardisierung der Speicherung sorgt für gleichartige Daten auf allen unterschiedlichen externen Quellen. Eine Vielzahl an externen Diensten (Storage Clouds, NAS-Systeme usw.) können so angespro-

chen und verwendet werden, ohne dass das Datenformat geändert werden muss. Abbildung 38 zeigt die Ordnerstruktur innerhalb der Speichermedien.

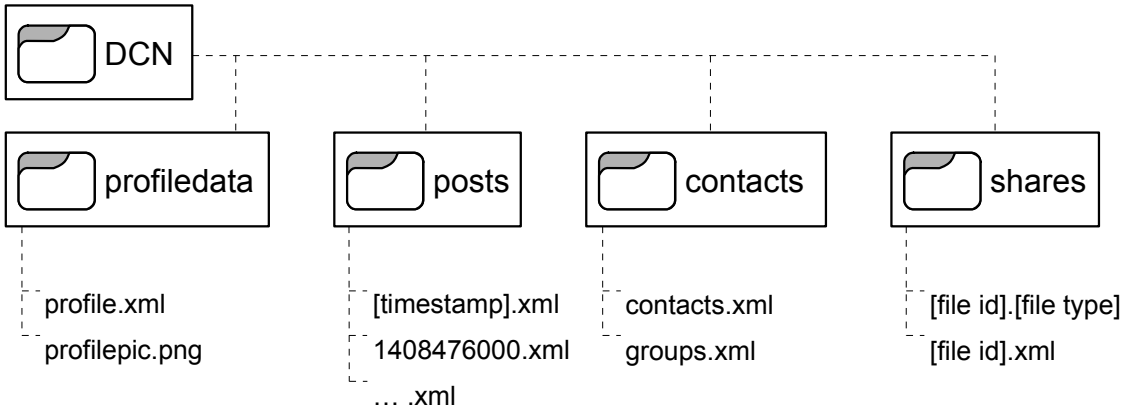


Abbildung 38: Verwendete Ordnerstruktur auf den Storage Clouds

Nachfolgend werden für die Steigerung der Nachvollziehbarkeit zwei Datenbereiche, Kontakte und Gruppen (contacts.xml und groups.xml), näher in deren Aufbau und Struktur erklärt. Der Bereich Kontakte (engl.: *contacts*) ordnet alle Nutzer, mit denen der Anwender in Kontakt steht, in eine Kategorie ein. Hierfür werden die Bezeichner 0−5 gewählt, wobei die Stufe 0 ehemalige Kontakte charakterisiert, welche nicht mehr zu den Bekannten gezählt werden. Der Bezeichner „user“ steht für einen Nutzer des Netzwerkes. Diese werden als eindeutige ID zwischen den Bezeichnern geschrieben. Jeder Nutzer kann hierbei nur einer Stufe angehören. Bei Konflikten gilt die niedrigste Stufe auf der der Nutzer eingeordnet ist (Privacy by Default). Listing 3 zeigt die beispielhafte Ausgestaltung mit XML.

```
<contacts>
        <0>
                <user></user>
        </0>
        <1>
                <user></user>
        </1>
        <2>
                <user></user>
        </2>
        <3>
                <user></user>
        </3>
        <4>
                <user></user>
        </4>
        <5>
                <user></user>
        </5>
</contacts>
```

Listing 3: Kontaktmanagement in XML

Für die Erweiterung des Rechtemanagement ist es dem Nutzer möglich Gruppen anzulegen. Diese Gruppen (engl.: *groups*) werden in einer speziellen Datei hinterlegt. Jede Gruppe erhält hierbei verschiedene Eigenschaften. Zum einen eine Identifikationsnummer, welche vom Netzwerk vergeben wird und einen Namen, den der Nutzer selbst bestimmt. Innerhalb einer Gruppe befindet sich eine Liste an Nutzern, welche dieser Gruppe angehören. Listing 4 zeigt die beispielhafte Ausgestaltung mit XML.

```
<groups>
	<group>
		<id></id>
		<name></name>
		<users>
			<user></user>
		</users>
	</group>
</groups>
```

Listing 4: Gruppenmanagement in XML

8.3 Aggregationssystem

Das Aggregationssystem hat die Aufgabe, Daten aus verschiedenen Datenquellen zusammenzuführen. Hierfür werden parallele Abfragen auf den Storage Clouds durchgeführt und Informationen innerhalb des Verbinders zusammengeführt. Das Ziel ist, die Lade- und Bearbeitungszeiten bis zum Anzeigen des Inhaltes beim Nutzer zu minimieren. (vgl. CO7) Dies wird erreicht, indem Daten, die bis zu einem bestimmten Zeitpunkt nicht abgerufen werden können, nicht in das Ergebnis mit aufgenommen werden. Für eine beschleunigte Fehlererkennung werden die Datenquellen zunächst angepingt[9] und nur bei Rückantwort wird eine Anfrage gestellt.

Die Nichtverfügbarkeit von Daten aus externen Quellen bedarf einer gesonderten Betrachtung. (vgl. SY2) Das Aggregationssystem setzt Datenbestände der Nutzer zusammen und reagiert auf die eventuelle Nichtverfügbarkeit durch Ausblenden dieser Daten. Handelt es sich dabei um eventuell benötigte Daten (Co-Datenschutz), werden diese mit einem Hinweis versehen und ausgeblendet. Fehlerhafte Daten und fehlerhafte Zugriffe werden abgefangen und unterbunden. Dadurch ist eine Steigerung der Robustheit des Systems möglich. (vgl. SY3)

Bei jeder Abfrage wird die Authentifizierung der Anfrage bei dem Nutzerverzeichnis überprüft. Das aus verschiedenen Elementen bestehende Ergebnis, welches Co-Datenschutzaspekte besitzt, wird zusammengesetzt. Hierbei kommt es zu einer Anreicherung der Daten

9 Senden einer kurzen Nachricht an einen Server.

um weitere Informationen. Die gewonnenen Ergebnisse werden anschließend an das Portal zurückgesendet. Grundsätzlich kann die Anzahl an Datenquellen sehr umfangreich sein. Dies hat eine Steigerung der Abarbeitungszeit zur Folge. Das Ziel ist es eine effiziente Aufgabenerledigung zu erreichen.

Abbildung 39 zeigt die schematische Darstellung des Aggregationssystems. Zunächst erfolgt hierbei die Anfrage durch das Portal nach einem bestimmten Datensatz oder nach Daten, die für den Nutzer von Interesse sind (aktuelle Daten). Der Verbinder prüft den mitgegebenen Authentifizierungsschlüssel, indem dieser zu dem Nutzerverzeichnis übertragen wird. Ist die Antwort positiv, wird der Datensatz mit Hilfe einer Ausprägung der Abstraktionsschicht von einer spezifischen Storage Cloud übertragen. Diese Daten werden anhand weiterer Quellen analysiert, wie etwa Nutzernamen oder soziale Erweiterungen. Dies hat zur Folge, dass weitere Anfragen an unterschiedliche Storage Clouds gestellt werden, um die Quellenverweise aufzulösen. Sind alle Daten vorhanden, erfolgt eine Umwandlung des kompletten Ergebnisses in ein Übertragungsformat. Abschließend wird diese Ergebnis an das Portal zurückgesendet.

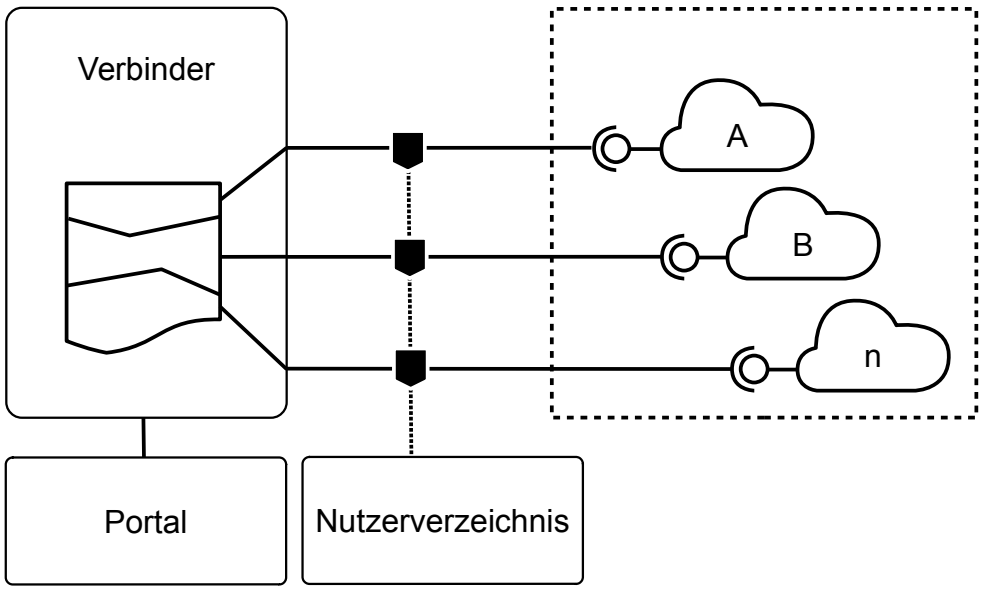


Abbildung 39: Verbinder-Aggregationssystem

8.4 Verifizierungssystem

Das Verifizierungssystem unterstützt in besonderem Maße den Co-Datenschutz (engl.: *Co-Privacy*). Hierbei handelt es sich um die Verwaltung der Datenschutzeinstellungen von

Elementen, die die Intimität von mehreren Individuen beeinflussen (vgl. [Fogues et al. 2015, 9]). Dies lässt sich zutreffend mit dem Satz „Das Ganze ist mehr als die Summe seiner Teile“ beschreiben. Neben der Gestaltpsychologie wurde der Satz bereits von Aristoteles geprägt (vgl. [ARISTOTELES 1907, 129] in einer freien Übersetzung[10]).

Das Ziel ist die Überprüfung der Richtigkeit der Daten im Sinne des Co-Datenschutzes. (vgl. DS7) Einzelne Bestandteile müssen somit immer von dem Nutzer bestätigt werden, dem sie gehören. Ist dies nicht möglich oder bestehen diese Daten nicht mehr bei dem Nutzer, ist deren Gültigkeit nicht mehr gegeben. Grundsätzlich besitzt ein Nutzer das Datum, welches aus mehreren Teilstücken besteht. Dieser behauptet, dass die angegebenen Teilstücke korrekt sind. Um dies zu bestätigen, müssen alle Teilstücke überprüft werden.

Der Ablauf der Verifizierung erfolgt mit der Anfrage des Portals nach einem Element an den Verbinder. Dieses Element kann aus Unterbereichen bestehen (z. B. Kommentare zu einem Beitrag). Das Verifizierungssystem teilt das gesamte Dokument in einzelne Datenbereiche auf, die jeweils einem anderen Nutzer gehören. Durch Abfragen auf den jeweiligen Storage Clouds wird ermittelt, ob dieses Datum bei dem betroffenen Nutzer ebenfalls vorhanden ist. Ist dies der Fall, so ist das Datum verifiziert. Ist es nicht vorhanden, wird es als ungültig deklariert und ausgeblendet. Dies kann dazu führen, dass das gesamte Konstrukt ungültig wird, da es nicht vollständig ist. Die Überprüfung der gesamten Datenbestände kann sehr aufwändig sein, wenn auf sehr viele Storage Clouds zugegriffen werden muss. Dennoch kann nur so der Datenschutz garantiert werden. Eine parallele Ausführung fördert die Geschwindigkeit des Prozesses enorm. Abbildung 40 zeigt eine schematische Darstellung des Verifizierungssystems.

10 „Das was aus Bestandteilen so zusammengesetzt ist, daß es ein einheitliches Ganzes bildet, nicht nach Art eines Haufens, sondern wie eine Silbe, das ist offenbar mehr als bloß die Summe seiner Bestandteile. Eine Silbe ist nicht die Summe ihrer Laute; ba ist nicht dasselbe wie b plus a, und Fleisch ist nicht dasselbe wie Feuer plus Erde. Denn zerlegt man sie, so ist das eine, das Fleisch und die Silbe, nicht mehr vorhanden, aber wohl das andere, die Laute, oder Feuer und Erde. Die Silbe ist also etwas für sich; sie ist nicht bloß ihre Laute, Vokal plus Konsonant, sondern noch etwas Weiteres, und das Fleisch ist nicht bloß Feuer und Erde oder das Warme und das Kalte, sondern noch etwas Weiteres.“ [ARISTOTELES 1907, 129]

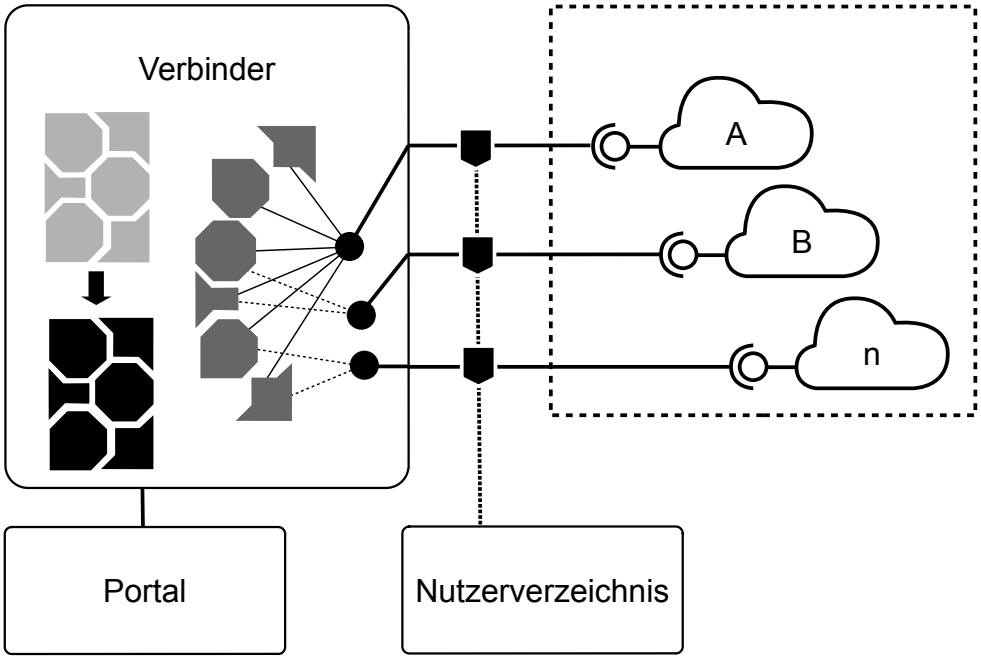


Abbildung 40: Verbinder-Verifizierungssystem

8.5 Zusammenfassung

Dieses Kapitel stellte den Verbinder und dessen Architektur detailliert vor. Weiterhin wurde gezeigt, welche Aufgabe er besitzt und inwieweit Unterkomponenten zur Erfüllung dieser Aufgabe dienen. Es wurde die verwendete Technologie vorgestellt und gezeigt, welche Schnittstellen zu den anderen Komponenten bestehen.

9 Technologische Erweiterungen der DCN-Architektur

Dieses Kapitel stellt eine technologische Erweiterung des in dieser Dissertation entwickelten Konzeptes vor. Das Ziel ist die Performanzsteigerung, vornehmlich durch Skalierung, und die Steigerung der Sicherheit der Anwendung. Hierfür werden aktuelle Technologien und Verfahren der Virtualisierung eingesetzt. Zunächst erfolgt eine Betrachtung der Container-basierten Virtualisierung, deren Einsatz im Kapitel Erweiterung der Architektur beschrieben wird.

9.1 Container-basierte Virtualisierung

Die Container-basierte Virtualisierung ist eine leichtgewichtige Virtualisierung auf Betriebssystemebene. Dadurch ist es nicht notwendig, ein weiteres Betriebssystem (engl.: *Operating System* [OS]) zu starten, wie es bei anderen Virtualisierungen der Fall ist. Die komplette Laufzeitumgebung befindet sich innerhalb eines geschlossenen Containers. Dies stellt somit insgesamt eine besonders effiziente Möglichkeit dar, mit verfügbaren Ressourcen umzugehen. (vgl. [Soltesz et al. 2007, 276 ff.]) Abbildung 41 zeigt einen Vergleich herkömmlicher und Container-basierter Virtualisierung.

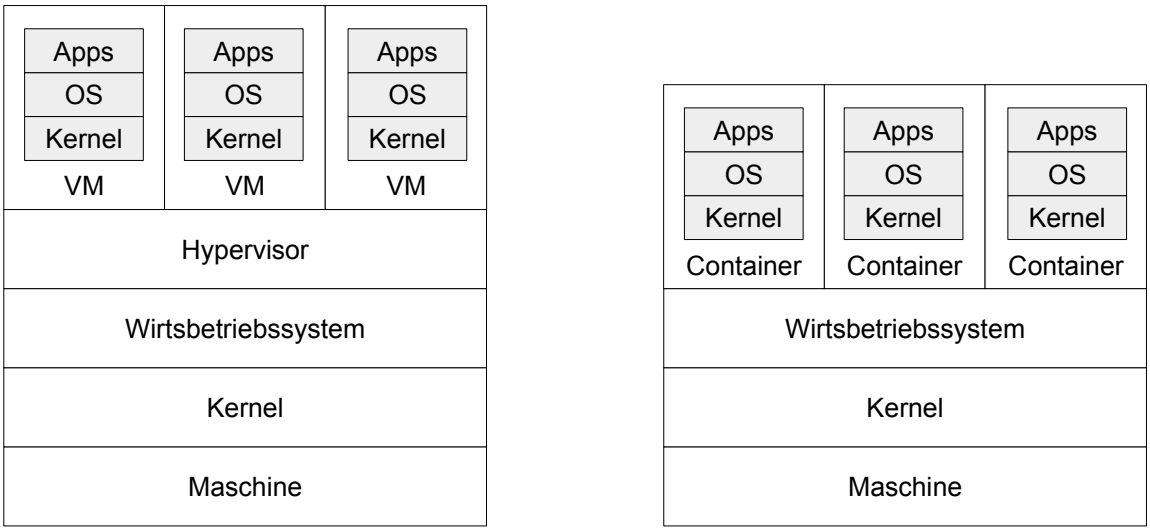


Abbildung 41: Vergleich herkömmliche und Container-basierte Virtualisierung [Newman 2015, 167]

Für eine umfassende Kategorisierung bietet es sich an, Virtualisierungen durch die verwendete Technologie zu differenzieren. Es sind zwei Varianten zu unterscheiden: Typ-1 und Typ-2. Typ-1 beschreibt eine Abstraktion der physischen Hardware-Schicht durch einen

Hypervisor, auch Bare-Metal-Hypervisor genannt. Der Typ-2-Hypersivor virtualisiert auf Betriebssystemebene. (vgl. [Gull 2014, 17 ff.]) Abbildung 42 zeigt den Vergleich zwischen Typ-1- und Typ-2-Hypervisor.

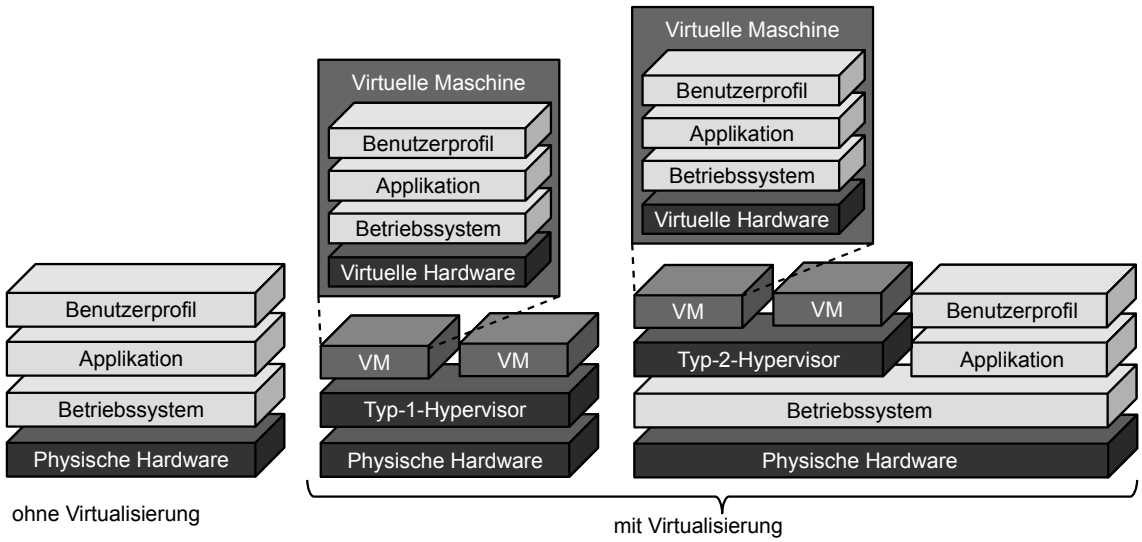


Abbildung 42: Vergleich zwischen Typ-1- und Typ-2-Hypervisor [Gull 2014, 17]

Hierbei ist zu beachten, dass auch Mischvarianten möglich sind. So ist die Virtualisierung eines Betriebssystems innerhalb einer Cloud-Infrastruktur mittels Typ-1-Hypervisor dafür geeignet, oberhalb des OS den Typ-2-Hypervisor zu verwenden. Der Mehrwert von Containern gegenüber bisherigen Virtualisierungen zeigt sich insbesondere in einem Ersetzen von Systembestandteilen zur Laufzeit. Tabelle 31 gibt einen Vergleich beider Varianten.

Vergleich Hypervisor- und Container-basierte Systeme		
Eigenschaften	Hypervisor	Container
Mehrere Kerne	✓	✓
Administrative Befugnisse	✓	✓
Kontrollpunkt & Fortsetzung	✓	✓
Live-Migration	✓	✓
Live-System Update	✗	✓

Tabelle 31: Funktionsvergleich zwischen Hypervisor- und Container-basierten Systemen (vgl. [Soltesz et al. 2007, 278])

Beim Einsatz der Container-basierten Virtualisierung ergeben sich Chancen und Herausforderungen, die im Folgenden aufgezeigt werden. Die primären Chancen liegen im Senken der Kosten und in der Steigerung von Flexibilität (vgl. [Gull 2014, 20 ff.]). Durch die Verwendung von Knoten in einer Cluster-Infrastruktur kann das Fehlermanagement positiv be-

einflusst werden (vgl. [Hindman et al. 2011, 5]). Weiterhin ist eine einfache Administration der Infrastruktur gegeben (vgl. [Gull 2014, 21]). Das Container as a Service-Paradigma erlaubt so eine Steigerung der Wirtschaftlichkeit durch eine schnelle Bereitstellung von benötigter Software (vgl. [Newman 2015, 171]). Demgegenüber existieren eine Reihe an Herausforderungen. Zunächst ist durch eine steigende Dezentralisierung der Infrastruktur mit einer zunehmenden Komplexität zu rechnen (vgl. [Wolff 2015, 25]). Dadurch ist es für den Anwender schwieriger, den Prozess der Bereitstellung nachzuvollziehen. Die Erstellung, Ausführung und Konfiguration von Containern erfordert zusätzlich ein höheres Know-how. Nachdem die technischen Grundlagen vorgestellt wurden, erfolgt im nächsten Abschnitt eine Betrachtung des Microservice-Paradigmas.

9.1.1 Microservice-Paradigma

Microservices sind kleine, eigenständige Services, die kollaborativ sind bzw. sich gegenseitig zuarbeiten [Newman 2015, 22]. Dieses Paradigma richtet sich primär an die Entwicklung von Software zum Bereitstellen von Funktionalitäten für Kunden und Unternehmen. Dies geschieht bei großen Softwareunternehmen durch eine Vielzahl an Teams, die viele Funktionalitäten entwickeln. Gewachsene Strukturen bilden hierbei das Arbeitsumfeld der meisten Entwickler. In den vergangen Jahren waren große Softwareanwendungen, sogenannte Software-Monolithen, der Stand der Technik. Die Forschung betrat hierbei sogar den Weg, mit generischer Softwareentwicklung aus einer Quellcodebasis mehrere angepasste Softwaremonolithen durch Generatoren zu erzeugen (vgl. [Czarnecki/Eisenecker 2000, 131]). Ein Softwaremonolith zeichnet sich, durch eine abgeschlossene ausführbare Softwareeinheit aus. Diese läuft in einem einzigen abgeschlossenen Prozess. Dadurch gibt es eine gemeinsame Entwicklung, sowie ein gemeinsames Deployment (deutsch.: *Softwareverteilung*). Alle Entwicklerteams nutzen hierbei eine gemeinsame Quellcodebasis, das heißt, gemeinsame Klassen, Funktionen und Namespaces. Abschließend erfolgt der Betrieb auf einem festgelegten Technologiestack. (vgl. [Fowler/Lewis 2014]) In der Praxis ergeben sich so diverse Probleme. Die Prozesse Deployment, Test, Abnahme und Release laufen sehr langsam in dessen Iteration ab (vgl. [Wolff 2015, 3]). Weiterhin müssen alle Beteiligten an einem Entwicklungsprojekt und dessen Quellcodebasis arbeiten. Diese beeinflussen sich gegenseitig in den Bereichen Programmierung, Generierung und Auslieferung. Im laufenden Betrieb kann eine Skalierung nur durch das Kopieren der kompletten Instanz erreicht werden. Funktionalitäten die nicht so häufig verwendet werden, werden

ebenfalls dupliziert.

Bei sehr umfangreichen Softwareprojekten erfolgte eine erste Aufteilung der Funktionalitäten in die Bereiche Präsentation, Logik und Datenhaltung. Dadurch ist jeweils ein eigener Technologiestack möglich sowie eine Spezialisierung mit eigener Quellcodebasis.

Der Grund für eben diese Aufteilung kann durch das Gesetz von Conway beschrieben werden. Jede Organisation, die ein System designt, wird ein Design entwickeln, welches einer Kopie seiner eigenen Kommunikationsstruktur entspricht (vgl. [Conway 1968, 28 ff.]). Dies deutet darauf hin, dass die jeweiligen Spezialisten in diesen Bereichen die Entwicklung der Software übernommen haben. Die Gefahr die hierbei entsteht, ist eine sich divergent entwickelnde Subunternehmenskultur, sowie eine sich deutlich unterscheidende Domänsprache in den jeweiligen Bereichen. Ziel sollte es sein, Spezialisten zusammenzubringen, damit die jeweiligen Erkenntnisse gewinnbringend für das gesamte Unternehmen ausgetauscht werden können. Bezogen auf die zukünftigen Herausforderungen, die im Wandel der Digitalisierung an Unternehmen gestellt werden, benötigt es ein Umdenken auf Managementebene. Die Herausforderungen bestehen in den Bereichen: steigende Anzahl an Endgeräten, steigende Funktionalitäten sowie steigende Komplexität, Outsourcing und größer und komplexer werdende Systeme.

Der Microservice Architektur-Stil setzt diesem ein neues Konzept entgegen. So werden Anwendungen als Einzelanwendung, das heißt als Zusammenstellung und Gruppe von kleinen Services designt. Jeder Service ist unabhängig und wird zumeist um eine Geschäftsfunktion entwickelt. Diese kommunizieren untereinander mit einer HTTP-API. Dadurch benötigt es ein Minimum an zentralem Management. (vgl. [Fowler/Lewis 2014])

Microservices besitzen eine Vielzahl an Eigenschaften, welche im Folgenden kurz erwähnt werden. Grundsätzlich werden Microservices durch eine standardisierte Schnittstelle miteinander kombiniert. Als Best Practice hat sich hierbei REST etabliert. Jeder Service ist ein eigener Prozess mit einer eigenen internen Kommunikation. Alle Funktionalitäten werden innerhalb des Service gekapselt. Containier-basierte-Ansätze helfen hier bei der technischen Realisierung. Somit ist ein unabhängiges und vollautomatisches Deployment möglich. Dadurch muss das Gesamtsystem in seinem Betrieb nicht unterbrochen werden, wenn ein Teil des Systems ausgetauscht werden muss. Für die Entwickler bietet sich die Möglichkeit des Einsatzes unterschiedlicher Programmiersprachen und Technologiestacks. Weiterhin wird die Fehlertoleranz wesentlich erhöht. Ein Cluster System verwaltet alle Ser-

vices in Containern und kann bei einem Ausfall einen weiteren Container als Ersatz bereitstellen. Für Performanzaspekte ist bei Bedarf eine Skalierung gegeben. Ein Cluster-System kann Services vollautomatisch duplizieren und verwaltet die Ressourcen sehr effizient. Bei der Konzeption muss darauf geachtet werden, die Datenhaltung so zu entwerfen, dass eine Skalierung einfach möglich ist und diese keine Schwachstelle des Systems darstellt. (vgl. [Newman 2015, 24 ff.]) Abbildung 43 gibt eine Übersicht zu den Eigenschaften von Microservices.

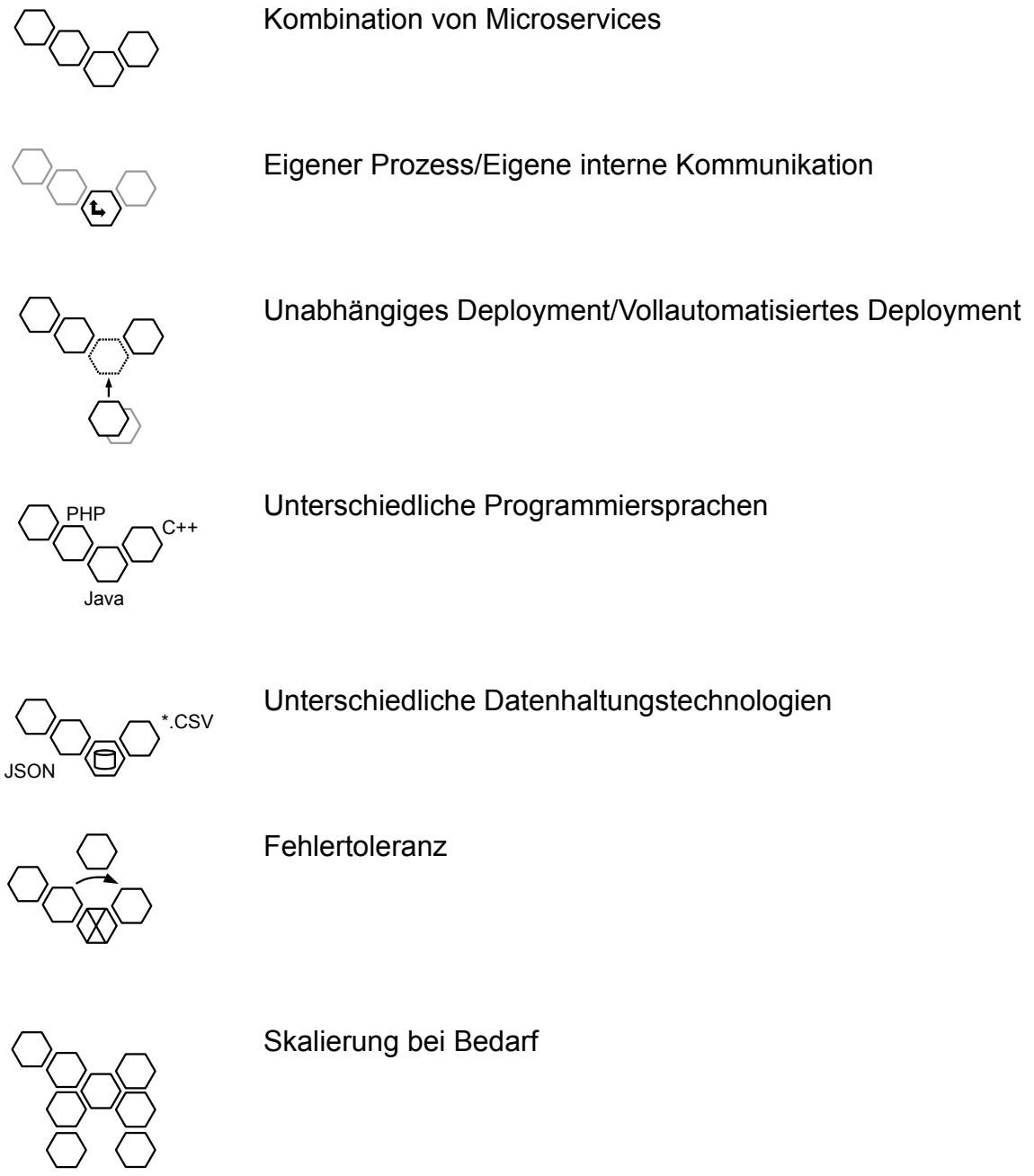


Abbildung 43: Microservice Eigenschaften (vgl. [Newman 2015, 24 ff.])

9.1.2 Cluster-Architektur

Für die Verwaltung der Container wird ein Cluster Management-System eingesetzt. Die Architektur folgt dem Master-Slave-Paradigma. Ein Master-Knoten verwaltet mehrere Slave-Knoten. Dies ermöglicht die Erweiterung der Infrastruktur um weitere Master- und Slave-Knoten, denkbar für die Verbindung von mehreren Rechenzentren. Alle Knoten repräsentieren die physischen Ressourcen in Form von Hardware. Insgesamt erfolgt so eine Abstraktion dieser Ressourcen. Dadurch ist nur eine logische Einheit/Infrastruktur für die Anwender sichtbar. (vgl. Abbildung 44)

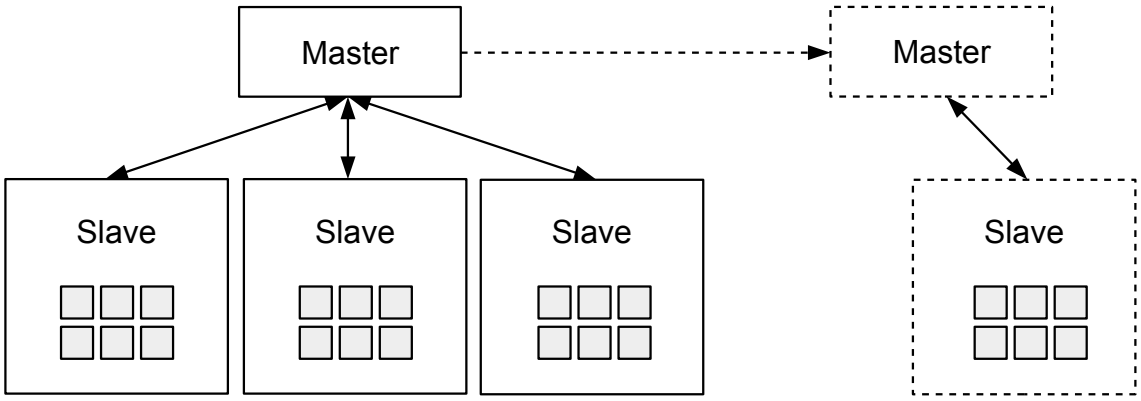


Abbildung 44: Standardisierte Cluster-Architektur

Sadashiv/Kumar [2011, 477] geben eine genaue Beschreibung eines Cluster-Management-Systems. Es handelt sich hierbei um eine Sammlung von parallelen oder verteilten Computern. Diese sind mit Highspeed-Netzwerken verbunden. Cluster werden hauptsächlich verwendet für eine hohe Verfügbarkeit, für Load Balancing und für die Durchführung von komplexen und ressourcenintensiven Berechnungsaufgaben. Das Cluster verhält sich hierbei wie eine Einheit für die Berechnung und verwaltet alle verfügbaren Ressourcen zentral. Weitere Aufgaben sind die Orchestrierung, Beschreibung und Entdeckung (Discovery) von Services und Anwendungen sowie die Netzwerkkommunikation. Im nächsten Abschnitt erfolgt eine Beschreibung der technologischen Realisierung der vorgestellten Konzepte.

9.1.3 Software-technische Realisierung

Im diesem Kapitel wird die technologische Realisierung einhergehend mit Softwareprodukten von Containern und Clustern vorgestellt.

Der Container als technologisches Prinzip findet überwiegend Einsatz auf dem Betriebssystem Linux und dessen Distributionen. Die App Container Specification [2016] dient hierbei als weit verbreiteter einheitlicher Standard zum Beschreiben von Containern. Da-

durch ist es möglich, Container auf verschiedenen Softwareprodukten auszuführen. Beispiele für Container-Technologien sind:

- Docker (containerd, runC)[11]
- rkt[12]
- LCX/LXD[13]
- OpenVZ[14]

Alle Container werden im professionellen Einsatz von einer Cluster-Management-Software verwaltet. Zumeist handelt es sich hierbei um einen Zusammenschluss von verschiedenen Softwarekomponenten für eine einfache und effiziente Nutzung. Beispiele für Cluster-Technologien sind:

- Docker Swarm[15]
- Apache Mesos[16]
- Google Kubernetes[17]
- CoreOS Fleet[18]

Im nächsten Kapitel werden die vorgestellten Paradigmen, Prinzipien und Realisierungen auf die in dieser Dissertation entworfene Architektur angewendet.

11 https://www.docker.com/

12 https://coreos.com/rkt/

13 https://linuxcontainers.org/

14 https://openvz.org/

15 https://www.docker.com/products/docker-swarm

16 http://mesos.apache.org/

17 http://kubernetes.io/

18 https://coreos.com/using-coreos/clustering/

9.2 Erweiterung der Architektur

Die im vorangegangenen Kapitel vorgestellten Technologien bilden die Grundlage für die Erweiterung der im Kapitel 5.2 vorgestellten Architektur. Unter Verwendung der Container- und Cluster-Technologie wird für alle drei Komponenten der Architektur (Portal, Verbinder und Nutzerverzeichnis) das Microservices-Paradigma umgesetzt. Hierbei werden die jeweiligen Stärken der Technologie verwendet, um einen Mehrwert für die Gesamtarchitektur zu realisieren. Insgesamt besteht das Ziel darin, eine Verbesserung bzw. Erweiterung des Konzeptes zu erreichen. Im Fokus stehen hierbei Skalierungs-, Performanz- und Sicherheitsaspekte. Diese Zielstellung ergibt sich aus der Erkenntnis, dass die vorgestellte Architektur eine hohe Zahl an Abfragen und Zugriffen realisieren muss und sich daraus eine hohe Rechenlast ergibt. Im Speziellen werden die Bereiche Storage Clouds, Endgeräte und die Datenbank des Nutzerverzeichnisses adressiert. Im Bereich Storage Clouds geht es dabei um die Verbesserung des Zugriffes auf die externen Datenquellen. Microservices werden eingesetzt für verschiedene Zugriffsarten auf Storage Clouds und als Möglichkeit der einfachen Skalierung, damit eine Vielzahl an Services für eine Vielzahl an Clouds zur Verfügung steht. Das Ziel im Bereich Endgeräte ist die Anbindung einer Vielzahl an Eingabegeräten mit unterschiedlichen Eigenschaften. Auch für die Darstellung der Inhalte für verschiedene Nutzer und deren heterogene Endgeräte werden jeweils Microservices eingesetzt. Im Bereich Nutzerverzeichnis ist die Erhöhung der Sicherheit ausschlaggebend. Für eine gesteigerte Sicherheit wird die Datenbank des Nutzerverzeichnisses durch eine Schicht aus Mircoservices geschützt, welche nicht die kompletten Daten der Datenbank enthalten. Die grundsätzliche Funktionsweise des Netzwerkes wird durch die konzeptionellen Eingriffe nicht verändert. Abbildung 45 zeigt die Erweiterung der Architektur bei jeder Komponente des Netzwerkes. Nachfolgend werden die technologischen Anpassungen im Detail jeweils separat betrachtet.

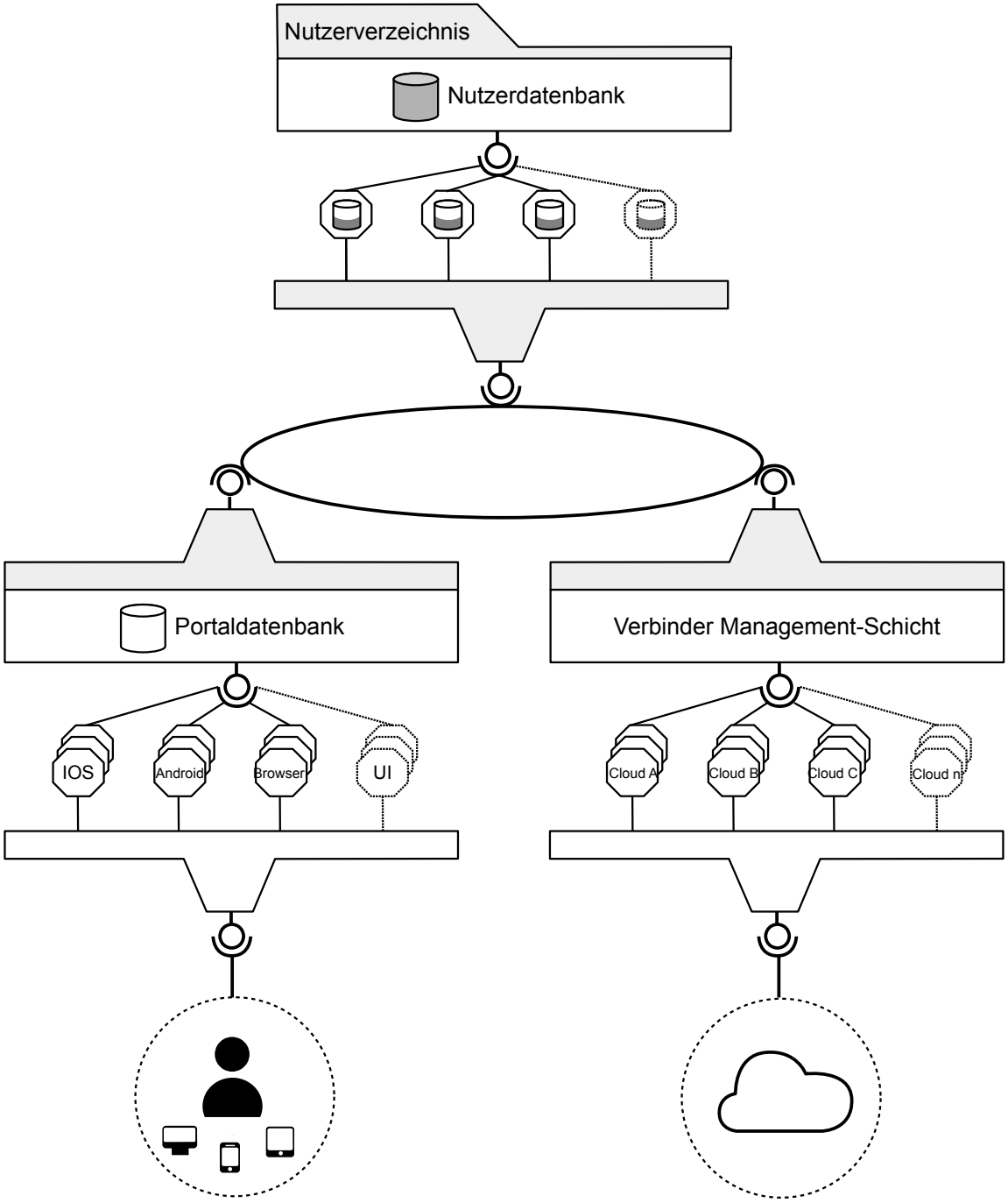


Abbildung 45: Erweiterte Decentral Cloud Network-Architektur

9.2.1 Skalierungs- und Performanzaspekte

Das Kapitel Skalierungs- und Performanzaspekte beleuchtet zum einen das Verbesserungspotential im Bereich der Storage Cloud-Zugriffe und zum anderen die Interaktionsschnittstelle zu den Nutzern basierend auf deren Endgeräten. In beiden Fällen ist von einer hohen Anzahl an Transaktionen auszugehen. Wobei im Bereich der Endgeräte die Heterogenität und deren Beherrschbarkeit im Mittelpunkt stehen und bei den Storage Clouds die hohe Zugriffsfrequenz. Nachfolgend werden Lösungsansätze basierend auf dem Microservice-Paradigma für die Bereiche Endgeräte und Storage Clouds näher erläutert.

Die Verbinder-Komponente realisiert die Kommunikation und Interaktion mit den Datenspeichern der Nutzer. Hierfür benötigt es eine hohe Anzahl an Zugriffen auf die jeweiligen Storage Clouds. Dies ergibt sich aus der Zielstellung des Konzeptes, keine Daten der Nutzer innerhalb des Systems zu speichern. Daher ist für die Speicherung und Ermittlung von Daten in jedem einzelnen Fall ein erneuter Zugriff notwendig. Diese Zugriffsfrequenz wird zusätzlich gesteigert durch das Verifizierungssystem, was das Konzept des Co-Datenschutzes umsetzt.

Aus den aufgestellten Annahmen kann geschlussfolgert werden, dass mit steigender Nutzeranzahl ein Risiko für die Überlastung der Komponente besteht. Dies kann zu sehr langen Antwortzeiten oder sogar zu Abbrüchen in der Kommunikation führen. Ein weiteres Optimierungspotential betrifft die Varianten an Zugriffsarten. Die vielen unterschiedlichen Aufgaben der Komponente, bedingt durch eine Reihe an Standards bei den Zugriffen, führt zu einem relativ umfangreichen und unflexiblen Softwarebestandteil. Aus Softwarearchitektursicht handelt es sich daher um einen typischen Monolithen. Die entworfene Architektur mit einer Dreiteilung wirkte diesem Nachteil bereits entgegen. Das Microservice-Paradigma erlaubt eine zusätzliche Optimierung.

Daher besteht das grundsätzliche Ziel bei der Nutzung der Cluster-Technologie darin, eine Funktions-orientierte Aufteilung in Verbindung mit einer flexiblen Skalierung zu ermöglichen. Die Aufteilung erfolgt hierbei basierend auf den unterschiedlichen Arten an Zugriffen von Nutzerseite bzw. der eingesetzten Technologie. Darauf aufbauend soll sich das System an die aktuell höchste Zugriffsart anpassen.

Für die Realisierung der benannten Zielstellungen wird nachfolgend eine Teilarchitektur für das Gesamtkonzept vorgestellt. Hierbei werden verschiedene Microservices verwendet, um die verschiedenen Storage Clouds zu adressieren. Die Verbinder-Komponente dient als

grundlegende Struktur und kann im softwaretechnischen Sinn als Vorlage betrachtet werden. Darauf aufbauend wird für verschiedene Zugriffsarten, unter anderem WebDAV und OAuth, ein spezialisierte Anwendung entwickelt. Das stetige Voranschreiten softwaretechnischer Produkte aus funktionalen und fehlerbehebenden Gründen sorgt nicht nur für eine Variantenvielfalt der Technologien, sondern auch der Softwareversionen. Da die Aktualisierung der Software durch die Nutzer teilweise nicht durchgeführt wird oder durchgeführt werden kann, besteht die Aufgabe darin, sehr viele Varianten zu entwickeln und anzubieten. Eine spezialisierte Anwendung greift die grundlegenden Verfahren auf und prägt Versionen für die verschiedenen Arten aus. Im Fokus stehen hierbei die Verfahren: WebDAV, OAuth/OAuth2, OpenIDConnect und weitere spezialisierte APIs. Es zeigte sich in ersten Untersuchungen, dass WebDAV ein sehr häufig anzutreffender Standard ist und es ist davon auszugehen, dass dieser Microservice sehr häufig zum Einsatz kommt.

Jede Anwendung wird innerhalb eines bereitgestellten Containers betrieben. Da es sich um einen statischen Service handelt, dass heißt, dass keine Datenbank benötigt wird und es keine Abhängigkeiten zu anderen Services gibt, kann eine einfache Entwicklung einhergehend mit einer einfachen Skalierung durchgeführt werden. Diese erfolgt je nach Bedarf zur Laufzeit der Anwendung. Sollten alle Container belegt sein, kann eine weitere Instanz erzeugt werden. Sollten weniger Services benötigt werden, können einzelne Container gestoppt oder sogar ausgeschaltet werden. Endgültig nicht mehr benötigte Instanzen können zusätzlich gelöscht werden, um Ressourcen freizugeben.

Ein API-Gateway in Verbindung mit einem Load-Balancer verwaltet die Zugriffe und verteilt Anfragen auf passende Container, die für die Bearbeitung zur Verfügung stehen. Die Ergebnisse der Storage Clouds-Anfragen werden abschließend übertragen an die Portal-Komponente. Abbildung 46 stellt die neue Container-basierte Teilarchitektur schematisch dar.

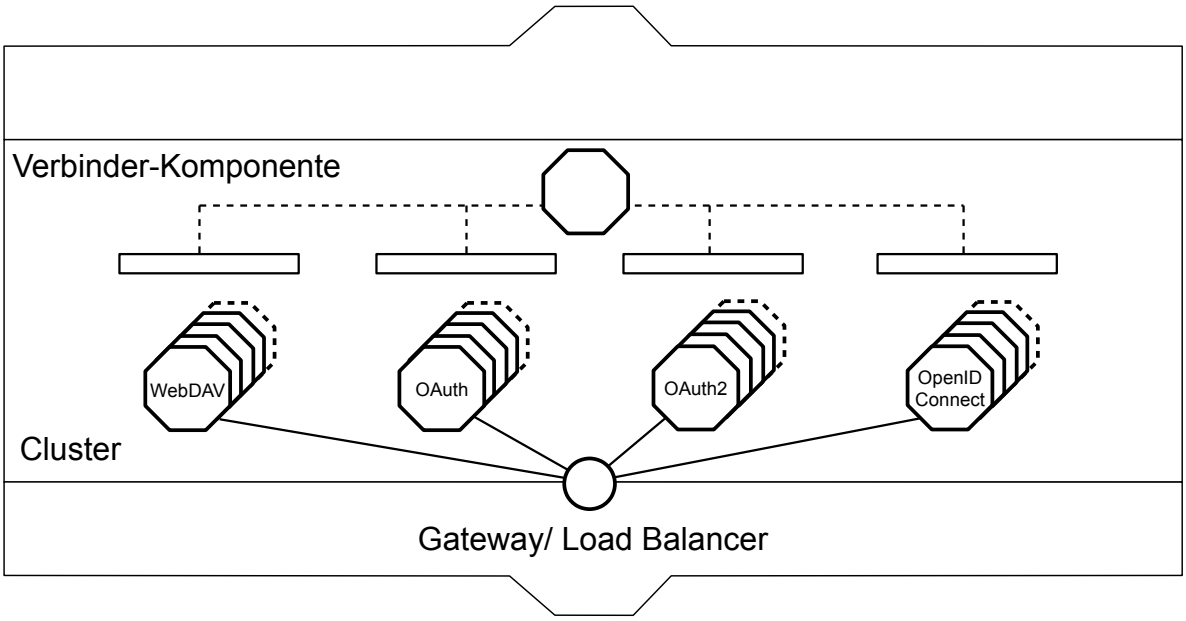


Abbildung 46: Erweiterung Architektur für Storage Cloud-Zugriff

Im Vergleich zur vormaligen Architektur ergeben sich eine Reihe von spezifischen Vorteilen, die nachfolgend kurz erläutert werden. Primär ist die Steigerung der Flexibilität bei neuen Standards bzw. Versionen einer der Hauptvorteile des Einsatzes von Cluster-Technologie. Wird eine Erweiterung der Infrastruktur notwendig, so kann einfach ein neuer Microservice entworfen werden. Dieser kann speziell nach den technischen Vorgaben, der bevorzugten Programmiersprache, den empfohlenen Frameworks und dem geeignetsten Technologiestack entwickelt werden. Gleichzeitig können verschiedene Varianten der Umsetzung einer Verbindung zu den Storage Clouds der Nutzer zur Laufzeit getestet werden, um die sinnvollste und performanteste zu identifizieren. Somit werden stets unterschiedliche Versionen der Zugriffsstandards unterstützt und unabhängig voneinander bereitgestellt. Eine neue Version bedeutet daher innerhalb dieses Konzeptes, dass ein zusätzlicher Service designt wird und der bestehende weiter betrieben wird.

Variantenvielfalt findet sich ebenfalls bei der Portal-Komponente im Bereich des Einbezuges von heterogenen Endgeräten. Die Portal-Komponente muss für verschiedene Endgeräte, welche unterschiedlichste Betriebssysteme in unterschiedlichen Versionen besitzen, eine Benutzerschnittelle zur Verfügung stellen. Hierbei können die Arten in Smartphones, Tablets und Desktop-PCs unterschieden werden. Betriebssysteme sind Android, iOS, Windows Phone und Windows/Linux Distributionen (Browser-basiert) und weitere weniger bekannte Systeme. Das Ziel der Nutzerschnittstelle ist die bestmögliche Darstellung von In-

halten sowie die Verwendung der spezifischen Funktionalitäten der Endgeräte, zum Beispiel Wischbewegungen oder Maussteuerung. Durch eine Individualisierung der Nutzerschnittstelle kann das angedachte Responsive Design gezielt abgelöst werden durch eine besser angepasste Nutzerinteraktion. Daraus ergibt sich folgende Herausforderung: Eine hohe Anzahl an Softwareversionen unterschiedlicher Betriebssysteme weist einen sich gegenseitig überlagernden Softwarelebenszyklus auf. Einzelne Versionen der Anbieter können unterschiedlich lange in Verwendung durch eine Gruppe an Nutzern sein. Die Möglichkeit zur Aktualisierung kann unter Umständen abhängig vom verwendeten Endgerät sein. Abbildung 47 zeigt basierend auf dem Lebenszyklus-Modell das Microservices-Lebenszyklus-Ökosystem.

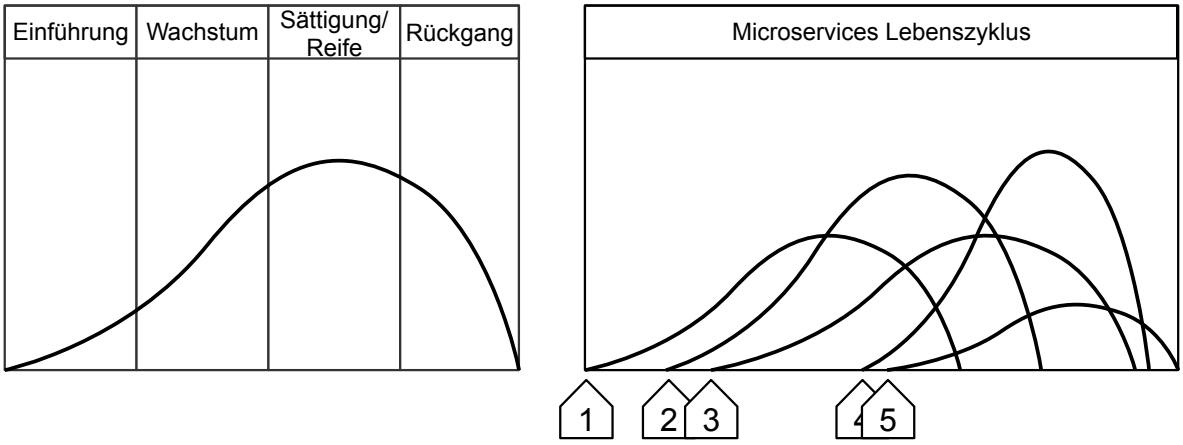


Abbildung 47: Das Lebenszyklus-Modell angewendet auf Microservices (vgl. [Heinrich 2005, 238] und eigene Darstellung)

Es ist zu erkennen, dass die Ablösung von Softwareversionen teilweise erst sehr spät, das heißt deutlich nach der Einführung von neuen Softwareversionen erfolgt. Zu beachten ist hierbei, dass Anwender in regelmäßigen Jahreszyklen Endgeräte ersetzen, und nicht alle Geräte lassen sich stets auf die neue Version updaten. Dadurch muss zu jeder Zeit eine große Anzahl an Softwareversionen unterstützt werden. Daraus ergibt sich das grundlegende Ziel einer effizienten Ressourcenallokation unabhängig von den Betriebssystemen und deren Versionen.

Die Aufgabe des Portals ist die Darstellung von bestehenden Informationen und Daten für die Anwender. Grundsätzlich wird eine Vorlage für jede Art von Betriebssystem und Gerätetyp bereitgestellt, welche als Grundlage für die spezifischen Versionen dient. Daraus werden Unterversionen abgeleitet für Versionen an Betriebssystemen. Weiterhin gibt es eine Standardversion, falls keine spezifische Version gefunden werden kann. Ein Gateway bzw.

Load Balancer verteilt die Anfragen an die jeweiligen verantwortlichen Container. Bei einer hohen Nutzungsfrequenz eines Typs wird dieser als Instanz dupliziert. Es ist weiterhin vorstellbar, dass für jeden Nutzer ein eigener Microservice gestartet wird, welcher alle Session-Daten enthält und nach der Verwendung gelöscht wird. Dies würde zusätzlich den Datenschutz positiv beeinflussen. Abbildung 48 zeigt die angepasste Architektur für die Portal-Komponente.

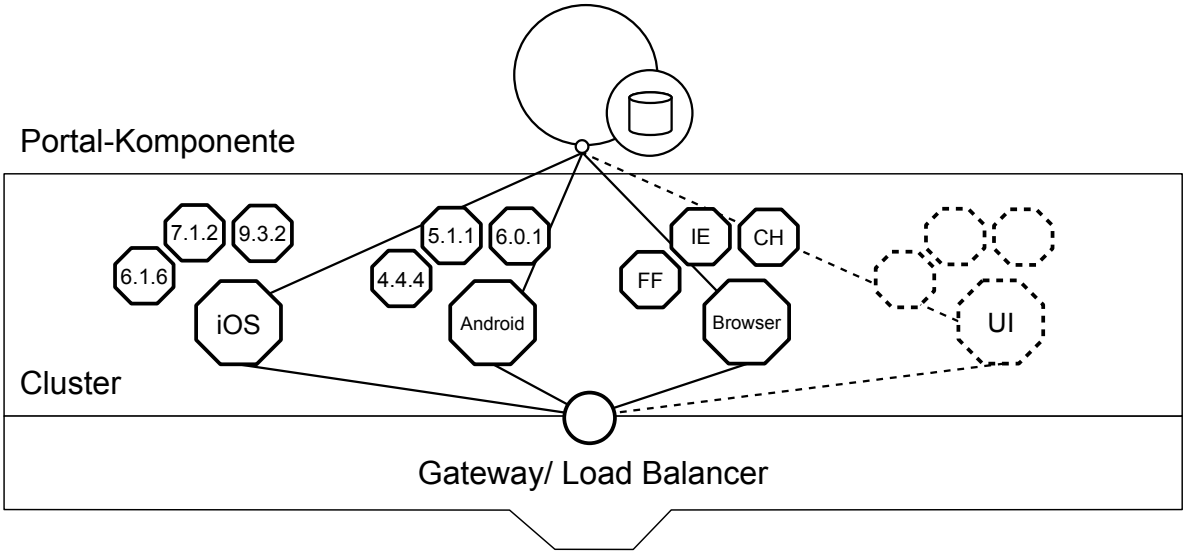


Abbildung 48: Erweiterung der Architektur für GUI

Im Vergleich zur vorhergehenden Architektur kann anstatt des Einsatzes von Responsive Design, gezielt für jeden Typ an Endgerät eine angepasste Darstellung und ein Nutzerinterface bereitgestellt werden. Bei diesem Konzept wird nur die Benutzerschnittstelle durch den Einsatz der Cluster-Technologie angepasst, die restliche Portal-Komponente bleibt unverändert.

9.2.2 Sicherheitsaspekte

Neben positiven Effekten auf die Ressourcenallokation kann das Microservice-Paradigma ebenfalls die Sicherheit von Anwendungen steigern. In diesem Kapitel wird ein Konzept vorgestellt, welches Cluster-Technologie und Container verwendet, um gezielt die bestehende Architektur zu erweitern. Im Mittelpunkt steht die Einrichtung einer demilitarisierten Zone durch die Auslagerung von Teilen der Datenbank des Nutzerverzeichnisses in Container. Die Hauptdatenbank bleibt bestehen und hält weiterhin alle Daten der Nutzer. Die Container halten die aktuell benötigten Daten für das Netzwerk. Somit können Container einen Beitrag leisten für eine Gesamtlösung mit einer sicheren Infrastruktur. Die Erstel-

lung einer Zwischenschicht dient hierbei als Schutz vor Eindringlingen in die eigentliche Infrastruktur. Durch regelmäßiges Löschen von Daten, die lange nicht mehr verwendet wurden, ist bei einem Angriff nur ein Teil der Datenbank betroffen. Die Früherkennung von Angriffen kann durch verschiedene Verfahren geschehen. Unter anderem durch die Mustererkennung von Zugriffen auf die Datenbank. Als Beispiel eines Musters der Zugriffsverteilung kann hierbei die Google-Websuche dienen. Abbildung 49 zeigt den normierten Zugriff pro Tag der Google-Websuche.

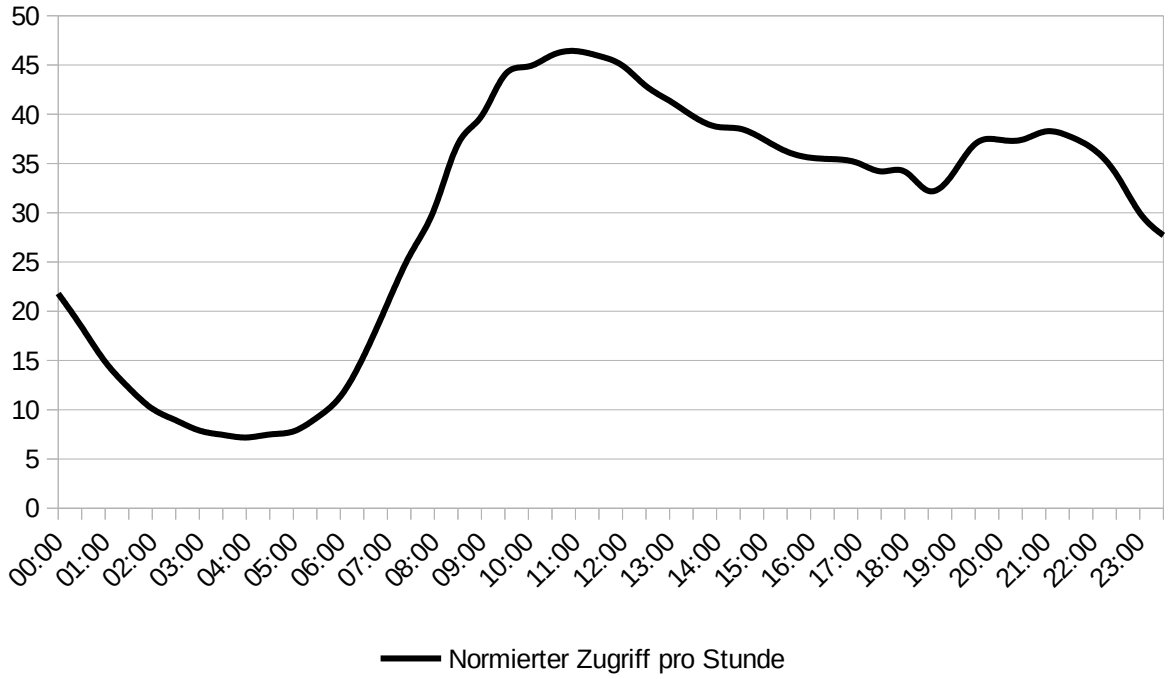


Abbildung 49: Normierter Zugriff pro Tag der Google-Websuche (16.02.2016) [Google 2016]

Eine Abweichung von diesem Muster kann als Hinweis für eine Unregelmäßigkeit gedeutet werden. Basierend darauf können gezielt Gegenmaßnahmen ergriffen werden zum Beispiel das kurzzeitige Sperren der Datenbank. Bezogen auf das Nutzerverzeichnis wird nachfolgend das Konzept der Microservice-Datenbankverteilung vorgestellt. Im Mittelpunkt steht dabei die parallele Auslagerung von Teilen des Hauptdatenbankinhaltes an Container. Daten, die lange nicht mehr abgerufen wurden, werden aus den Container-Datenbanken gelöscht.

Das Ziel dieser angepassten Architektur ist, neben einer Steigerung der Sicherheit, eine einfache Skalierung der verwendeten Container nach Bedarf, basierend auf der Frequenz an Zugriffen. Alle Container halten dabei dieselben Daten der aktuell aktiven Nutzer des Netzwerkes. Daten, auf welche seit einer zuvor festgelegten Zeit nicht mehr zugegriffen

wurde, werden parallel aus allen Containern entfernt. Denkbar ist ebenfalls, Gruppen an Containern zu bilden welche die gleichen Daten halten, zum Beispiel kategorisiert nach Ländern und Regionen.

Zu keiner Zeit sind somit alle Daten der Datenbank innerhalb des Netzwerkes verfügbar. Nur die Datenbank-Container haben das Recht, auf die Datenbank des Nutzerverzeichnisses direkt zuzugreifen. Ein Status-Container überwacht dabei den Inhalt aller Container. Dieser weist die Container an, Daten zu löschen, falls diese nicht mehr benötigt werden. Weiterhin sorgt er dafür, dass alle Container jederzeit denselben Datenbestand aufweisen. Werden Daten für den Betrieb des Netzwerkes benötigt, erfolgt die Anfrage direkt an einen der zur Verfügung stehenden Container. Ein Gateway bzw. Load Balancer sorgt für eine gleichmäßige Verteilung der Anfragen. Sind die angefragten Daten vorhanden, werden diese übertragen. Sind Daten nicht vorhanden werden diese aus der Datenbank des Nutzerverzeichnisses geladen. Zusätzlich erfolgt eine Information an den Status-Container, damit dieser alle anderen Container anweist, ebenfalls diese Daten zu laden. Abbildung 50 zeigt die angepasste Teilarchitektur für das Nutzerverzeichnis.

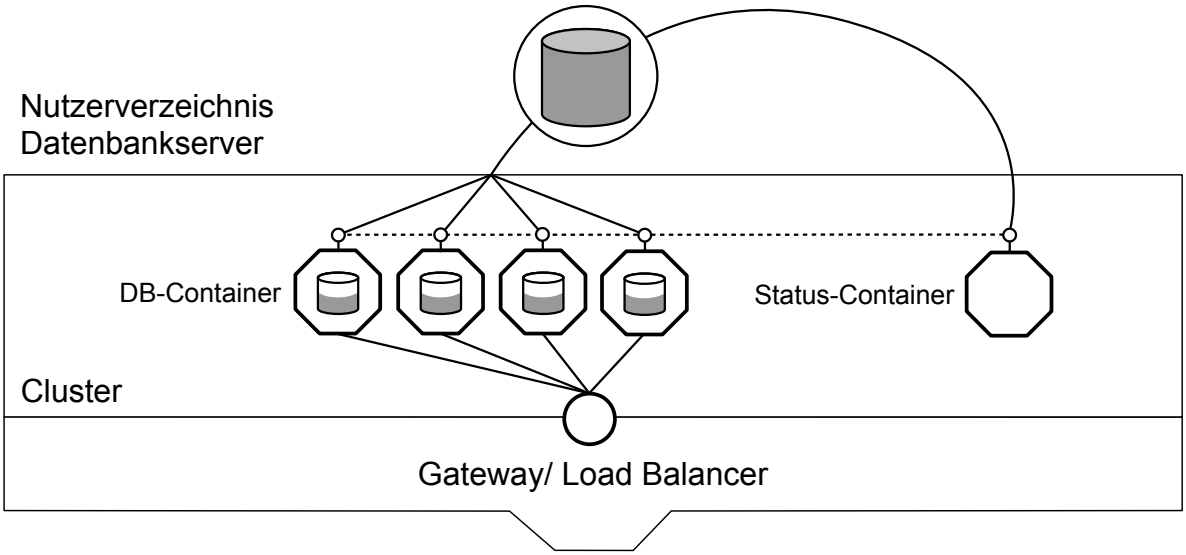


Abbildung 50: Erweiterung Architektur für Nutzerverzeichnis

Im Vergleich zur vorhergehenden Architektur wurde die Sicherheit deutlich gesteigert, da nicht auf alle Inhalte der Datenbank zugegriffen werden kann. Durch die Überwachung der Zugriffe können frühzeitig Gegenmaßnahmen zum Schutz ergriffen werden, bevor es zu einer Kompromittierung der Datenbank gekommen ist.

9.3 Zusammenfassung

In diesem Kapitel wurde die Container-basierte Virtualisierung in Verbindung mit der Cluster-Technologie vorgestellt und für die Erweiterung der bestehenden Architektur eingesetzt. Es konnte gezeigt werden, wie dadurch Potentiale im Bereich Performanz, Skalierung und Sicherheit erreicht werden.

Teil III: Evaluation

Die Evaluation stellt den Übergang von der Entwurfsphase hin zur Evaluationsphase dar. Sie dient einer sach- und fachgerechten Untersuchung und Bewertung der erstellten Artefakte. Damit ist der Nachweis gegeben, dass die erhobenen Thesen erfüllt und die gewonnenen Erkenntnisse zur Festigung und Übertragung in die gesamte Wissensbasis geeignet sind. Dies schließt ebenfalls eine eventuelle Falsifizierung mit ein.
Die Evaluationsmethodik orientiert sich hierbei an dem verwendeten Forschungsrahmen Design Science. Drauf basierend ergibt sich aus der Auswahl der Entwurfsmethodik (Konstruktion eines IT-infrastrukturellen Modells) eine bedarfsgerechte Evaluationsmethodik: der Machbarkeitsnachweis in Form eines Proof of Concept. Dieser ist ausgeprägt als prototypische Implementierung des konstruierten Architekturmodells. Dadurch wird der grundsätzliche Nachweis der Machbarkeit und Umsetzbarkeit des Konzeptes erbracht, einhergehend mit dem Aufzeigen der Erfüllung aller Anforderungen.

10 Proof of Concept

10 Proof of Concept

In diesem Kapitel erfolgt zunächst eine Definition und Abgrenzung von Begrifflichkeiten, welche in diesem Abschnitt verwendet werden. Im Fokus steht hierbei die Machbarkeitsstudie, welche einen Prototypen (Machbarkeitsnachweis) beinhaltet. Dieser wird als konkrete Ausprägung der Machbarkeit im Sinne eines Proof of Concept gesehen. Weiterhin werden die damit verbundenen Ziele und der Ablauf kurz beschrieben.

Die Machbarkeitsstudie, auch Vorstudienphase genannt, dient dem Festellen der realistischen Durchführbarkeit der Problembearbeitung. In dieser werden unter anderem die technische, wirtschaftliche und politische Realisierbarkeit (Machbarkeit) dargelegt. Aufgaben dieser Phase umfassen die Problemerfassung, die Erarbeitung sowie die Festlegung von Zielen, die Diskussion grundsätzlicher Lösungsrichtungen und das Vorschlagen einer Lösungsrichtung. (vgl. [Kuster et al. 2011, 20 f.]) Die Machbarkeitsstudie im Bereich Software- und Konzeptentwicklung beinhaltet zumeist einen Realisierbarkeitsnachweis in Form eines Prototypen. Die zunächst grundsätzlich entworfenen Prinzipien dieser Arbeit sowie deren Evaluation (Proof of Principal) wurden bereits durch konzeptionelle Entwürfe und umfangreiche Belege durchgeführt.

Der Proof of Concept belegt die praktische Durchführbarkeit eines Vorhabens (Machbarkeitsnachweis). Dies kann erreicht werden durch die Entwicklung eines Prototypen, welcher alle benötigten Kernfunktionalitäten beinhaltet. Ziele dieses Vorgehens sind die Risikominimierung für Entscheidungsgrundlagen, die Validierung kritischer Anforderungen an die Anwendung und ein Akzeptanztest der Anwendung in Zusammenarbeit mit Herstellern und Partnern. (vgl. [Rat für Forschung und Technologieentwicklung 2013, 2]) Im Bezug auf diese Arbeit wird dieser Nachweis durch einen Softwareprototypen erreicht, welcher die technische Machbarkeit isoliert betrachtet.

Die angestrebten Ziele des in dieser Arbeit entwickelten Prototypen umfassen die Evaluation der erstellten Architektur durch eine technische Umsetzung sowie das Aufzeigen der Realisierbarkeit der aufgestellten Anforderungen. Hierfür wird zunächst ein Anwendungsszenario als Ausprägung des erstellten Konzeptes der Architektur entworfen. Dies ermöglicht die Entwicklung einer Software. Abschließend wird die lauffähige Anwendung durch Testen auf die Erfüllung sämtlicher Anforderungen geprüft. Nachfolgend wird ein Anwendungsszenario vorgestellt, welches es ermöglicht, einen Prototypen zu realisieren.

10.1 Anwendungsszenario

Diese Dissertation folgt dem Grundgedanken ein Szenario zu entwerfen, welches als Fundament diverser domänenspezifischer Umgebungen dient. Dies kann erreicht werden, indem ein möglichst domänenunabhängiges Anwendungsbeispiel zum Einsatz kommt, welches um individualisierte Funktionalitäten erweitert werden kann. Die bereits thematisierte Sozialisierung moderner Informationssysteme, welche den Menschen stärker in den Mittelpunkt der Betrachtung rücken, stellt das Leitmotiv dieses Szenarios.

Durch die zunehmende Transformation traditioneller Geschäftsumgebungen, durch neue Technologien entstanden neue Ansätze unter dem Namen Smart Business Ecosystems. Geschäftsökosysteme (engl. *Business Ecosystems*) zeichnen sich durch eine ökonomisch geprägte Gemeinschaft aus Organisationen und Individuen der Geschäftswelt aus, welche Güter und Dienstleistungen gemeinschaftlich erstellen und anbieten. (vgl. [Moore 1997, 26]) Das Ökosystem beschreibt hierbei ein Netzwerk verschiedener Teilnehmer. Sowohl zwischen Unternehmen, als auch zu den Kunden hin entstehen vielfältige Beziehungen. Somit zeichnen sich moderne Geschäftsnetzwerke durch eine steigende Interaktion und Kommunikation aus und können damit im weiteren Sinne auch als soziale Netzwerke interpretiert werden. Aus diesem Gedanken heraus hat das Anwendungsszenario die Form eines sozialen Netzwerkes. Dies erlaubt eine vollständige Evaluation aller aufgestellten Anforderungen. Konzeptionell wird eine zentrale Verwaltung als Trusted Party (deutsch: *vertrauenswürdige Instanz*) etabliert, welche die Daten der Anwender extern in Storage Clouds gespeichert und bei Bedarf in das Netzwerk integriert.

10.2 Prototypische Implementierung

Die prototypische Implementierung dient der Umsetzung und Evaluation der entworfenen DCN-Architektur. Das Ergebnis ist eine Webanwendung, welche über eine Webadresse erreichbar ist. Die Entwicklung basiert hierbei auf dem Einsatz von verschiedenen Komponenten, welche über REST-Schnittstellen miteinander kommunizieren. Nachfolgend werden Ziel und Umfang, Infrastruktur und Software, Funktionalitäten, Software-struktureller Aufbau sowie die Umsetzung aufgezeigt.

10.2.1 Ziel und Umfang

Ziel des Prototypen ist die Abbildung aller notwendigen Kernfunktionalitäten für das Aufzeigen der Erfüllung aller aufgestellten Anforderungen. Basierend auf dieser Zielstellung, wird der Umfang der Anwendung auf ein zweckentsprechendes Maß begrenzt. Der festge-

legte Umfang an Nutzern ist mit sechs Nutzerkonten ausreichend groß, um die Interaktion zwischen Anwendern nachzustellen. Jeder Nutzer erhält eine eindeutige Identifikationsnummer und die entsprechenden Daten für den Zugang zu den Storage Clouds. Primär wird eine eigens betriebene ownCloud[19]-Instanz verwendet. Dies ermöglicht eine umfassende Kontrolle während der Entwicklung. Darüber hinaus wird eine externe Dienstleistung in Anspruch genommen, die MagentaCLOUD[20]. Dieser Cloud-Dienst der Deutschen Telekom[21] ist kostenlos und bietet den Vorteil, das alle Server innerhalb Deutschlands betrieben werden. Dadurch kann in besondere Weise die einfache Benutzung durch unerfahrene Nutzer nachgewiesen werden.

10.2.2 Infrastruktur und Software

Nachfolgend werden die verwendete Infrastruktur sowie die eingesetzten Technologien näher erläutert. Diese sind in folgende Bereiche gegliedert: Hardwareinfrastruktur, Betriebssystem, Softwareinfrastruktur, Anwendungsinfrastruktur und Anwendung. Grundsätzlich wurde darauf geachtet, nur Open-Source-Lösungen zu verwenden, damit keine Lizenzkosten anfallen und um eventuelle Abhängigkeiten zu vermeiden.

Die Hardwareinfrastruktur wird realisiert durch den Einsatz von Virtualisierung. In diesem speziellen Fall wird VMWare als Typ-1-Hypervisor verwendet. Dadurch ist eine flexible Skalierung sowie eine einfache Ressourcenallokation möglich. Jede Komponente erhält hierbei eine eigene virtuelle Maschine. Auf dieser ist als Betriebssystem Ubuntu in der Version 16.01.1 LTS installiert. Linux eignet sich besonders gut, Serverinfrastrukturen zu betreiben und ist frei verfügbar. Die Softwareinfrastruktur ist aufgeteilt in den Bereich Anwendungsserver und Datenbankserver. Die Komponenten unterscheiden sich hierbei, da der Verbinder keine Datenbank benötigt. Als Anwendungsserver wird Wildfly in der Version 8.1.0 verwendet. Dieser ermöglicht es, Java-Anwendungen nach dem Java-EE-Standard auszuführen. Bei diesem Prototyp handelt es sich um Webanwendungen, die in einem Web Application Archive (WAR) zusammengefasst sind und vom Server ausgeführt werden. Der Datenbankserver ist mit MySQL realisiert. Für die Anwendungsinfrastruktur findet Java 1.7 Einsatz, ergänzt durch das Java Spring Framework[22], welches eine einfache Ent-

19 https://owncloud.org/

20 https://cloud.telekom-dienste.de/

21 Deutsche Telekom AG: Deutsches Telekommunikationsunternehmen mit Sitz in Bonn, europaweit größtes Unternehmen dieser Art

22 Java Spring Framework: Quelloffenes Framework für die Java-Plattform unter der Apache Lizenz

wicklung von Webanwendungen ermöglicht. Die Anwendungsschicht beinhaltet die drei Komponenten: Java-Anwendung Portal, Java-Anwendung Nutzerverzeichnis und Java-Anwendung Verbinder. Abbildung 51 zeigt eine Übersicht zu dem soeben beschriebenen Prototyp-Stack.

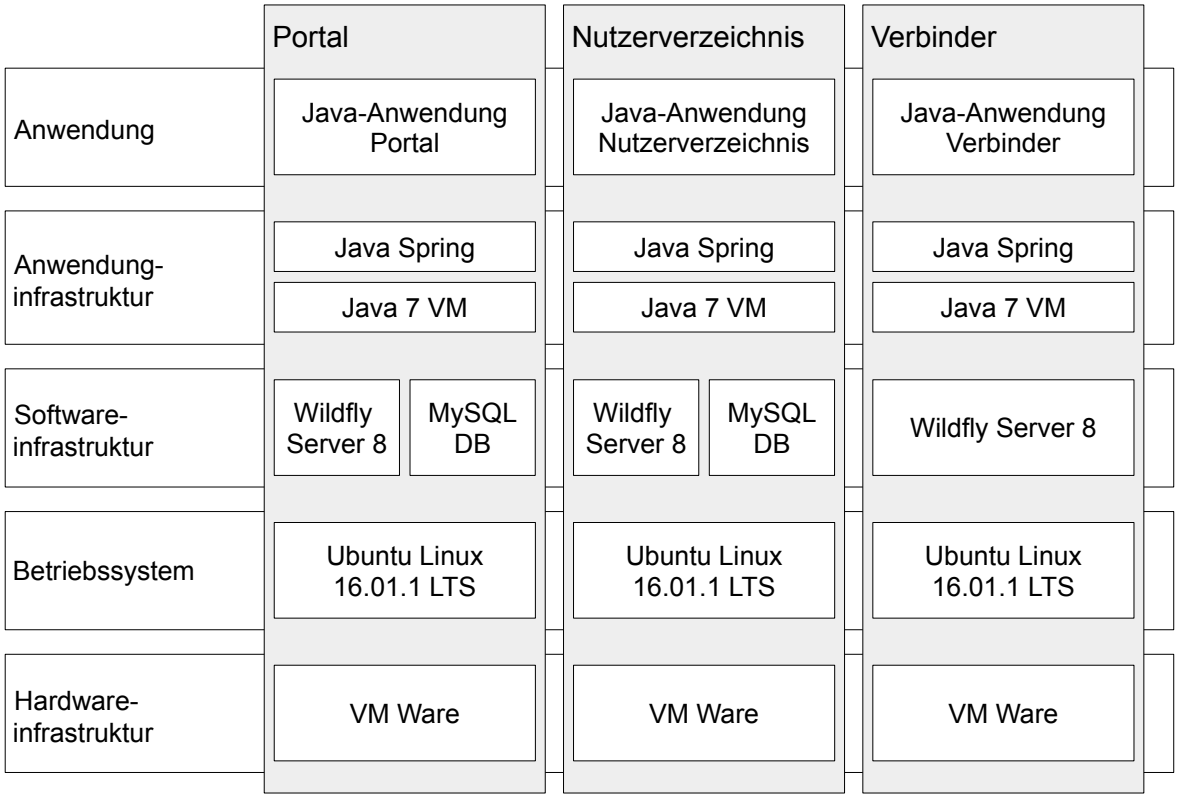


Abbildung 51: Prototyp-Stack der drei Komponenten

Mit Hilfe der aufgezeigten Infrastruktur ist es möglich, die benötigten Funktionalitäten zu realisieren, welche der Evaluation des Konzeptes dienen. Im nächsten Abschnitt werden diese aufgelistet und beschrieben.

10.2.3 Funktionalitäten

Die umgesetzten Funktionalitäten wurden festgelegt mit dem Ziel, die Realisierbarkeit des Konzeptes nachzuweisen. Es wurde darauf geachtet, nur jene Funktionen zu realisieren, welche für die Evaluation notwendig sind. Tabelle 32 listet alle umgesetzten Funktionalitäten mit einer Beschreibung auf. Im Mittelpunkt steht hierbei die Umsetzung eines Beitragsmanagementsystems, welches der Interaktion der Nutzer untereinander dient. An diesem Szenario lässt sich der überwiegende Teil an aufgestellten Anforderungen nachweisen.

Realisierte Funktionalitäten	
Name	Beschreibung
Kontoverwaltung	• Anmelden/Abmelden • Verwaltung Cloud-Zugriffsdaten
Beitragsverwaltung	• Beiträge erstellen, löschen • Beitragseinstellungen
Kommentare	• Kommentieren von Beiträgen
Bewertung	• „Gefällt mir“-Angaben für Beiträge und Kommentare
Nutzerverwaltung	• Nutzer und deren Cloud-Zugriffsdaten werden in einer Komponente verwaltet
Cloud-Zugriff	• Kommunikation zu verschiedenen Storage Clouds
Speicherverwaltung	• Speichern von Daten auf den Storage Clouds
REST-Kommunikation	• Kommunikation zu den jeweiligen Komponenten per REST
Rechteverwaltung	• Mehrstufiges Rechtekonzept als angeheftete Schutzrichtlinien
Freundeverwaltung	• Freundesliste • Freunde hinzufügen und Freundschaft beenden

Tabelle 32: Realisierte Funktionalitäten des Prototypen

10.2.4 Software-struktureller Aufbau

Dieses Kapitel beschreibt den Software-strukturellen Aufbau des Prototypen. Hierfür werden die Paketstrukturen der jeweiligen Anwendungen und Bibliotheken dargestellt. Die Komponenten werden basierend auf dem Architekturkonzept in drei Anwendungen gegliedert: **scn2_portal** für das Portal, **scn2_connector** für den Verbinder und **scn2_directory** für das Nutzerverzeichnis. Daraus werden lauffähige Anwendungen erstellt, welche auf den bereitgestellten Servern veröffentlicht werden. Für eine umfassende Entwicklung der Gesamtstruktur wurden drei Bibliotheken entworfen, welche in die jeweiligen Komponenten integriert werden. Die Bibliothek **scn_communication_classes** stellt Kommunikationsklassen für die Interaktion und Kommunikation der Komponenten untereinander bereit. Mit Hilfe dieser werden Daten in Form von Instanzen zwischen den Elementen ausgetauscht und eine fehlerhafte Kommunikation vermieden. Das sogenannte Mapping wird im Kapitel Umsetzung nochmalig detailliert beschrieben. Ebenfalls wird die Kommunikation zu den Storage Clouds, in diesem Fall in Form einer WebDAV-Schnittstelle ermöglicht. Die Bibliothek **scn_management** beinhaltet Funktionen und Methoden, welche häufig in den jeweiligen Komponenten Anwendung finden. Dadurch werden gezielt die Wiederverwendung sowie die Fehlervermeidung unterstützt. Ein Beispiel ist die Integration einer angepassten Logger-Klasse für die Ausgabe von Informationen auf die Konsole. Die Biblio-

thek **scn_rest_webservices** erzeugt URIs bzw. URLs für die REST-Kommunikation. Alle Bibliotheken werden von den Komponenten während des Kompilierens integriert. Abbildung 52 zeigt eine schematische Paketdarstellung der Struktur des Gesamtprototypen auf.

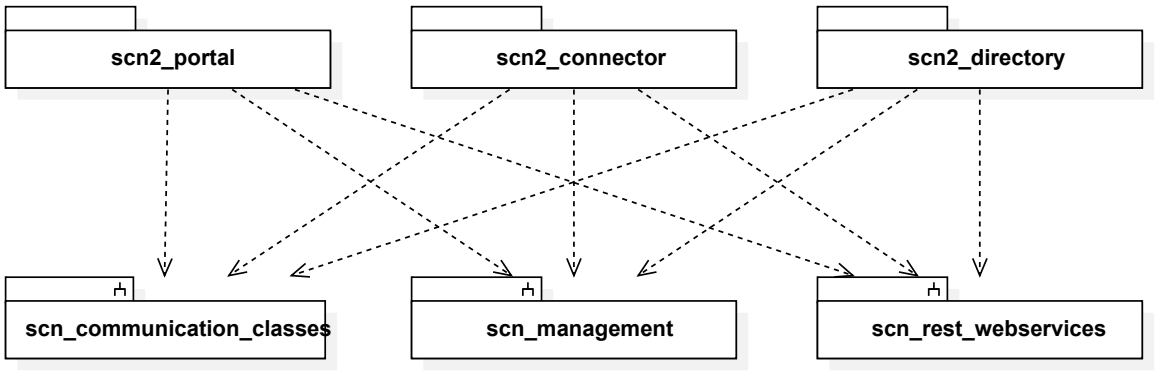


Abbildung 52: Paket-Struktur des Gesamtprojektes

Nachfolgend werden der Abstraktionsgrad der Beschreibung verringert und die jeweiligen Elemente des Prototypen in dessen interner Struktur näher erläutert. Für alle Komponenten existiert eine zentrale Controller-Klasse, welche alle Anfragen verarbeitet. Dieses Vorgehen basiert auf dem Konzept des Java Spring Frameworks. Die Ergebnisse werden anschließend durch Java Server Pages (JSP) verarbeitet und an den Aufrufer zurück gesendet.

Sämtliche Anfragen an die Portal-Komponente werden durch einen Controller gesteuert. Hierbei ist eine Trennung von Seitenaufrufen und Ajax-Aufrufen gegeben. Im Gegensatz zu Seitenaufrufen werden Ajax-Aufrufe währen des Darstellens der Webseite durch den Browser durchgeführt und deren Ergebnisse direkt durch JavaScript verarbeitet. Beispiele sind Zusatzinformationen von anderen Anwendern, wie der Nutzername oder das Profilbild. Für eine bessere Strukturierung wurden umfangreiche Methoden in eine Klasse mit dem Namen *PostManagement* ausgelagert. Abbildung 53 zeigt die Controller-Klasse in Verbindung mit der Klasse PostManagement und alle enthaltenen Methoden. Eine wichtige Funktion eines jeden Portals ist die Sitzungsverwaltung. Bei diesem Prototyp wird dies gesteuert durch das Java Spring Framework. Anwender erhalten nach erfolgreicher Anmeldung automatisch eine Session-ID in Form eines Cookies, das durch den Browser des Nutzers gespeichert wird. Die Abmeldung vom Portal sorgt für die Löschung desselbigen. Weiterhin werden Klassen verwendet, um dass Seitenmenü innerhalb der Anwendung zu erzeugen. Für die Darstellung von Informationen auf der Portalseite werden je nach Bedarf Ajax-Aufrufe durch die HTML-Seite durchgeführt und deren Ergebnis in die Portalseite in-

tegriert. Alle notwendigen Informationen werden, nachdem sie erfragt wurden, durch die Portal-Komponente bereitgestellt.

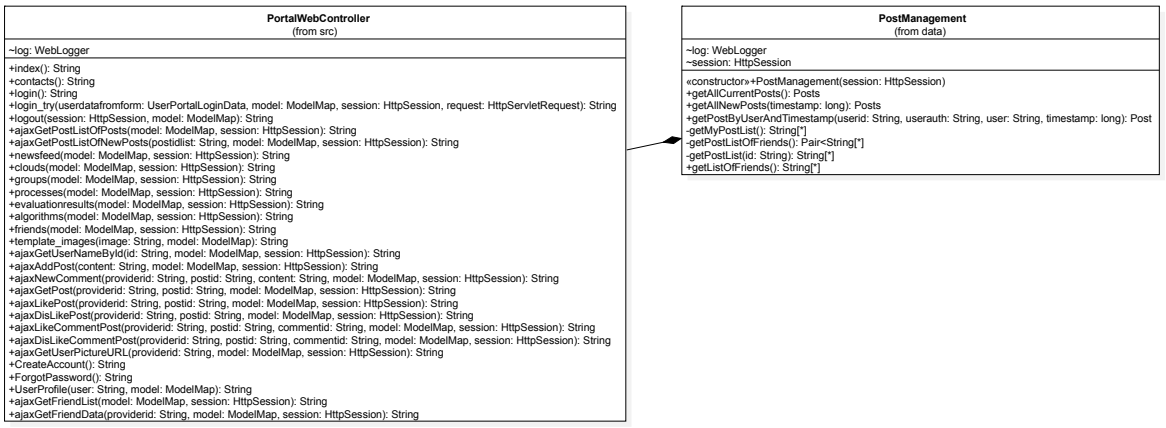


Abbildung 53: Klassen-basierte Grobstruktur der Portal-Komponente

Die Verbinder-Komponente stellt die direkte Kommunikation zu den Storage Clouds der Nutzer dar. Hierfür sind eine Reihe an Funktionen für den Zugriff auf die Informationen der Anwender implementiert. Eine Abstraktionsschicht für die unterschiedlichen Zugriffsstandards erlaubt eine einfache Entwicklung innerhalb der Komponente. Bei jedem Aufruf wird überprüft, ob der Empfänger die notwendigen Berechtigungen besitzt, die abgespeicherten Informationen zu sehen bzw. zu bearbeiten.

Damit ein umfänglicher Datenschutz gewährleistet werden kann, kommt zum einen ein Aggregationssystem und zum anderen ein Verifizierungssystem zum Einsatz. Das Aggregationssystem hat die Aufgabe, die verschiedenen Informationen aus verschiedenen Quellen zusammenzuführen. Das Verifizierungssystem stellt sicher, dass keine unterschiedlichen Informationen für einen Informationsbereich existieren. Beispielsweise könnte ein Kommentar zu einem späteren Zeitpunkt bearbeitet werden. Dieser verliert daraufhin seine Gültigkeit. Nur der Ersteller des Kommentars hat das Recht diesen anzupassen. Jeder Kommentar wird daher zusätzlich bei dem Ersteller angefragt und es wird überprüft, ob dieser übereinstimmt mit dem gespeicherten Kommentar des Postbesitzers. Abbildung 54 zeigt sowohl das Aggregations- als auch das Verifizierungssystem anhand eines eines Nutzerbeitrages innerhalb des Systems. Diese Herangehensweise lässt sich übertragen auf unterschiedlichste Anwendungsfälle. Primär erfolgt dadurch die Realisierung des Co-Datenschutzes, welcher mit zunehmender Kollaboration zwischen Unternehmen weiter an Bedeutung gewinnt.

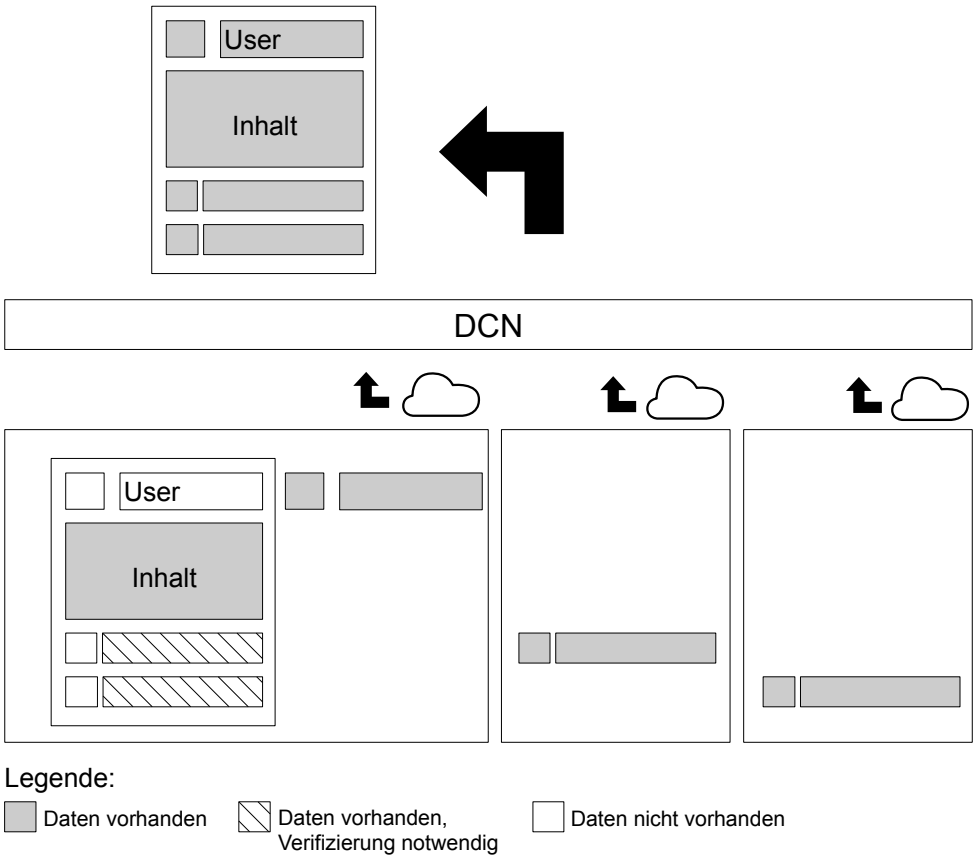


Abbildung 54: Aggregationssystem und Verifizierungssystem angewendet auf einen Beitrag schematisch dargestellt.

Für die Umsetzung der notwendigen Funktionalitäten sind eine Reihe an Methoden in der Klasse *ConnectorWebController* implementiert. (vgl. Abbildung 55) Je nach Anfrage als HTTP-Aufruf werden die Parameter übernommen und die notwendigen Daten aus den jeweiligen Storage Clouds geladen und als Ergebnis übertragen. Bei jedem Aufruf einer Methode erfolgt die Überprüfung der Rechte anhand des Rechtemanagements.

ConnectorWebController (from src)
~log: WebLogger
+index(): String +IsCloudAvailable(userid: String, userauthid: String, fromid: String, model: ModelMap): String +ListAllPosts(userid: String, userauthid: String, fromuserid: String, model: ModelMap): String +GetUserFriendList(userid: String, userauthid: String, fromuserid: String, model: ModelMap): String -getCloudData(id: String): CloudUserAccessData +GetPost(userid: String, userauthid: String, fromuserid: String, postid: String, model: ModelMap): String +AddPost(userid: String, userauthid: String, content: String, model: ModelMap): String +GetUserData(userid: String, userauthid: String, userproviderid: String, model: ModelMap): String +CommentPost(userid: String, userauthid: String, providerid: String, postid: String, content: String, model: ModelMap): String +LikePost(userid: String, userauthid: String, providerid: String, postid: String, model: ModelMap): String +DisLikePost(userid: String, userauthid: String, providerid: String, postid: String, model: ModelMap): String

+LikeCommentPost(userid: String, userauthid: String, providerid: String, postid: String, commentid: String, model: ModelMap): String
+DisLikeCommentPost(userid: String, userauthid: String, providerid: String, postid: String, commentid: String, model: ModelMap): String
+GetUserProfilePicture(userid: String, userauthid: String, providerid: String, model: ModelMap): String
+GetHallo(model: ModelMap): String
+GetUserFile(fileName: String, model: ModelMap, response: HttpServletResponse): void
+LikePost2(userid: String, userauthid: String, providerid: String, postid: String, model: ModelMap): String

Abbildung 55: Klassendiagramm ConnectorWebController

Die Nutzerverzeichnis-Komponente ist in ihrem Umfang deutlich geringer ausgeprägt, da sie nur einen geringen Funktionsumfang aufweist. Primär speichert sie die Anmeldedaten der Nutzer für die Storage Clouds. Weiterhin überprüft sie den Authentifizierungsschlüssel innerhalb des Netzwerkes für jeden Anwender. Abbildung 56 zeigt die Klasse *DirectioryWebController* sowie die dazugehörigen Methoden.

DirectoryWebController
(from src)

~log: WebLogger

+index(): String
+CheckUserIsAuth(id: String, userAuthId: String, model: ModelMap): String
+GetAuthKey(id: String, scn_password: String, model: ModelMap): String
+CheckUser(): String
+GetCloudData(id: String, model: ModelMap): String
+SetCloudData(): String

Abbildung 56: Klassendiagramm DirectoryWebController

Nachfolgend werden die drei Bibliotheken, welche die Hauptanwendungen unterstützen, näher beschrieben. Diese bestehen aus: scn_communication_classes, scn_management und scn_rest_webservices.

Die Bibliothek scn_communication_classes dient als Hilfsklasse für die Kommunikation der Komponenten. In ihr sind alle Klassen vereint, die für die Kommunikation bzw. den Datenaustausch benötigt werden. Innerhalb der Anwendungen werden, basierend auf den Klassen, Instanzen erzeugt und umgewandelt in das JSON-Format (Mapping). Anschließend werden die Daten per REST übertragen. Die empfangenen Daten werden anschließend zurück in Instanzen umwandelt. Für diesen Kommunikationsprozess ist es zielführend, die gleichen Klassen auf beiden Seiten der Kommunikation zu verwenden. Für jegliche Kommunikation zwischen den Elementen existieren spezielle Klassen. Abbildung 57 zeigt die Paketstruktur der Kommunikationsklassen-Bibliothek.

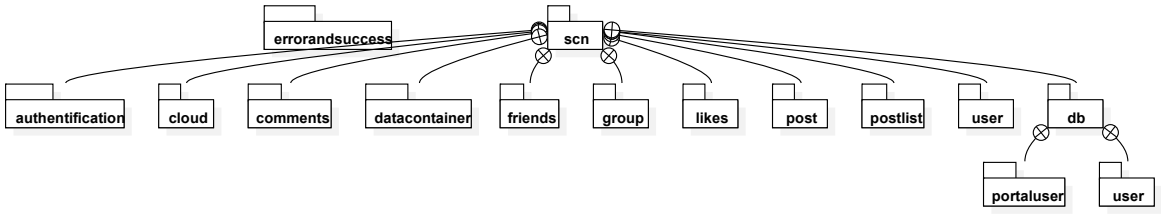


Abbildung 57: Paketstruktur der Kommunikationsklassen-Bibliothek

Die Bibliothek scn_management stellt Funktionen bereit, welche von allen Komponenten benötigt werden. Beispiele sind die angepasste und erweiterte Logging-Funktion und die Helper-Klassen. Dadurch wird unter anderem die Wiederverwendung von Quellcode gefördert. Abbildung 58 gibt eine Übersicht zu der Unterpaketstruktur von scn_management.

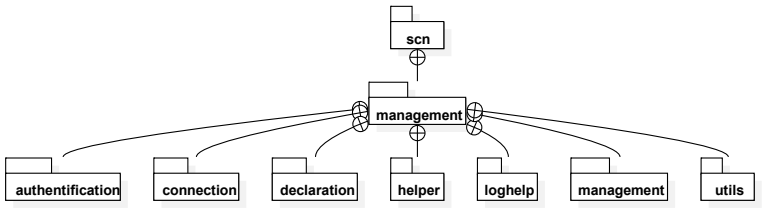


Abbildung 58: Paket-Struktur des Management

Die Bibliothek scn_rest_webservices ist verantwortlich für eine systemübergreifende Erzeugung von URI/URL-basierten Zugriffsschlüsseln auf die unterschiedlichen Webservices. Innerhalb dieser Anwendung wird mit Hilfe von Methodenverkettung auf die einzelnen Methoden zurückgegriffen, welche insgesamt einen String erzeugen. Diese repräsentieren eine eindeutige URL, welche direkt auf Funktionalitäten der Hauptkomponenten verweist. Diese Konzeption ermöglicht eine einfache und fehlerreduzierte Art und Weise der Verknüpfung von Komponenten. Abbildung 59 zeigt die Paketstruktur dieser Bibliothek.

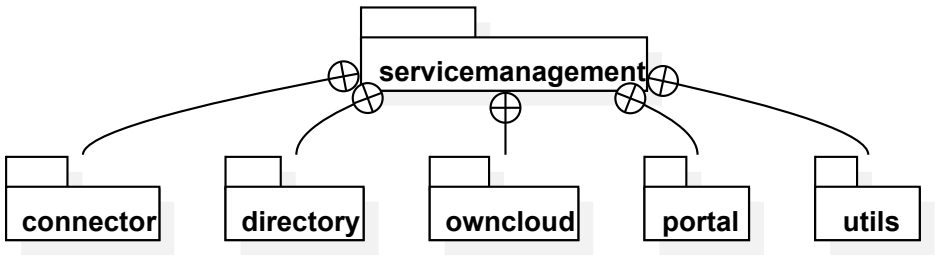


Abbildung 59: Paket-Struktur des Rest-Webservice-Zugriffsverwaltung

Nachdem der Software-strukturelle Aufbau des Prototypen beschrieben wurde, erfolgt im nächsten Abschnitt die Umsetzung und Anwendung, konkretisiert mit Anwendungsdarstellungen und Quellcodebeispielen.

10.2.5 Umsetzung und Anwendung

Die Umsetzung des Prototypen ist sehr umfangreich und die Dokumentation der gesamten Entwicklung würde den Rahmen dieser Dissertation sprengen. Aus diesem Grund sind einige Software-basierte Umsetzungen herausgenommen und werden im Folgenden kurz vorgestellt. Im Mittelpunkt stehen hierbei die Darstellung und die Zusammenführung eines Beitrages mit all seinen Elementen. Dies beinhaltet Kommentare und die Auflösung von Nutzernamen und Profilbildern. Abbildung 60 zeigt eine visuelle Darstellung des endgültigen Entwurfes eines Beitrages innerhalb des Portals. Es wird deutlich, dass zunächst der gesamte Beitrag bei einem Nutzer auf dessen Storage Cloud hinterlegt ist. Um Elemente, in diesem Fall Kommentare, anderer Nutzer zu integrieren, werden diese aus den jeweiligen Storage Clouds bezogen (Aggregation) und darauf geprüft, ob diese gleich sind mit den Informationen des Gesamtbeitrages (Verifizierung).

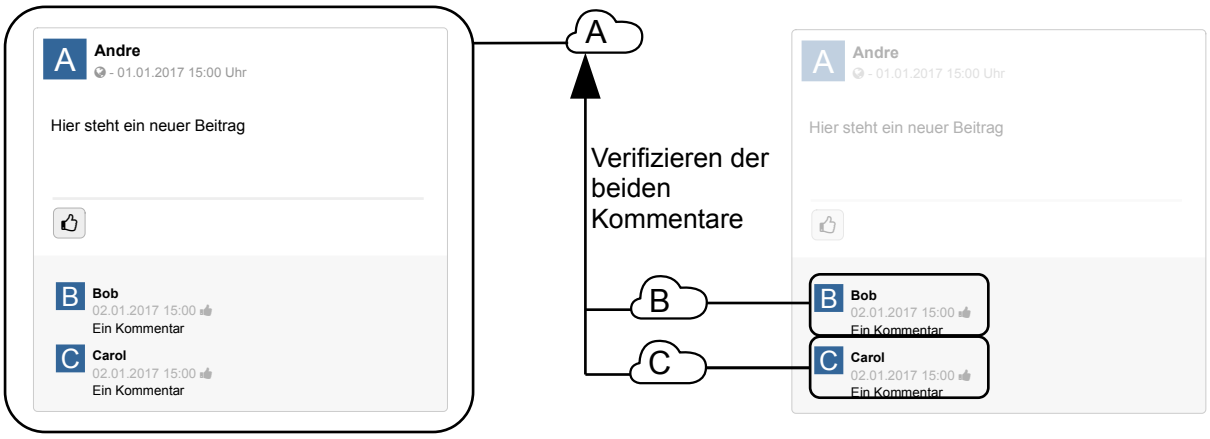


Abbildung 60: Verifizierung von Kommentaren eines Beitrages

Für die Realisierung der notwendigen Kommunikation zu den Komponenten und den jeweiligen Storage Clouds werden Daten durch eine REST-Kommunikation und den Einsatz von Mapping übertragen. Abbildung 61 zeigt eine beispielhafte Übertragung aller Freunde eines Nutzers innerhalb des Netzwerkes. Hierfür wird die Klasse *Friends* sowohl innerhalb des Verbinders als auch innerhalb des Portals integriert. Dies geschieht durch eine externe Bibliothek, welche beide Komponenten bei der Kompilierung integriert. Für den Austausch einer Instanz der Friends-Klasse wird diese zunächst beim Verbinder in ein JSON-Format umgewandelt. Anschließend erfolgt die Übertragung per REST. Die empfangenen Daten werden abschließend innerhalb des Portals wieder zu einer Instanz der Friends-Klasse umgewandelt und stehen für die weitere Nutzung bereit. Dadurch ist insgesamt eine verlust-

freie und fehlerreduzierte Möglichkeit gegeben, Daten innerhalb von Modulen zu übertragen.

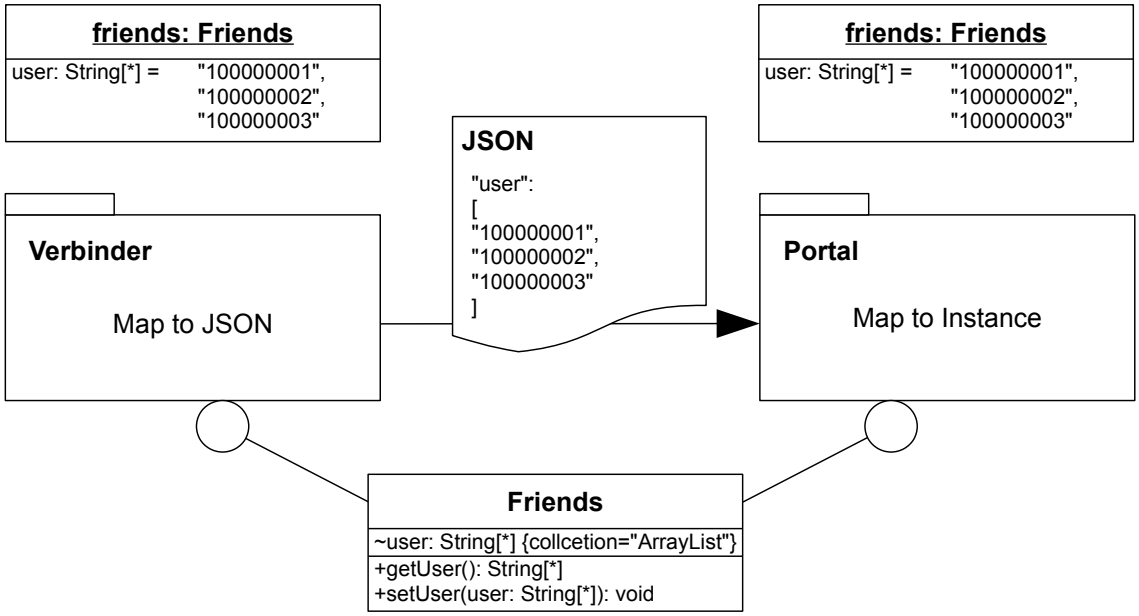


Abbildung 61: Beispiel Mapping einer Instanz der Friends-Klasse

Alle Daten, welche von den Nutzern erzeugt werden, sind in deren Storage Clouds gespeichert. Hierfür werden Informationen mit Hilfe von XML-Dateien hinterlegt. Listing 5 zeigt beispielhaft die Speicherung eines Beitrages.

```
1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <post>
3      <rights>
4          <level>2</level>
5          <include />
6          <exclude><user>10000004</user></exclude>
7      </rights>
8      <author>10000001</author>
9      <comments>
10         <comment>
11             <content>Ein Kommentar</content>
12             <time>02.01.2017 15:00:00</time>
13             <user>10000002</user>
14         </comment>
15         <comment>
16             <content>Ein Kommentar</content>
17             <time>02.01.2017 15:00:00</time>
18             <user>10000003</user>
```

```
        </comment>
    </comments>
    <content>Hier steht ein Beitrag</content>
    <id>1483279200</id>
    <likes/>
    <time>01.01.2017 15:00:00</time>
</post>
```

Listing 5: XML eines Beitrages

Für die REST-basierte Datenübertragung zwischen Modulen ist die Generierung einer URL notwendig. Damit eine einheitliche und standardisierte Kommunikation gewährleistet werden kann, ist eigens dafür eine Bibliothek entwickelt worden. Mit Hilfe der Verkettung von Methodenaufrufen ist die Erstellung eines Strings einfach und fehlerfrei möglich. Listing 6 zeigt, wie die Erstellung einer URL zum Aufruf der Funktionalität vom Anzeigen aller Freunde eines Nutzers beim Verbinder abläuft.

```
new RestWebServices().Connector().User().GetUserFriendList
(session.getAttribute("scn_id").toString(),
session.getAttribute("scn_auth_id").toString(),
session.getAttribute("scn_id").toString());
```

Listing 6: Konzipierung einer URL.

Darüber hinaus existiert eine umfangreiche Anzahl an möglichen Aufrufen, welche auszugsweise für den Verbinder in Abbildung 62 dargestellt sind. So entsteht eine hierarchische Struktur zur Generierung aller notwendigen REST-URLs.

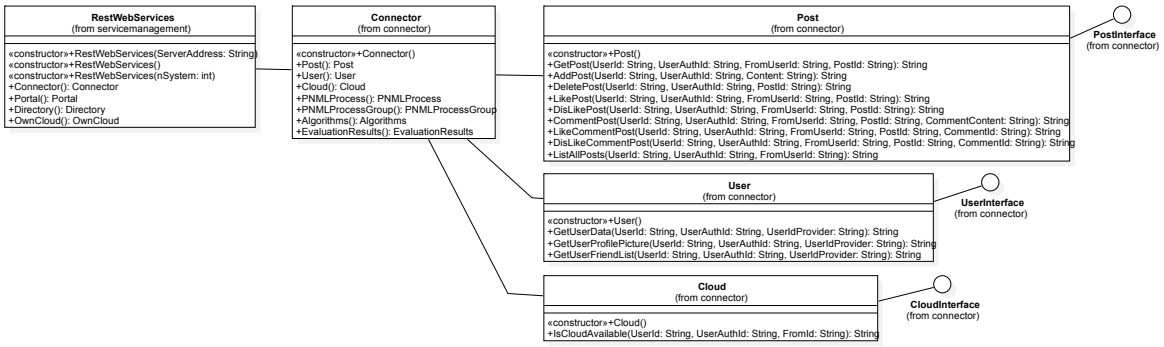


Abbildung 62: Klassenverbund des Verbinders für REST-URL-String-Erzeugung

Auf die gleiche Weise werden Zusatzinformationen während des Darstellens eines Beitrages mit Hilfe von Ajax-Aufrufen übermittelt. So ist insgesamt ein System entwickelt worden, welches grundsätzliche Funktionalitäten eines sozialen Netzwerkes basierend auf dem Konzept der DCN-Architektur bereitstellt. Abbildung 63 zeigt die Login- und Newsfeed-

Seite des Portals als Webinterface.

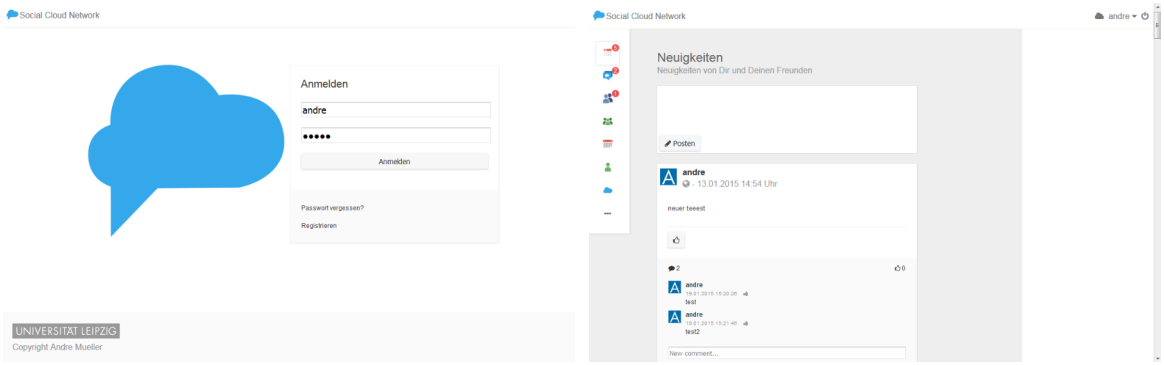


Abbildung 63: Webinterface Anmeldung und Newsfeed des Portals

Ein Anwender des Cloud Netzwerkes meldet sich über die Startseite des Portals an und erhält automatisiert alle notwendigen Zugriffsdaten, welche im Nutzerverzeichnis hinterlegt sind. Anschließend wird innerhalb des Netzwerkes nach den neusten Beiträgen aller Freunde gesucht und diese werden angezeigt. Im nächsten Abschnitt werden alle Anforderungen tabellarisch aufgelistet und deren Erfüllung detailliert beschrieben.

10.3 Überprüfen der Erfüllung der Anforderungen

In diesem Kapitel werden die 30 zuvor aufgestellten Anforderungen auf deren Erfüllung hin untersucht. Die Verwirklichung fokussiert sich hierbei auf die entworfene Architektur und deren Komponenten. Die aufgezeigten Anforderungen sind gegliedert in die Bereiche: Konzept, Datenschutz, Vertrauens- und Beziehungsmanagement und System. Für die Nachvollziehbarkeit ist die ID sowie die Beschreibung der jeweiligen Anforderung aufgeführt. Für jede Erfüllung erfolgt ein Verweis zu dem spezifischen Kapitel, sowie eine Erfüllungsbeschreibung.

Erfüllung der Konzeptanforderungen		
ID	Kap.	Erfüllung
CO1	5	**System als zentrale Instanz des Vertrauens („trusted party“)** Zentrales Informationssystem, welches das Management des Netzwerkes übernimmt.
CO2	5	**Am Markt angebotener Dienstleistungen verwenden, welche eine geringe Konfiguration für den Nutzer benötigen** Durch die Integration externer Dienste (Datenspeicherung) werden am Markt angebotene Dienstleistungen genutzt.
CO3	5	**Grundsätzliche Verwendung des Systems ohne Kosten für den Nutzer** Kosten für den Nutzer entstehen bei der Verwendung eines externen Dienstleisters für die Datenspeicherung. Dieser kann auch einen kostenlosen Anbieter verwenden.
CO4	5	**Alternative Auswahl an Datenschutzhöhe** Datenschutzhöhe abhängig von der Auswahl des Storage Cloud-Anbieters oder des eigenen Systems. Innerhalb des IS ist der Datenschutz gewahrt, da keine Daten gespeichert werden.
CO5	5	**Möglichkeit der Datenverschlüsselung auf den Speichermedien anbieten** Einige externe Storage Cloud-Dienstleister bieten Verschlüsselung an, somit besteht grundsätzlich die Möglichkeit Daten zu verschlüsseln.
CO6	8.1.2	**Beziehungsmanagement integrieren** Die Umsetzung des Rechtemanagement bezieht sich auf ein Beziehungsmanagement. Dies wird mit speziellen Regeln abgedeckt.
CO7	8.3	**Ladezeiten bis zum Anzeigen des Inhaltes minimieren auf unter 1 Minute** Durch ein überdachtes Aggregationssystem werden Ladezeiten eingehalten.
CO8	5	**Vollständige Verfügbarkeit der notwendigen Ressourcen in Form von Daten** Professionelle (kommerzielle) Storage Cloud-Dienstleister unterstützen eine sehr hohe Verfügbarkeit ihrer Systeme.
CO9	6.1.1	**Chat System Integration** Integration innerhalb des GUI in die Oberfläche in Form eines externen Dienstes.

Tabelle 33: Erfüllung der Konzeptanforderungen

Erfüllung der Datenschutzanforderungen		
ID	Kap.	Erfüllung
DS1	6.1.1	**Datenschutz muss für den Nutzer deutlich hervorgehoben werden** Durch eine grafische Darstellung innerhalb der Benutzerschnittstelle wird der Datenschutz deutlich hervorgehoben.
DS2	5	**Hohe Kontrolle über die Datenverwendung mit der Option der (automatischen) Löschung** Übernommen durch den externen Anbieter oder durch das System selbst.
DS3	8.1.2	**Einfaches Rechtemanagement, um Konflikte zu vermeiden** Das Rechtemanagement ist umgesetzt durch ein Beziehungsmanagement. Dies fördert die einfache Verwendung, einhergehend mit sehr wenigen Regeln, wodurch Konflikte vermieden werden.
DS4	7	**Anonymität der Nutzer innerhalb des Systems** Es werden keine Daten über die Nutzer innerhalb des Systems gespeichert, nur Daten für die interne Administration und für die individuellen Anpassung des Portals zur Komfortsteigerung.
DS5	5	**Der Schutz der persönlichen Daten sollte durch den Nutzer bzw. dessen Datenverwalter erfolgen** Die Storage Cloud-Lösung unterstützt diese Anforderung. Durch das System werden keine Daten der Nutzer gespeichert.
DS6	8.1.2	**Angeheftete Schutzrichtlinien** Zu allen erzeugten Daten wird eine Metadatei im XML-Format angeheftet, welche die Richtlinien enthält.
DS7	8.4	**Co-Datenschutz (Co-Privacy) bei gemeinsam erstellten Daten der Nutzer** Durch ein Verifizierungs- und Aggregationssystem aller Datenbestandteile wird Co-Datenschutz unterstützt.
DS8	6.1.1	**Für den Nutzer einfach nachvollziehbar dargestellter Datenschutz** Das GUI stellt den Datenschutz für die Nutzer deutlich dar.
DS9	6.1.2	**Empfehlung für Datenschutzeinstellung** Die Logikschicht des Portals berechnet automatisch Empfehlungen für den Nutzer, basierend auf Kommunikationsintensität. Bevorzugt Privacy by Default.

Tabelle 34: Erfüllung der Datenschutzanforderungen

Erfüllung der Vertrauens- und Beziehungsmanagementanforderungen		
ID	Kap.	Erfüllung
TR1	5	**Zentralisierung des Vertrauens** Durch das Grundkonzept realisiert.
TR2	6.1.1	**Es muss Vertrauen beim Nutzer erzeugt werden (z. B. mit Hilfe von anderen Nutzern)** Darstellung anderer Nutzer, die das IS erfolgreich einsetzten. Weiterhin der deutliche Hinweis auf die Erfüllung datenschutzrechtlicher Anforderungen.
RM1	6.1.2	**Unterstützung beim Kontaktmanagement** Das Portal bietet die Möglichkeit, Anfragen an Nutzer des Netzwerkes für Freundschaftsanfragen zu senden. Diese werden in den jeweiligen Storage Clouds gespeichert.
RM2	6.1.2	**Automatisch Beziehungen ableiten** Basierend auf der Interaktionsintensität werden Vorschläge unterbreitet.

Tabelle 35: Erfüllung der Vertrauens- und Beziehungsmanagementanforderungen

Erfüllung der Systemanforderungen		
ID	Kap.	System
SY1	8.1.3	**Quellenvielfalt an Daten durch hohe Abstraktion der Zugriffsschicht** Durch die Entwicklung einer Abstraktionsschicht ist es möglich, eine große Anzahl an externen Quellen für die Datenintegration anzusprechen.
SY2	8.2	**Konzept für die Verfügbarkeit und Nichtverfügbarkeit von Daten der Nutzer** Das Aggregationssystem setzt Datenbestände der Nutzer zusammen und reagiert auf eventuelle Nichtverfügbarkeit.
SY3	8.2	**Robustheit des Systems gegen Angriffe und fehlerhafte Daten** Fehlerhafte Daten und massenhafte und fehlerhafte Zugriffe werden abgefangen und unterbunden. (Container as a Service)
SY4	5.2	**Feingranular** Architektur ist in kleine Komponenten unterteilt. Diese sind wiederum unterteilt in Unterkomponenten.
SY5	8.2	**Interoperabilität** Die Datenspeicherung erfolgt im XML-Format und ist frei verwendbar für den Nutzer. Weiterhin werden eine Vielzahl an externen Diensten (Storage Clouds, NAS-Systeme) angesprochen.

SY6	8.1.2	**Rechte- und Interaktionsmanagement basierend auf Beziehungen** Das Rechtemanagement ist beziehungsbasiert. Die Interaktionen basieren auf den vorhandenen Rechten der Nutzer.
SY7	8.1.1	**Performante Suche im Netzwerk** Für die Nutzer die untereinander befreundet sind übernimmt dies der Verbinder. Eine externe Suche muss eingerichtet werden als externer Dienst.
SY8	6.1.2	**Selbstdarstellungsmanagement (Monitoring/ Feedback)** Zugriffsdaten können bei einem externen Anbieter bezogen werden. Innerhalb des Netzwerkes werden soziale Interaktionen integriert wie „Gefällt mir“.

Tabelle 36: Erfüllung der Systemanforderungen

Es konnte gezeigt werden, dass sämtliche Anforderungen durch das vorgestellte Konzept erfüllt worden sind.

10.4 Zusammenfassung

In diesem Kapitel wurde für die Durchführung eines Machbarkeitsnachweises des erstellten Konzeptes zunächst ein Anwendungsszenario definiert und festgelegt, anhand dessen eine Software-strukturelle Ausprägung durchgeführt werden konnte. Basierend darauf fand die Umsetzung eines Proof of Concept in Form einer prototypischen Implementierung statt. Abschließend konnte mit der Überprüfung der Erfüllung aller aufgestellten Anforderungen gezeigt werden, dass das vorgestellte Konzept als solches realisierbar ist.

11 Zusammenfassung und Ausblick

Das abschließende Kapitel Zusammenfassung und Ausblick summiert die wissenschaftliche Ausarbeitung dieser Publikation und zeigt einen forschungsorientierten Ausblick. Das Kapitel ist strukturiert in die Bereiche Resümee, Hauptbeitrag und zukünftige Forschung. Zunächst wird die Arbeit in einem Resümee nochmals chronologisch nachvollzogen. Im Abschnitt Hauptbeitrag wird anschließend der Schwerpunkt auf den Ablauf der wissenschaftlichen Bearbeitung sowie auf die gewonnenen Erkenntnisse gelegt. Abschließend zeigt die zukünftige Forschung den Bedarf für eine weitere Bearbeitung des Themengebietes auf.

11.1 Resümee

Diese Dissertation zeigte die Konzipierung eines dezentralen Informationssystems, welches alle Daten seiner Nutzer extern in Storage Clouds speichert und bei Bedarf integriert. Der Fokus lag hierbei, entgegen der Entwicklung einer reinen verteilten Anwendung, auf der Dezentralisierung des Systems. Es zeigte sich bei der Ausarbeitung, dass das ausschließliche Grundvertrauen in Technologie nicht ausreicht, um das gesellschaftliche Problem des Nichteinhaltens des Datenschutzes zu lösen. Für die wissenschaftliche Bearbeitung des Themas wurde eine Reihe an Thesen und Zielstellungen aufgestellt, welche nachfolgend kurz erwähnt werden.

1) Es ist mit den heutigen technischen Gegebenheiten möglich ein System zu entwickeln, welches Datenschutzaspekte deutlich besser adressiert als bisherige am Markt befindliche Angebote. 2) Dies schließt die Möglichkeit der Entwicklung eines Konzeptes ein, welches den Datenschutz deutlich erhöht. 3) Für einen nachhaltigen Lösungsansatz ist es notwendig, die technischen Möglichkeiten, soziale Interaktion und Kommunikation sowie den rechtlichen Rahmen zu verbinden. 4) Ziel muss es sein, den Fokus auf die Anwender des Systems zu legen.

Daraus ergibt sich der in dieser Dissertation gewählte Ablauf der wissenschaftlichen Publikation. Zunächst wurden die Aspekte der Dezentralisierung in Informationssystemen untersucht sowie verschiedene technische Realisierungen aufgezeigt. Anschließend erfolgte die umfangreiche Betrachtung von Datenschutz mit dem Schwerpunkt auf dem Datentransfer zwischen der Europäischen Union und den USA. Dies stellte die Vorbedingung für eine Anforderungsanalyse, welche, basierend auf einer systematischen Literaturanalyse sowie

einem Systemvergleich existierender Lösungsansätze, einen Anforderungskatalog für ein zukünftiges System beinhaltet. Dieser stellte die Grundlage für den Entwurf eines Architekturmodells. Basierend auf einem Grundkonzept wurden verschiedene Komponenten konstruiert: Portal, Verbinder und Nutzerverzeichnis. Mit Hilfe einer Erweiterung dieser Architektur konnten Herausforderungen in den Bereichen Performanz und Sicherheit adressiert werden. Die Evaluation des Modells fand durch einen Proof of Concept, ausgeprägt als prototypische Implementierung, statt. Das nachfolgende Kapitel Hauptbeitrag zeigt den in dieser Arbeit generierten Beitrag zur Wissensbasis auf.

11.2 Hauptbeitrag

Das Kapitel Hauptbeitrag beschreibt den in dieser Publikation erstellten Beitrag zur Wissensbasis und die relevanten Erkenntnisse für die Praxis nochmals detailliert. Die Adressaten für eine Verwertung der Ergebnisse sind die Wissenschaft, die Wirtschaft und die Gesellschaft bzw. Politik. Als explizite Ausprägung der Adressaten sind KMU und Privatpersonen, welche eine große wirtschaftlich relevante Gruppe an Beteiligten darstellen, zu nennen.

Im Mittelpunkt der wissenschaftlichen Bearbeitung der Forschungsfrage stand der Entwurf eines Konzeptes für die Stärkung des Datenschutzes. Dies ist in einem direkten Zusammenhang zum Einbezug des Nutzers in die Gestaltung des Gesamtsystems zu sehen. Hierbei sollte die Flexibilität für den Anwender möglichst hoch sein, um eine hohe Akzeptanz des Lösungsansatzes zu gewährleisten. Für die wissenschaftliche Ausarbeitung wurde eine Forschungsfrage mit vier Unterforschungsfragen gebildet, welche durch wissenschaftliche Methoden beantwortet werden konnten. Die übergeordnete Forschungsfrage beschäftigte sich mit der Gestaltung eines Informationssystems bzw. eines Cloud-Ökosystems.

Die zentrale Methode dieser Arbeit ist eine Konstruktion als Ausprägung einer Modellierung. Tabelle 37 zeigt eine detaillierte Übersicht der Forschungsfrage aus Sicht des Design Science. Die Forschungsfrage 1 beschäftigte sich mit den verschiedenen Arten der Dezentralisierung, sowohl auf technischer als auch auf organisatorischer Ebene. Die technisch geprägte Sicht untersuchte im Besonderen technische Möglichkeiten der Dezentralisierung, wie etwa verteilte Anwendungen. Organisatorische Aspekte der Dezentralisierung setzten den Fokus auf Institutionen, Vertrauen und Interaktionen der Beteiligten. Tabelle 38 zeigt eine Übersicht zur Forschungsfrage 1. Forschungsfrage 2 hatte zum Ziel, einen Anforderungskatalog zu entwerfen. Mit insgesamt 30 Anforderungen wurde ein umfangreicher Ka-

talog bereitgestellt für die anschließende Gestaltung und Modellierung. Die Anforderungen wurden auf ihre Qualität hin geprüft und dienen als Grundlage für die Entwicklung des Systems bzw. Konzeptes. (siehe Tabelle 39) Forschungsfrage 3 war die eigentliche Gestaltung des Konzeptes. Hierbei wurde die Methode Modellierung verwandt, um ein Architekturmodell zu modellieren. Tabelle 40 zeigt eine Übersicht zur UFF 3. Forschungsfrage 4 beschäftigte sich mit der Erweiterung der entworfenen Architektur. Durch den Einbezug neuer Technologien, wie etwa Microservices und Cluster-Infrastrukturen, konnte die Architektur in den Bereichen Sicherheit und Performanz verbessert werden (siehe Tabelle 41). Insgesamt wurden so vier Artefakte in dieser Arbeit entworfen: Modell der organisatorischen Dezentralisierung, Anforderungskatalog, Architekturmodell und Erweitertes Architekturmodell. Das Modell der organisatorischen Dezentralisierung hilft bei der Einordnung bestehender Konzepte und Ansätze und ermöglicht deren Kategorisierung und Abgrenzung. Weiterhin dient es der Unterstützung der Konzeptphase zukünftiger Anwendungen. Der Anforderungskatalog verdeutlicht die aktuellen Anforderungen von datensatzorientierten Informationssystemen. In diesem Zusammenhang zeigt es Anforderungen und Hinweise aus der Forschung auf. Das erstellte Architekturmodell zeigt einen möglichen Lösungsansatz für die Erfüllung aller aufgestellten Anforderungen und ist in diesem Sinne als Referenzarchitektur zu verstehen. Das erweiterte Architekturmodell bezieht aktuelle technische Möglichkeiten in die Gestaltung der Architektur mit ein und zeigt, wie diese Techniken in ein bestehendes Modell integriert werden können. Basierend auf diesen Artefakten und deren Evaluation thematisiert der folgende Abschnitt die zukünftige Forschung.

11.3 Zukünftige Forschung

Der Abschnitt zukünftige Forschung beleuchtet Potentiale einer weiteren wissenschaftlichen Auseinandersetzung mit dem Thema der vorliegenden Publikation. Das Modell der organisatorischen Dezentralisierung diente als Grundlage für die Konstruktion des DCN-Architekturmodells. Darauf aufbauend ist es zielführend, ein Modell zu entwerfen, welches als Entwicklungsgrundlage für neue Architekturmodelle dient und den Prozess der Konstruktion unterstützt. Des Weiteren ist die Optimierung und Anpassung des vorliegenden Anforderungskataloges bei neuen Erkenntnissen aus der Forschung und Praxis ein erfolgversprechender Ansatz. Schlussendlich ist die Übertragung des DCN-Modells in die Praxis, unter anderem durch Feldexperimente oder Expertenbefragungen, der nächste folgerichtige forschungsorientierte Schritt.

Literaturverzeichnis

[Abdelshkour 2015] Abdelshkour, M., IoT, from Cloud to Fog Computing, blogs@Cisco - Cisco Blogs, 2015 Abgerufen am 02.02.2016 von http://blogs.cisco.com/perspectives/iot-from-cloud-to-fog-computing.

[Andersson et al. 2013] Andersson, M., Hjalmarsson, A., Avital, M., Peer-to-Peer Service Sharing Platforms: Driving Share and Share Alike on a Mass-Scale, in: ICIS 2013 Proceedings, 2013.

[APPC 2016] APPC, appc/spec, GitHub, 2016 Abgerufen am 13.07.2016 von https://github.com/appc/spec.

[ARISTOTELES 1907] ARISTOTELES, Metaphysik. Ins deutsche übertragen von Adolf Lasson., Jena, Eugen Diederichs, 1907.

[Baden et al. 2009] Baden, R., Bender, A., Spring, N., et al., Persona: An Online Social Network with User-defined Privacy, in: Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication, ACM, New York, NY, USA (SIGCOMM '09) 2009, S. 135–146, DOI: 10.1145/1592568.1592585 — ISBN: 978-1-60558-594-9.

[Bar-Magen Numhauser et al. 2013] Bar-Magen Numhauser, J., Mesa, G. de, Antonio, J., XMPP Distributed Topology as a Potential Solution for Fog Computing, in: 2013, S. 26–32 — ISBN: 978-1-61208-299-8.

[Baun et al. 2009] Baun, C., Kunze, M., Nimis, J., et al., Cloud Computing: Web-basierte dynamische IT-Services, 1. Aufl. Springer Berlin Heidelberg 2009.

[BDSG 2015] BDSG,; Deutscher Bundestag (Hrsg.) Bundesdatenschutzgesetz, o.V. 2015.

[BFDI 2014] BFDI, Informationelle Selbstbestimmung – Datenschutz-Wiki, 2014 Abgerufen am 22.02.2016 von https://www.bfdi.bund.de/bfdi_wiki/index.php/Informationelle_Selbstbestimmung.

[BFDI 2010] BFDI, Privacy by Design, 2010.

[Bieber 2012] Bieber, C., Datenschutz als politisches Thema – von der Volkszählung zur Piratenpartei, in: Weichert, Jan-Hinrik; Schmidt, Thili (Hrsg.) Datenschutz Grundlagen, Entwicklungen und Kontroversen, Bundeszentrale für politische Bildung, Bonn 2012, S. 34–41 — ISBN: 978-3-8389-0190-9.

[BMBF 2014] BMBF, Die neue Hightech-Strategie: Innovationen für Deutschland, 2014.

[Bond 2015] Bond, J., The Enterprise Cloud: Best Practices for Transforming Legacy IT, 1 edition. O'Reilly Media, Sebastopol, CA 2015 — ISBN: 978-1-4919-0762-7.

[Bonomi et al. 2012] Bonomi, F., Milito, R., Zhu, J., et al., Fog Computing and Its Role in the Internet of Things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, ACM, New York, NY, USA (MCC '12) 2012, S. 13–16, DOI: 10.1145/2342509.2342513 — ISBN: 978-1-4503-1519-7.

[Brocke 2007] Brocke, J., Design Principles for Reference Modeling-Reusing Information Models by Means of Aggregation, Specialization, Instantiation, and Analogy, in: Reference Modeling for Business Systems Analysis, 2007, S. 47–75.

[vom Brocke/Buddendick 2006] vom Brocke, J., Buddendick, C., Reusable Conceptual Models – Requirements Based on the Design Science Research Paradigm, in: First International Conference on Design Science Research in Information Systems and Technology : February 24–25, 2006, Claremont, CA ; Proceedings, 2006, 2006.

[Buchegger et al. 2009] Buchegger, S., Schiöberg, D., Vu, L.-H., et al., PeerSoN: P2P Social Networking: Early Experiences and Insights, in: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems, ACM, New York, NY, USA (SNS '09) 2009, S. 46–52, DOI: 10.1145/1578002.1578010 — ISBN: 978-1-60558-463-8.

[Burghardt 2012] Burghardt, M., Projektmanagement: Leitfaden für die Planung, Überwachung und Steuerung von Projekten, 9. überarb. u. erw. Auflage. Publicis Publishing 2012 — ISBN: 978-3-89578-399-9.

[Chard et al. 2012] Chard, K., Bubendorfer, K., Caton, S., et al., Social Cloud Computing: A Vision for Socially Motivated Resource Sharing, in: IEEE Transactions on Services Computing, 5 (4) 2012, S. 551–563, DOI: 10.1109/TSC.2011.39.

[Chaturvedi et al. 2011] Chaturvedi, A.R., Dolk, D.R., Drnevich, P.L., Design Principles for Virtual Worlds, in: MIS Q., 35 (3) 2011, S. 673–684.

[Conway 1968] Conway, M.E.,; Thompson, F. D. (Hrsg.) How Do Committees Invent?, in: Datamation, 14 (5) 1968, S. 28–31.

[Cutillo et al. 2009] Cutillo, L.A., Molva, R., Strufe, T., Safebook: A privacy-preserving online social network leveraging on real-life trust, in:

IEEE Communications Magazine, 47 (12) 2009, S. 94–101, DOI: 10.1109/MCOM.2009.5350374.

[Czarnecki/Eisenecker 2000] Czarnecki, K., Eisenecker, U., Generative Programming: Methods, Tools, and Applications, 1 edition. Addison-Wesley Professional, Boston 2000 — ISBN: 978-0-201-30977-5.

[Däubler 2012] Däubler, W., Die kontrollierten Belegschaften, in: Weichert, Jan-Hinrik; Schmidt, Thili (Hrsg.) Datenschutz Grundlagen, Entwicklungen und Kontroversen, Bundeszentrale für politische Bildung, Bonn 2012, S. 188–197 — ISBN: 978-3-8389-0190-9.

[Davison et al. 2013] Davison, R.M., Ou, C.X.J., Martinsons, M.G., Information technology to support informal knowledge sharing, in: Information Systems Journal, 23 (1) 2013, S. 89–109, DOI: 10.1111/j.1365-2575.2012.00400.x.

[Denyer/Tranfield 2015] Denyer, D., Tranfield, D., Doing a literature review in business and management, 2015.

[Denyer/Tranfield 2009] Denyer, D., Tranfield, D., Producing a litertature review, in: SAGE Handbook of Organizational Research Methods, SAGE Publications Ltd, England 2009.

[DLA Piper 2015] DLA Piper, Global Data Protection Handbook, 2015 Abgerufen am 01.02.2016 von http://dlapiperdataprotection.com/#handbook/world-map-section/c1_US/c2_BR.

[Dostal et al. 2005] Dostal, W., Jeckle, M., Melzer, I., et al., Service-orientierte Architekturen mit Web Services: Konzepte - Standards - Praxis, Spektrum Akademischer Verlag 2005 — ISBN: 978-3-8274-1457-1.

[Duden 2016] Duden, Duden | Datenschutz | Rechtschreibung, Bedeutung, Definition, 2016 Abgerufen am 23.02.2016 von http://www.duden.de/rechtschreibung/Datenschutz.

[ECMA 2013] ECMA, Standard ECMA-404, 2013 Abgerufen am 14.04.2016 von http://www.ecma-international.org/publications/standards/Ecma-404.htm.

[EMRK 1950] EMRK, Europäischen Menschenrechtskonvention, o.V. 1950.

[O. A. 2005] Erklärung von Montreux: Ein universelles Recht auf den Schutz personenbezogener Daten und der Privatsphäre unter Beachtung der Vielfalt in einer globalisierten Welt 27. Internationale Datenschutzkonferenz in Montreux, Montreux 2005.

[Erway et al. 2015] Erway, C.C., Küpçü, A., Papamanthou, C., et al., Dynamic Provable Data Possession, in: ACM Trans. Inf. Syst. Secur., 17 (4) 2015, S. 15:1–15:29, DOI: 10.1145/2699909.

[EU-Kommission 2016] EU-Kommission, European Commission - PRESS RELEASES - Press release - Kommission und Vereinigte Staaten einigen sich auf neuen Rahmen für die transatlantische Datenübermittlung: den EU-US-Datenschutzschild, 2016 Abgerufen am 25.02.2016 von http://europa.eu/rapid/press-release_IP-16-216_de.htm.

[EU-Kommission 2012a] EU-Kommission, Verordnung des europäischen Parlaments und des Rates zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Datenverkehr (Datenschutz-Grundverordnung), o.V. 2012.

[EU-Kommission 2012b] EU-Kommission, Wie stärkt die Datenschutzreform die Rechte der Bürgerinnen und Bürger?, 2012.

[Fogues et al. 2015] Fogues, R., Such, J.M., Espinosa, A., et al., Open Challenges in Relationship-Based Privacy Mechanisms for Social Network Services, in: International Journal of Human-Computer Interaction, 31 (5) 2015, S. 350–370, DOI: 10.1080/10447318.2014.1001300.

[Fowler/Lewis 2014] Fowler, M., Lewis, J., Microservices, martinfowler.com, 2014 Abgerufen am 31.03.2016 von http://martinfowler.com/articles/microservices.html.

[Gambs/Lolive 2013] Gambs, S., Lolive, J., SlopPy: Slope One with Privacy, in: Pietro, Roberto Di; Herranz, Javier; Damiani, Ernesto; et al. (Hrsg.) Data Privacy Management and Autonomous Spontaneous Security, Springer Berlin Heidelberg (Lecture Notes in Computer Science) 2013, S. 104–117, DOI: 10.1007/978-3-642-35890-6_8 — ISBN: 978-3-642-35889-0.

[GG 2014] GG,; Deutscher Bundestag (Hrsg.) Grundgesetz für die Bundesrepublik Deutschland, o.V. 2014.

[Goldammer 2013] Goldammer, G., Informatik für Wirtschaft und Verwaltung: Einführung In Die Grundlagen, 1994. Aufl. Dr. Th. Gabler Verlag 2013 — ISBN: 978-3-409-13539-9.

[Goldstein 2002] Goldstein, E.B., Wahrnehmungspsychologie, 2. Aufl. Spektrum Akademischer Verlag 2002 — ISBN: 978-3-8274-1083-2.

[Goll 2014] Goll, J., Architektur- und Entwurfsmuster der Softwaretechnik: Mit lauffähigen Beispielen in Java, 2., aktualisierte Aufl. 2014. Springer Vieweg 2014 — ISBN: 978-3-658-05531-8.

[Google 2016] Google, Deutschland – Google-Suche – Traffic-Grafik – Google Transparenzbericht, 2016 Abgerufen am 17.02.2016 von https://www.google.com/transparencyreport/traffic/explorer/?r=DE&l=WEBSEARCH&csd=1453737528476&ced=1454950728476.

[Gull 2014] Gull, D.-I.D., Erfolgsfaktoren beim Einsatz virtueller Infrastrukturen im Unternehmen, in: HMD Praxis der Wirtschaftsinformatik, 47 (5) 2014, S. 16–24, DOI: 10.1007/BF03340508.

[Günterberg 2012] Günterberg, B., Unternehmensgrößenstatistik: Unternehmen, Umsatz und sozialversicherungspflichtig Beschäftigte 2004 bis 2009 in Deutschland, Ergebnisse des Unternehmensregisters (URS 95), 2. Aufl. Institut für Mittelstandsforschung Bonn, Bonn 2012 — ISBN: 2193-1895.

[Hammon/Hippner 2012] Hammon, D.-K.L., Hippner, P.D.H., Crowdsourcing, in: WIRTSCHAFTSINFORMATIK, 54 (3) 2012, S. 165–168, DOI: 10.1007/s11576-012-0321-7.

[Heckmann 2012] Heckmann, D., Grundprinzipien des Datenschutzrechts, in: Weichert, Jan-Hinrik; Schmidt, Thili (Hrsg.) Datenschutz Grundlagen, Entwicklungen und Kontroversen, Bundeszentrale für politische Bildung, Bonn 2012, S. 267–277 — ISBN: 978-3-8389-0190-9.

[Heinrich 2005] Heinrich, L.J., Informationsmanagement: Planung, Überwachung und Steuerung der Informationsinfrastruktur, vollständig überarbeitete und ergänzte Auflage. Oldenbourg Wissenschaftsverlag, München; Wien 2005 — ISBN: 978-3-486-57772-3.

[Heinrich et al. 2010] Heinrich, L.J., Heinzl, A., Riedl, R., Wirtschaftsinformatik: Einführung und Grundlegung, 4. Aufl. Springer 2010 — ISBN: 978-3-642-15425-6.

[Hevner 2007] Hevner, A., A Three Cycle View of Design Science Research, in: Scandinavian Journal of Information Systems, 19 (2) 2007.

[Hevner et al. 2004] Hevner, A.R., March, S.T., Park, J., et al., Design Science in Information Systems Research, in: MIS Q., 28 (1) 2004, S. 75–105.

[Hindel et al. 2006] Hindel, B., Hörmann, K., Müller, M., et al., Basiswissen Software-Projektmanagement: Aus- und Weiterbildung zum Certified Professional for Project Management nach iSQL-Standard, 2., überarb. u. erw. Aufl. dpunkt 2006 — ISBN:

978-3-89864-390-0.

[Hindman et al. 2011] Hindman, B., Konwinski, A., Zaharia, M., et al., Mesos: A Platform for Fine-grained Resource Sharing in the Data Center, in: Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, USENIX Association, Berkeley, CA, USA (NSDI'11) 2011, S. 295–308.

[Holoubek 2007] Holoubek, M.,; Potacs, Michael (Hrsg.) Handbuch des öffentlichen Wirtschaftsrechts: Band 1 / Band 2, 2., vollst. überarb. und erw. Aufl. Springer Vienna 2007 — ISBN: 978-3-211-36738-4.

[Hosanagar et al. 2010] Hosanagar, K., Han, P., Tan, Y., Diffusion Models for Peer-to-Peer (P2P) Media Distribution: On the Impact of Decentralized, Constrained Supply, in: Info. Sys. Research, 21 (2) 2010, S. 271–287, DOI: 10.1287/isre.1080.0221.

[Howe 2010] Howe, J., Crowdsourcing: Crowdsourcing: A Definition, Crowdsourcing: Crowdsourcing: A Definition, 2010 Abgerufen am 15.02.2016 von http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html.

[Howe 2008] Howe, J., Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business, 1. Aufl. Crown Business, New York 2008 — ISBN: 978-0-307-39620-4.

[IETF/Bray 2014] IETF, Bray, T., The JavaScript Object Notation (JSON) Data Interchange Format, 2014 Abgerufen am 14.04.2016 von https://tools.ietf.org/html/rfc7159.

[IfD et al. 2015] IfD, glh, Institut für Demoskopie Allensbach, et al., Cyber Security Report 2015: Ergebnisse einer repräsentativen Befragung von abgeordneten sowie Top-Führungskräften in mittleren und großen Unternehmen, 2015.

[Karla 2010] Karla, J., Can Web 2.0 Ever Forget?, in: Business & Information Systems Engineering, 2 (2) 2010, S. 105–107.

[Kii 2013] Kii, IaaS, PaaS, SaaS. What's that?, Kii Developer Community, 2013 Abgerufen am 08.02.2016 von http://community-.kii.com/t/iaas-paas-saas-whats-that/16.

[Krempl 2016] Krempl, S., Gutachten: Transatlantisches Rahmenabkommen zum Datenschutz ist rechtswidrig | heise online, 2016 Abgerufen am 25.02.2016 von http://www.heise.de/newsticker/meldung/Gutachten-Transatlantisches-Rahmenabkommen-zum-Datenschutz-ist-rechtswidrig-3108745.html.

[Kuster et al. 2011] Kuster, J., Huber, E., Lippmann, R., et al., Handbuch Projekt-

management, 3. Aufl. Springer, Berlin, Heidelberg 2011 — ISBN: 978-3-642-21242-0.

[Laborenz/Ertel 2014] Laborenz, K., Ertel, A., Responsive Webdesign: Anpassungsfähige Websites programmieren und gestalten, 2. Aufl. Galileo Computing 2014 — ISBN: 978-3-8362-3200-5.

[Laudon et al. 2009] Laudon, K.C., Laudon, J.P., Schoder, D., Wirtschaftsinformatik: Eine Einführung, 2. Aufl. Pearson Studium 2009 — ISBN: 978-3-8273-7348-9.

[Liu et al. 2015] Liu, D., Brass, D., Lu, Y., et al., Friendships in Online Peer-to-Peer Lending: Pipes, Prisms, and Relational Herding, in: Management Information Systems Quarterly, 39 (3) 2015, S. 729–742.

[Lüke 2012] Lüke, F., Datenschutz aus Verbrauchersicht, in: Weichert, Jan-Hinrik; Schmidt, Thili (Hrsg.) Datenschutz Grundlagen, Entwicklungen und Kontroversen, Bundeszentrale für politische Bildung, Bonn 2012, S. 154–162 — ISBN: 978-3-8389-0190-9.

[Mell/Grance 2011] Mell, P.M., Grance, T., SP 800-145. The NIST Definition of Cloud Computing, National Institute of Standards & Technology, Gaithersburg, MD, United States 2011.

[Melzer 2010] Melzer, I., Service-orientierte Architekturen mit Web Services: Konzepte - Standards - Praxis, 4. Aufl. 2010. Spektrum Akademischer Verlag 2010 — ISBN: 978-3-8274-2549-2.

[Moore 1997] Moore, J.F., The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems, Reprint. Harper Paperbacks, New York 1997 — ISBN: 978-0-88730-850-5.

[Müller/Ludwig 2016] Müller, A., Ludwig, A., Dezentrale Datenhaltung und Datenschutz in Cloud-Netzwerken, in: Wirtschaftsinformatik & Management, 8 (5) 2016, S. 20–27, DOI: 10.1007/s35764-016-0082-y.

[Müller et al. 2017] Müller, A., Ludwig, A., Franczyk, B., Data security in decentralized cloud systems – system comparison, requirements analysis and organizational levels, in: Journal of Cloud Computing, 6 2017, S. 15, DOI: 10.1186/s13677-017-0082-3.

[Müller et al. 2016] Müller, A., Ludwig, A., Franczyk, B., Einfluss neuer Technologien auf die Digitalisierung von urbanen Handelsräumen, in: stationär und online HANDEL IN DER STADT, IKMZ - Universitätsbibliothek, Cottbus 2016 — ISBN: 978-3-940471-26-0.

[Narayanan et al. 2012] Narayanan, A., Toubiana, V., Barocas, S., et al., A Critical Look at Decentralized Personal Data Architectures, in: arXiv:1202.4503 [cs], 2012.

[Newman 2015] Newman, S., Microservices: Konzeption und Design, 1., 2015. mitp 2015 — ISBN: 978-3-95845-081-3.

[Ni et al. 2010] Ni, Q., Bertino, E., Lobo, J., et al., Privacy-aware Role-based Access Control, in: ACM Trans. Inf. Syst. Secur., 13 (3) 2010, S. 24:1–24:31, DOI: 10.1145/1805974.1805980.

[Österle et al. 2010] Österle, H., Becker, J., Frank, U., et al., Memorandum zur gestaltungsorientierten Wirtschaftsinformatik, in: Zeitschrift für betriebswirtschaftliche Forschung, (62) 2010, S. 664–672.

[Pohl 2008] Pohl, K., Requirements Engineering: Grundlagen, Prinzipien,Techniken, 2., korrigierte Auflage. dpunkt.Verlag GmbH 2008 — ISBN: 978-3-89864-550-8.

[Rat für Forschung und Technologieentwicklung 2013] Rat für Forschung und Technologieentwicklung, Empfehlung zu einer optimierten Proof-of-Concept-Unterstützung im Wissenstransfer., 2013.

[Recker 2012] Recker, J., Scientific Research in Information Systems: A Beginner's Guide, 2013. Aufl. Springer, New York 2012 — ISBN: 978-3-642-30047-9.

[Repschläger et al. 2014] Repschläger, D.-I.J., Pannicke, D.-W.-I.D., Zarnekow, P.D.R., Cloud Computing: Definitionen, Geschäftsmodelle und Entwicklungspotenziale, in: HMD Praxis der Wirtschaftsinformatik, 47 (5) 2014, S. 6–15, DOI: 10.1007/BF03340507.

[Richardson/Ruby 2007] Richardson, L., Ruby, S., Web Services mit REST, 1. Aufl. O'Reilly Verlag GmbH & Co. KG 2007 — ISBN: 978-3-89721-727-0.

[Roßnagel 2012] Roßnagel, A., Modernisierung des Datenschutzrechts, in: Weichert, Jan-Hinrik; Schmidt, Thili (Hrsg.) Datenschutz Grundlagen, Entwicklungen und Kontroversen, Bundeszentrale für politische Bildung, Bonn 2012, S. 331–341 — ISBN: 978-3-8389-0190-9.

[Sadashiv/Kumar 2011] Sadashiv, N., Kumar, S.M.D., Cluster, grid and cloud computing: A detailed comparison, in: 2011 6th International Conference on Computer Science Education (ICCSE), 2011, S. 477–482, DOI: 10.1109/ICCSE.2011.6028683.

[Sarker et al. 2011] Sarker, S., Ahuja, M., Sarker, S., et al., The Role of Communication and Trust in Global Virtual Teams: A Social Network Perspective, in: J. Manage. Inf. Syst., 28 (1) 2011, S. 273–

310, DOI: 10.2753/MIS0742-1222280109.

[Sarodnick/Brau 2006] Sarodnick, F., Brau, H., Methoden der Usability Evaluation: Wissenschaftliche Grundlagen und praktische Anwendung, 1., Aufl. Verlag Hans Huber 2006 — ISBN: 978-3-456-84200-4.

[Schill 2012] Schill, A., Verteilte Systeme: Grundlagen und Basistechnologien, 2. Aufl. 2012. Springer 2012 — ISBN: 978-3-642-25795-7.

[Schmidt 2012] Schmidt, J.-H., Persönliche Öffentlichkeiten und informationelle Selbstbestimmung im Social Web, in: Weichert, Jan-Hinrik; Schmidt, Thili (Hrsg.) Datenschutz Grundlagen, Entwicklungen und Kontroversen, Bundeszentrale für politische Bildung, Bonn 2012, S. 215–223 — ISBN: 978-3-8389-0190-9.

[Seong et al. 2010] Seong, S.-W., Seo, J., Nasielski, M., et al., PrPl: A Decentralized Social Networking Infrastructure, in: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, ACM, New York, NY, USA (MCS '10) 2010, S. 8:1–8:8, DOI: 10.1145/1810931.1810939 — ISBN: 978-1-4503-0155-8.

[Shakimov et al. 2011] Shakimov, A., Lim, H., Caceres, R., et al., Vis-à-Vis: Privacy-preserving online social networking via Virtual Individual Servers, in: 2011 Third International Conference on Communication Systems and Networks (COMSNETS), 2011, S. 1–10, DOI: 10.1109/COMSNETS.2011.5716497.

[Shakimov et al. 2009] Shakimov, A., Varshavsky, A., Cox, L.P., et al., Privacy, Cost, and Availability Tradeoffs in Decentralized OSNs, in: Proceedings of the 2Nd ACM Workshop on Online Social Networks, ACM, New York, NY, USA (WOSN '09) 2009, S. 13–18, DOI: 10.1145/1592665.1592669 — ISBN: 978-1-60558-445-4.

[Shannon/Weaver 1949] Shannon, C.E., Weaver, W., The Mathematical Theory of Communication, University of Illinois Press 1949 — ISBN: 978-0-252-72546-3.

[Sharma/Datta 2012] Sharma, R., Datta, A., SuperNova: Super-peers based architecture for decentralized online social networks, in: 2012 Fourth International Conference on Communication Systems and Networks (COMSNETS), 2012, S. 1–10, DOI: 10.1109/COMSNETS.2012.6151349.

[Software Cluster 2015] Software Cluster, Emergente Software, 2015 Abgerufen am 24.11.2015 von http://www.software-cluster.org/de/for-

schung/themen/emergente-software.

[Software-Cluster 2015a] Software-Cluster, EMERGENT, 2015 Abgerufen am 03.02.2016 von http://www.software-cluster.com/de/forschung/projekte/verbundprojekte/emergent.

[Software-Cluster 2015b] Software-Cluster, Emergente Software, 2015 Abgerufen am 03.02.2016 von http://www.software-cluster.com/de/forschung/themen/emergente-software.

[Soltesz et al. 2007] Soltesz, S., Pötzl, H., Fiuczynski, M.E., et al., Container-based Operating System Virtualization: A Scalable, High-performance Alternative to Hypervisors, in: Proceedings of the 2Nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007, ACM, New York, NY, USA (EuroSys '07) 2007, S. 275–287, DOI: 10.1145/1272996.1273025 — ISBN: 978-1-59593-636-3.

[Statewatch 2016] Statewatch, Statewatch News Online: EU-US data deal incompatible with EU law and fundamental rights - European Parliament legal service, 2016 Abgerufen am 25.02.2016 von http://statewatch.org/news/2016/feb/ep-legal-opinion-umbrella.pdf.

[Statista 2012] Statista, Daten - Volumen der weltweit generierten Daten bis 2020 | Statistik, Statista, 2012 Abgerufen am 27.01.2016 von http://de.statista.com/statistik/daten/studie/267974/umfrage/prognose-zum-weltweit-generierten-datenvolumen/.

[Statista 2015] Statista, Prognose privater Internet-Traffic - nach Segment bis 2019 | Statistik, Statista, 2015 Abgerufen am 27.01.2016 von http://de.statista.com/statistik/daten/studie/152551/umfrage/prognose-zum-internet-traffic-nach-segment/.

[TecChannel 2014] TecChannel, Cloud Computing - der deutsche Mittelstand hinkt hinterher - TecChannel-Studie | TecChannel.de, Cloud Computing - der deutsche Mittelstand hinkt hinterher - TecChannel-Studie | TecChannel.de, 2014 Abgerufen am 29.01.2016 von http://www.tecchannel.de/wege_in_die_cloud/2053924/cloud_studie_tc_2014_tx/index.html.

[TeleTrusT 2015] TeleTrusT, Bundesverband IT-Sicherheit warnt vor Absenkung des IT-Sicherheitsniveaus durch TTIP - TeleTrusT – Bundesverband IT-Sicherheit e.V. - Pioneers in IT security. - www.teletrust.de, 2015 Abgerufen am 29.02.2016 von https://www.teletrust.de/startseite/pressemeldung/?tx_ttnews%5Btt_news%5D=804&.

[Tigelaar et al. 2012] Tigelaar, A.S., Hiemstra, D., Trieschnigg, D., Peer-to-Peer In-

formation Retrieval: An Overview, in: ACM Trans. Inf. Syst., 30 (2) 2012, S. 9:1–9:34, DOI: 10.1145/2180868.2180871.

[Tilkov et al. 2015] Tilkov, S., Eigenbrodt, M., Schreier, S., et al., REST und HTTP: Entwicklung und Integration nach dem Architekturstil des Web, 3., u. erw. Aufl. dpunkt.verlag GmbH 2015 — ISBN: 978-3-86490-120-1.

[TKG 2016] TKG,; Deutscher Bundestag (Hrsg.) Telekommunikationsgesetz, o.V. 2016.

[Trepte 2012] Trepte, S., Privatsphäre aus psychologischer Sicht, in: Weichert, Jan-Hinrik; Schmidt, Thili (Hrsg.) Datenschutz Grundlagen, Entwicklungen und Kontroversen, Bundeszentrale für politische Bildung, Bonn 2012, S. 59–65 — ISBN: 978-3-8389-0190-9.

[Tsai et al. 2011] Tsai, J.Y., Egelman, S., Cranor, L., et al., The Effect of Online Privacy Information on Purchasing Behavior: An Experimental Study, in: Info. Sys. Research, 22 (2) 2011, S. 254–268, DOI: 10.1287/isre.1090.0260.

[Vaquero/Rodero-Merino 2014] Vaquero, L.M., Rodero-Merino, L., Finding Your Way in the Fog: Towards a Comprehensive Definition of Fog Computing, in: SIGCOMM Comput. Commun. Rev., 44 (5) 2014, S. 27–32, DOI: 10.1145/2677046.2677052.

[VHB 2015] VHB, Teilrating WI: Verband der Hochschullehrer für Betriebswirtschaft e.V., 2015 Abgerufen am 19.11.2015 von http://vhbonline.org/service/jourqual/vhb-jourqual-3/teilrating-wi/.

[Vossen et al. 2012] Vossen, G., Haselmann, T., Hoeren, T., Cloud-Computing für Unternehmen: Technische, wirtschaftliche, rechtliche und organisatorische Aspekte, 1., Auflage. dpunkt.verlag GmbH 2012 — ISBN: 978-3-89864-808-0.

[W3C 2008] W3C, Extensible Markup Language (XML) 1.0 (Fifth Edition), 2008 Abgerufen am 15.04.2016 von https://www.w3.org/TR/2008/REC-xml-20081126/.

[W3C 2012] W3C, Media Queries, Media Queries: W3C Recommendation 19 June 2012, 2012 Abgerufen am 01.04.2016 von https://www.w3.org/TR/css3-mediaqueries/.

[W3C 2004] W3C, Web Services Glossary, Web Services Glossary, 2004 Abgerufen am 31.03.2016 von https://www.w3.org/TR/ws-gloss/#webservice.

[Weichert/Schmidt 2012] Datenschutz Grundlagen, Entwicklungen und Kontroversen

Bundeszentrale für politische Bildung, Bonn 2012 — ISBN: 978-3-8389-0190-9.

[Weichert 2012] Weichert, T., Codex Digitalis Universalis, in: Weichert, Jan-Hinrik; Schmidt, Thili (Hrsg.) Datenschutz Grundlagen, Entwicklungen und Kontroversen, Bundeszentrale für politische Bildung, Bonn 2012, S. 345–348 — ISBN: 978-3-8389-0190-9.

[Welt 2014] Welt, Gefeit vor der NSA? Deutsche Cloud-Anbieter werben mit Sicherheit, Welt Online, 18.3.2014 2014.

[Williams/Sion 2013] Williams, P., Sion, R., Access Privacy and Correctness on Untrusted Storage, in: ACM Trans. Inf. Syst. Secur., 16 (3) 2013, S. 12:1–12:29, DOI: 10.1145/2535524.

[Wolff 2015] Wolff, E., Microservices: Grundlagen flexibler Softwarearchitekturen, 1., Auflage. dpunkt.verlag GmbH 2015 — ISBN: 978-3-86490-313-7.

[Yeung et al. 2009] Yeung, C.A., Liccardi, I., Lu, K., et al., Decentralization: The future of online social networking, in: In W3C Workshop on the Future of Social Networking Position Papers, 2009.

[ZeitOnline 2015] ZeitOnline, TTIP: TTIP bedroht Datenschutz in Europa, Die Zeit, Hamburg 2015 2015.

[Zhang/Mislove 2013] Zhang, L., Mislove, A., Building Confederated Web-based Services with Priv.Io, in: Proceedings of the First ACM Conference on Online Social Networks, ACM, New York, NY, USA (COSN '13) 2013, S. 189–200, DOI: 10.1145/2512938.2512943 — ISBN: 978-1-4503-2084-9.

Anhang

A Anwendung der Qualitätskriterien

CO1 Zentrale Instanz als „trusted party“		
ID		Erfüllung
RE1	●	*Erfüllt.*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	◐	*Was oder wer ist die zentrale Instanz?*
RE5	●	*Erfüllt.*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach		System als zentrale Instanz des Vertrauens („trusted party“)

CO2 Einsatz bestehender DL ohne aufwändige Konfiguration		
ID		Erfüllung
RE1	●	*Erfüllt.*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	◐	*Welche DL? / Welcher Aufwand?*
RE5	◐	*Welche Dienstleistungen sind gemeint?*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach		Am Markt angebotene DL verwenden, welche eine geringe Konfiguration für den Nutzer benötigen

CO3 System ohne Kosten/variable Kosten

ID	Erfüllung
RE1	● *Erfüllt.*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	● *Erfüllt.*
RE5	● *Erfüllt.*
RE6	◐ *Entweder ohne Kosten oder mit Kosten. Was heißt variabel?*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*

Nach Grundsätzliche Verwendung des Systems ohne Kosten für den Nutzer

CO4 Alternative Auswahl an Datenschutzhöhe

ID	Erfüllung
RE1	● *Erfüllt.*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	● *Erfüllt.*
RE5	● *Erfüllt.*
RE6	● *Erfüllt.*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*

Nach Alternative Auswahl an Datenschutzhöhe

CO5 Möglichkeit der Verschlüsselung anbieten

ID	Erfüllung
RE1	◐ *Welche Verschlüsselung und wo?*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	◐ *Welche Verschlüsselung?*
RE5	● *Erfüllt.*
RE6	● *Erfüllt.*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*

Nach Möglichkeit der Datenverschlüsselung auf den Speichermedien anbieten

CO6 Beziehungsmanagement integrieren		
ID		Erfüllung
RE1	●	*Erfüllt.*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	●	*Erfüllt.*
RE5	●	*Erfüllt.*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach	Beziehungsmanagement integrieren	

CO7 Ladezeiten minimieren bis zu einem Punkt x		
ID		Erfüllung
RE1	○	*Punkt x ist nicht definiert!*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	◐	*Welche Ladezeiten?*
RE5	●	*Erfüllt.*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach	Ladezeiten bis zum Anzeigen des Inhaltes minimieren auf unter 1 Minute	

CO8 Vollständige Verfügbarkeit der Ressourcen		
ID		Erfüllung
RE1	●	*Erfüllt.*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	◐	*Welche Ressourcen?*
RE5	◐	*Wieso müssen Ressourcen vollständig verfügbar sein?*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach	Vollständige Verfügbarkeit der notwendigen Ressourcen in Form von Daten	

CO9 Chat System Integration	
ID	Erfüllung
RE1	● *Erfüllt.*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	● *Erfüllt.*
RE5	● *Erfüllt.*
RE6	● *Erfüllt.*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*
Nach	Chat System Integration

DS1 Datenschutz muss für den Nutzer deutlich hervorgehoben werden	
ID	Erfüllung
RE1	● *Erfüllt.*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	● *Erfüllt.*
RE5	● *Erfüllt.*
RE6	● *Erfüllt.*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*
Nach	Datenschutz muss für den Nutzer deutlich hervorgehoben werden

DS2 Hohe Kontrolle über die Daten mit der Option der (automatischen) Löschung	
ID	Erfüllung
RE1	● *Erfüllt.*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	◐ *Wie ist die Kontrolle ausgestaltet?*
RE5	● *Erfüllt.*
RE6	● *Erfüllt.*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*
Nach	Hohe Kontrolle über die Datenverwendung mit der Option der (automatischen) Löschung

DS3 Einfaches Rechtemanagement, um Konflikte zu vermeiden	
ID	Erfüllung
RE1	● *Erfüllt.*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	● *Erfüllt.*
RE5	● *Erfüllt.*
RE6	● *Erfüllt.*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*
Nach	Einfaches Rechtemanagement, um Konflikte zu vermeiden

DS4 Anonymität der Nutzer	
ID	Erfüllung
RE1	● *Erfüllt.*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	● *Erfüllt.*
RE5	○ *Welche Anonymität?*
RE6	● *Erfüllt.*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*
Nach	Anonymität der Nutzer innerhalb des Systems

DS5 Der Schutz der persönlichen Daten sollte durch den Nutzer bzw. dessen Datenverwalter erfolgen	
ID	Erfüllung
RE1	● *Erfüllt.*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	● *Erfüllt.*
RE5	● *Erfüllt.*
RE6	● *Erfüllt.*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*
Nach	Der Schutz der persönlichen Daten sollte durch den Nutzer bzw. dessen Datenverwalter erfolgen

DS6 Angeheftete Schutzrichtlinien		
ID		Erfüllung
RE1	●	*Erfüllt.*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	●	*Erfüllt.*
RE5	●	*Erfüllt.*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach Angeheftete Schutzrichtlinien		

DS7 Co-Datenschutz (Co-privacy)		
ID		Erfüllung
RE1	○	*In welchem Zusammenhang?*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	◐	*Bei welchen Interaktionen?*
RE5	●	*Erfüllt.*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach Co-Datenschutz (Co-privacy) bei gemeinsam erstellten Daten der Nutzer		

DS8 Verständlichkeit des Datenschutzes		
ID		Erfüllung
RE1	◐	*Für wen und in welcher Weise?*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	◐	*Für wen und in welcher Weise?*
RE5	●	*Erfüllt.*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach Für den Nutzer einfach nachvollziehbar dargestellter Datenschutz		

DS9 Empfehlung für Datenschutzeinstellung	
ID	Erfüllung
RE1	● *Erfüllt.*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	● *Erfüllt.*
RE5	● *Erfüllt.*
RE6	● *Erfüllt.*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*
Nach	Empfehlung für Datenschutzeinstellung

TR1 Zentralisierung des Vertrauens	
ID	Erfüllung
RE1	● *Erfüllt.*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	● *Erfüllt.*
RE5	● *Erfüllt.*
RE6	● *Erfüllt.*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*
Nach	Zentralisierung des Vertrauens

TR2 Es muss Vertrauen beim Nutzer erzeugt werden (z. B. mit Hilfe von anderen Nutzern)	
ID	Erfüllung
RE1	● *Erfüllt.*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	● *Erfüllt.*
RE5	● *Erfüllt.*
RE6	● *Erfüllt.*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*
Nach	Es muss Vertrauen beim Nutzer erzeugt werden (z. B. mit Hilfe von anderen Nutzern)

RM1 Unterstützung beim Kontaktmanagement		
ID		Erfüllung
RE1	●	*Erfüllt.*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	●	*Erfüllt.*
RE5	●	*Erfüllt.*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach Unterstützung beim Kontaktmanagement		

RM2 Automatisch Beziehungen ableiten		
ID		Erfüllung
RE1	●	*Erfüllt.*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	●	*Erfüllt.*
RE5	●	*Erfüllt.*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach Automatisch Beziehungen ableiten		

SY1 Quellenvielfalt		
ID		Erfüllung
RE1	◐	*Welche Quellen?*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	◐	*Welche Quellen?*
RE5	●	*Erfüllt.*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach Quellenvielfalt an Daten durch hohe Abstraktion der Zugriffsschicht		

SY2 Verfügbarkeit von Daten		
ID		Erfüllung
RE1	◐	*Was geschieht bei Nichtverfügbarkeit?*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	◐	*Welche Daten?*
RE5	●	*Erfüllt.*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach Konzept für die Verfügbarkeit und Nichtverfügbarkeit von Daten der Nutzer		

SY3 Robustheit des Systems		
ID		Erfüllung
RE1	◐	*Wogegen?*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	●	*Erfüllt.*
RE5	●	*Erfüllt.*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach Robustheit des Systems gegen Angriffe und fehlerhafte Daten		

SY4 Feingranular		
ID		Erfüllung
RE1	●	*Erfüllt.*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	●	*Erfüllt.*
RE5	●	*Erfüllt.*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach Feingranular		

SY5 Interoperabilität	
ID	Erfüllung
RE1	● *Erfüllt.*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	● *Erfüllt.*
RE5	● *Erfüllt.*
RE6	● *Erfüllt.*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*
Nach	Interoperabilität

SY6 Auf Beziehungen basierend	
ID	Erfüllung
RE1	● *Erfüllt.*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	◐ *Was soll auf Beziehungen basieren?*
RE5	● *Erfüllt.*
RE6	● *Erfüllt.*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*
Nach	Rechte- und Interaktionsmanagement basierend auf Beziehungen

SY7 Performante Suche im Netzwerk	
ID	Erfüllung
RE1	● *Erfüllt.*
RE2	● *Erfüllt.*
RE3	● *Erfüllt.*
RE4	● *Erfüllt.*
RE5	● *Erfüllt.*
RE6	● *Erfüllt.*
RE7	● *Erfüllt.*
RE8	● *Erfüllt.*
RE9	● *Erfüllt.*
RE10	● *Erfüllt.*
Nach	Performante Suche im Netzwerk

SY8 Selbstdarstellungsmanagement (Monitoring/Feedback)		
ID		Erfüllung
RE1	●	*Erfüllt.*
RE2	●	*Erfüllt.*
RE3	●	*Erfüllt.*
RE4	●	*Erfüllt.*
RE5	●	*Erfüllt.*
RE6	●	*Erfüllt.*
RE7	●	*Erfüllt.*
RE8	●	*Erfüllt.*
RE9	●	*Erfüllt.*
RE10	●	*Erfüllt.*
Nach Selbstdarstellungsmanagement (Monitoring/Feedback)		

B Forschungsrahmenwerk für Forschungsfragen

B.1 Forschungsfrage

Forschungsfrage (FF)
Frage Wie muss ein Informationssystem (IS) gestaltet sein, welches den Schutz und die Kontrolle von Datenbeständen in zentralen Anwendungssystemen mit dezentraler Datenhaltung ermöglicht und welches in seiner Komplexität beherrschbar bleibt?
Methodiken • Systematische Literaturanalyse • Systemvergleich • Konstruktion
Artefakte • Modell der organisatorischen Dezentralisierung • Anforderungskatalog • Architekturmodell (DCN) • Architekturmodelle (DCN-Erweiterung)
Ablauf ► Analyse Eine Literaturanalyse und ein Systemvergleich erlauben das Aufstellen eines Anforderungskataloges. Dieser wird beeinflusst von Möglichkeiten der org. Dezentralisierung. ► Entwurf Ein Informationssystem wird konzipiert (Proof of Principal), welches die Anforderungen erfüllt und Datenschutzaspekte berücksichtigt. ► Evaluation Das konzipierte Informationssystem wird auf seine Realisierbarkeit durch einen Prototypen (Proof of Concept) evaluiert.
Diffusion [Müller/Ludwig 2016], [Müller et al. 2016], [Müller et al. 2017]

Tabelle 37: Übersicht Forschungsfrage

B.2 Unterforschungsfrage 1

Unterforschungsfrage 1 (UFF1)
Frage Welche Arten der Dezentralisierung gibt es in einem IS auf organisatorischer und technischer Ebene?
Methodiken • Systematische Literaturanalyse • Systemvergleich • Modellierung
Artefakte • Modell der organisatorischen Dezentralisierung
Ablauf ► Analyse Mit Hilfe einer systematischen Literaturanalyse und eines Systemvergleichs werden Erkenntnisse gewonnen, die es ermöglichen, ein Modell der organisatorischen Dezentralisierung zu modellieren. ► Entwurf Es werden fünf Stufen der organisatorischen Dezentralisierung modelliert. ► Evaluation Bestehende Systeme werden in die fünf Stufen eingeordnet.
Diffusion [Müller et al. 2017]

Tabelle 38: Übersicht Forschungsfrage 1

B.3 Unterforschungsfrage 2

Unterforschungsfrage 2 (UFF2)
Frage Welche Anforderungen ergeben sich in einem zentralen Anwendungssystem mit dezentraler Datenhaltung im Bereich Datenschutz und -kontrolle?
Methodiken • Systematische Literaturanalyse • Systemvergleich • Anforderungsermittlung aus Literatur und Praxis
Artefakte • Anforderungskatalog
Ablauf ► Analyse Mit Hilfe einer systematischen Literaturanalyse und eines Systemvergleichs werden Erkenntnisse gewonnen, die es ermöglichen Anforderungen für ein Informationssystem zu erstellen. ► Entwurf Aufstellen eines Anforderungskataloges mit 30 Anforderungen in den Bereichen: Konzept, Datenschutz, Vertrauen, Beziehungsmanagement und System. ► Evaluation Gewonnene Anforderungen aus Literatur und Systemvergleich werden nach zehn Qualitätskriterien evaluiert. Weiterhin wird der Anforderungskatalog bei dem Konzept/Architekturmodell sowie bei dem Proof of Concept evaluiert.
Diffusion [Müller et al. 2017]

Tabelle 39: Übersicht Forschungsfrage 2

B.4 Unterforschungsfrage 3

Unterforschungsfrage 3 (UFF3)
Frage Wie können bei der Gestaltung eines Informationssystems die Anforderungen hinsichtlich Datenschutz und Komplexität in einem dezentralen Anwendungssystem berücksichtigt werden?
Methodiken • Modellierung
Artefakte • Architekturmodell
Ablauf ► Analyse Die aufgestellten Anforderungen bilden die Analysephase. ► Entwurf Mit Hilfe der Anforderungen wird ein Architekturmodell modelliert. ► Evaluation Ein Prototyp dient der Evaluation des Architekturmodells.
Diffusion [Müller/Ludwig 2016]

Tabelle 40: Übersicht Forschungsfrage 3

B.5 Unterforschungsfrage 4

Unterforschungsfrage 4 (UFF4)
Frage Welche Softwaretechniken eignen sich, den Anforderungen in den Bereichen Sicherheit und Performanz gerecht zu werden?
Methodiken • Modellierung
Artefakte • Architekturmodelle (Erweiterung der DCN-Architektur) ◦ Storage Cloud-Zugriffs-Architektur ◦ GUI-Architektur ◦ Nutzerverzeichnis-Architektur
Ablauf ► Analyse Literatur zur Container-basierten Virtualisierung zeigt die aktuellen Trends und Technologien moderner Softwaresysteme auf. Das Microservice-Paradigma dient als Grundlage des Entwurfes. ► Entwurf Das bestehende Architekturmodell wird erweitert um aktuelle Softwaretechniken. ► Evaluation Gezeigt wird die verbesserte Skalierbarkeit und gesteigerte Sicherheit anhand von konkreten Verbesserungspotentialen.
Diffusion [Müller et al. 2016]

Tabelle 41: Übersicht Forschungsfrage 4

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe, insbesondere ohne die Hilfe eines Promotionsberaters, angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder erbrachten Dienstleistungen als solche gekennzeichnet.

Die vorgelegte Dissertation wurde weder im Inland noch im Ausland in gleicher oder in ähnlicher Form einer anderen Prüfungsbehörde zum Zwecke einer Promotion oder eines anderen Prüfungsverfahrens vorgelegt und wurde insgesamt noch nicht veröffentlicht.

_____ _____

Ort, Datum *(Unterschrift)*