Institut für Parallele und Verteilte Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Diplomarbeit Nr. 3415

Richtlinien für Datensicherheit von RDBMS in Cloud Diensten

Yi Xu

Studiengang:	Informatik
Prüfer/in:	Prof. Dr. Bernhard Mitschang
Betreuer/in:	Dipl.-Inf. Matthias Wieland
Beginn am:	2012-10-26
Beendet am:	2013-05-26
CR-Nummer:	H.2.4, H.2.7, H.3.5

Kurzfassung

Durch die schnelle Verbreitung von Cloud-Technologien und die fortschreitende Globalisierung stößt Cloud Computing auf wachsende Interessen und die Anzahl der Cloud-Benutzer stieg in den letzten Jahren rasant an. Während mehrere Kunden ihre Daten in die Cloud auslagern, verschärfen sich die Datensicherheitsprobleme aber auch gleichzeitig.

Das Ziel der vorliegenden Diplomarbeit war die Erstellung und Beschreibung von RDBMS-zentrischen Sicherheitsrichtlinien, die beim Betrieb der Cloud-Diensten eingesetzt werden, damit die Cloud-Dienste in der Zukunft im Rahmen von TOSCA mit garantierter Sicherheit und Compliance bereitgestellt werden könnten.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Im Einleitungskapitel wird zuerst die Motivation der Themenstellung erklärt und der Rahmen dieser Arbeit definiert. Anschließend folgt die Beschreibung der Zielsetzung dieser Arbeit. Am Ende des Kapitels wird die Struktur des Dokuments aufgezeigt.

1.1 Motivation

Durch die schnelle Verbreitung von Cloud-Technologien und die fortschreitende Globalisierung stößt Cloud Computing auf wachsendes Interesse und die Anzahl der Cloud-Benutzer stieg in den letzten Jahren rasant an. Nach einer Studie der Experton Group für BITKOM von Januar 2011 wird der Umsatz mit Cloud Computing in Deutschland von 1,9 Milliarden Euro im Jahr 2011 auf 8,2 Milliarden Euro im Jahr 2015 steigen [BIT11]. Es erläutert, dass sich der Trend der Verlagerung und Bearbeitung der Daten in einer virtuellen Umgebung in Zukunft noch schnell weiterentwickeln wird.

Besonders auffällig ist für die Unternehmen, dass Cloud Computing eine immer wichtigere Rolle für die Erreichung ihre Unternehmensziele spielt. Die Wirtschaftsprüfungs- und Beratungsgesellschaft PricewaterhouseCoopers (PwC) hat im Jahr 2011 eine Befragung über die Nutzung der Cloud-Dienste durchgeführt, an der insgesamt 351 Unternehmen teilgenommen haben [VG11]. Die Ergebnisse zeigen, dass fast alle befragten Unternehmen ihre Erwartungen an die Cloud entweder voll und ganz (46%) oder überwiegend (47%) erfüllt sehen [Kö11]. Der Grund liegt offensichtlich daran, dass Unternehmen mittels Cloud Computing in der Lage sind, sich unmittelbar an dynamische Marktbedürfnisse anzupassen und auf die Marktkonkurrenz schnell und flexibel zu reagieren, um damit Kosten einzusparen und Kapazitäten effektiver zu nutzen. Darüber hinaus verspricht es höhere Benutzerfreundlichkeit: kein spezielles Know-How ist mehr erforderlich für Administration und Support der Hardware im eigenen Haus, deswegen können die Unternehmen sich auf ihre Kernaktivitäten konzentrieren.

Gleichzeitig haben sich die Anforderungen der Cloud-Benutzer ebenfalls erhöht, um die Cloud-Dienste bequemer und ohne zusätzliche Sorge zu nutzen.

Abbildung 1.1 zeigt die Anteile von wesentlichen Erfolgsfaktoren für die Kundenzufriedenheit, welche aus Sicht der Cloud-Anbieter gegenwärtig aktuell sind. Die Informationen stammen von PwC aus dem Jahre 2010. Auffällig ist, dass Informationssicherheit und zuverlässige Serviceverfügbarkeit jeweils mit 96% und 88% (als eher wichtig) an ersten zwei Stellen stehen, d.h. zurzeit sind fast alle Benutzer über die gebotene Sicherheit von der Cloud-Dienste besorgt.

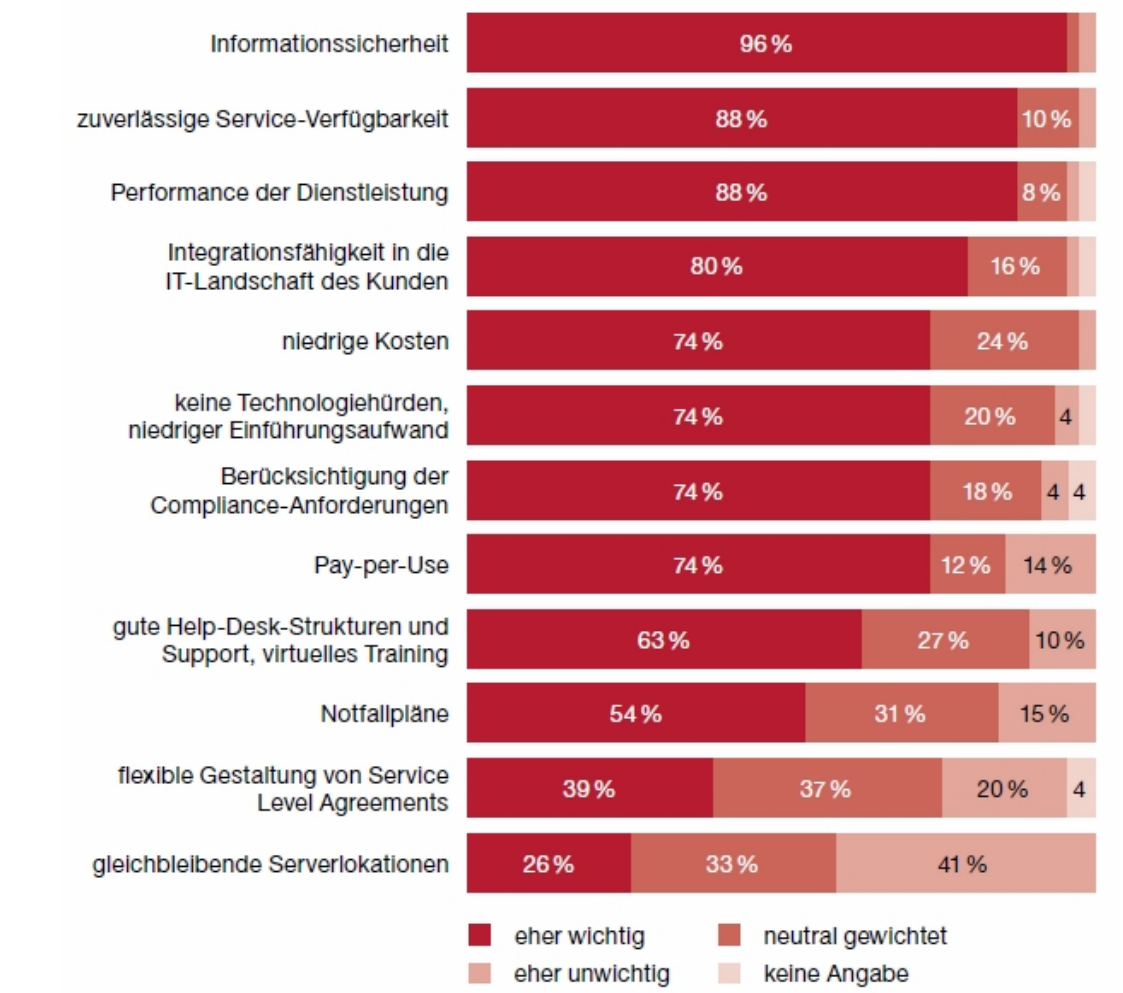

Abbildung 1.1: Wesentliche Faktoren für die Kundenzufriedenheit [VG10]

Die Ursache dafür kann man sich leicht vorstellen. Wegen der Gemeinsamkeit der Cloud (kurz gesagt, viele Benutzer könnten ihre Daten möglicherweise gerade auf der gleichen Hardware bzw. physischen Maschine speichern, bei der die Daten nur durch die virtuellen Partitionen getrennt werden) sind viele Benutzer allerdings noch zögernd, ihre Daten in eine unkontrollierbare und unsichtbare Umgebung auszulagern, die weder ihnen selbst gehört, noch von ihnen beaufsichtigt werden. Besonders gefährlich ist für Unternehmen, wenn die Geschäftsdaten in fremde Hände gelangen oder wichtige Systeme ausfallen. Sogar gäbe es das Risiko, falls die kritische Daten wegen des interessensorientierten Verkaufs von Cloud-Anbieter in die Hände von Konkurrenten fällen.

1.2 Zielstellung

Aus Anbietersicht ist die Sicherheits- bzw. Datenschutz-Anforderung momentan die größte Herausforderung des Cloud-Marktes. Jedoch sind die Sicherheitsprobleme in der Cloud vielfältig und können dadurch in verschiedene Aspekte wie Daten-, Netz-, Plattform- und Transportsicherheit usw. unterteilt werden. Diese Diplomarbeit fokussiert sich auf das Datenbankmanagementsystem in der Cloud und beschreibt im diesen Fall ausschließlich die RDBMS-zentralen Sicherheitsrichtlinien in Cloud-Diensten.

1.3 Verwandte Arbeiten

Vorliegende Diplomarbeit mit dem Schwerpunkt Datensicherheit von RDBMS in Cloud Diensten basiert auf dem Katalog funktionaler und nicht-funktionaler Sicherheitsanforderungen in CloudCycle aus Thomas Kunz et. al. [TK12], die die Cloud Diensten allerdings nicht spezifisch unter dem Aspekt der Datenbank, sondern Lebenszyklus-übergreifend beachtet hat. [Aba09, CPK10, Dat, Gö10, Han12, Rup10] bieten dazu mögliche Ergänzungen und Erweiterungen, sowohl im Bereich Cloud-Sicherheit als auch zum Thema Datenbankmanagement.

Heinz-Wilhelm Fabry stellt in [HWF] das TDE Verfahren für die Datenverschlüsselung vor, die ausschließlich bei Oracle Datenbank verwendet wird. In [MyS13a, Gab11, MyS13b] befinden sich das Datenverschlüsselungsverfahren für MySQL Datenbank. Und das entsprechende Vorgehen bei DB2 wird in [Sof] und [Cor13] beschrieben.

Abadi et. al. beschreiben in [Aba09] den Unterschied zwischen der transaktionalen Datenbank und der analytischen Datenbank, der in Kapitel 4 als eine der Taxonomien berücksichtigt wird. Andere Taxonomien werden in [Bey, Inf, Vio12] vorgestellt.

In [BBLS12, Com13a, Com13b, KBBL12] werden die Grundlage von TOSCA vorgestellt. Und das Ergebnis von dem Vergleich von Policy Sprachen zur Anwendung bei TOSCA wird in [SR12] von Renner et. al. dargestellt.

1.4 Aufbau der Arbeit

Diese Diplomarbeit ist inhaltlich in sechs Teile gegliedert: Kapitel 1 –Einleitung bietet zunächst einen Überblick über die Arbeit. Anschließend wurden die Ziele der vorliegenden Diplomarbeit verdeutlicht (1.2). Es folgen die verwandten Arbeiten(1.3) und der Aufbau dieser Arbeit (1.4). Kapitel 2 – Grundlagen geht auf die Grundlagen der Themen ein, die zum Verständnis dieser Arbeit hilfreich sind. Dazu gehören Cloud Computing (2.1), DBMS in Cloud Diensten (2.2), Datensicherheitsprobleme (2.3) und Einführung in TOSCA (2.4). Nach der Auflistung der möglichen Richtlinien für Datensicherheit und Erläuterung von Verschlüsselungstechnologien bezüglich verschiedener Datenbankprodukte in den Abschnitten 3.1 und 3.2 folgt die Taxonomie in Kapitel 4, die anhand dieses Katalogs von Richtlinien

abgeleitet werden und jeweils auf eine bestimmten Perspektive fokussieren. Kapitel 5 – Implementierung stellt die Umsetzung mit der Policy Sprache WS-Policy vor. Abschließend wird die Ergebnisse der Arbeit in Kapitel 6 – Zusammenfassung und Ausblick zusammengefasst und ein Ausblick auf mögliche Weiterentwicklung gegeben.

2 Grundlagen

In diesem Kapitel werden die Grundlagen und Technologien vorgestellt, die zum Verständnis der Datensicherheit von Relational Database Management System (RDBMS) in Cloud Diensten notwendig sind. Dieses Kapitel werden die folgenden Themen behandeln:

- 2.1 Cloud Computing
- 2.2 DBMS in Cloud Diensten
- 2.3 Datensicherheitsproblem
- 2.4 Topology and Orchestration Specification for Cloud Applications (TOSCA)

Die vorgestellten Konzepte werden nicht in voller Ausführlichkeit erklärt, sondern nur auf die Schwerpunkte konzentriert, die für das Verständnis dieser Arbeit von Nutzen sind. Den interessierten Lesern sei die referenzierte Literatur ans Herz gelegt.

2.1 Cloud Computing

Cloud Computing ist ein Ergebnis der Entwicklung der Technologien Distributed Computing, Parallel Computing und Grid Computing. Im Jahr 2011 veröffentlichte das National Institute of Standards and Technology (NIST) die Sonderveröffentlichung SP 800-145 [MG11], in der Cloud Computing wie folgt definiert wird:

„... a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models."

2.1.1 Grundlagen

Gemäß der NIST-Definition zeichnet sich Cloud Computing durch fünf essenzielle Charakteristiken aus [Inf, Vio12, Bus12]:

- **On-demand self-service:** Darunter wird verstanden, dass die Benutzer die Möglichkeit haben, sich die Ressourcen (z.B. Rechenkapazität, Speicherplatz) nach Bedarf und ohne Interaktion mit dem Dienst-Anbieter selbst zu bereitstellen.

- **Broad network access:** Alle Dienste sind über das Netzwerk verfügbar. Der Zugang zu der von einem Dienst-Anbieter zur Verfügung gestellten Ressource soll nicht abhängig von den Arten des Clients (z.B. Laptop, Smartphone, Tablets etc.) sein, sondern für möglichst viele Benutzer sollen die Ressourcen erreichbar und gemeinsam nutzbar sein.
- **Resource pooling:** Die Ressourcen des Cloud Anbieters liegen in einem Daten-Pool vor, und sie werden nach Bedarf über das sogenannte Multi-Mandanten-Modell dynamisch an die Benutzer verteilt. Dabei wissen die Benutzer nicht, wo sich die Ressourcen befinden. Sie können jedoch durch eine vertragliche Einigung den physikalischen Speicherort, z. B. Region, Land oder Rechenzentrum, festlegen.
- **Rapid elasticity:** Der Cloud-Anbieter wird die Ressourcen und Dienste schnell, dynamisch und elastisch bereitstellen, um den Bedürfnissen des Benutzers flexibel gerecht werden zu können. Aus Benutzersicht wären die Ressourcen deswegen nahezu unbegrenzt verfügbar. Unter dem wirtschaftlichen Aspekt kann sich auch sogenannte Green IT (Energie- und CO_2-sparende IT Landschaften) damit rechnen.
- **Measured Service:** Alle verbrauchten Ressourcen können gemessen, überwacht und berichtet werden. Dadurch wird eine entsprechende Diensttransparenz sowohl für den Kunden als auch den Cloud-Anbieter selbst geboten.

Zudem bietet Cloud Computing als Deployment-Models Private Cloud, Community Cloud, Hybrid Cloud und Public Cloud [Inf, Vio12].

- **Private Cloud:** Diese Cloud-Infrastruktur wird nur für eine bestimmte Organisation (z.B. innerhalb eines Unternehmers oder einer Universität) betrieben. Der Betrieb wird von der Organisation selbst oder von einem vertrauenswürdigen und zertifizierten Dritten geführt.
- **Public Cloud:** Die Public Cloud wird normalerweise von einer Organisation (z.B. von einem Unternehmen) betrieben, die ihre Dienste offen über das Internet entweder kostenlos oder kostenpflichtig für die Allgemeinheit anbietet. Und die angebotene Dienste sind beispielsweise Google-Docs, Dropbox, Microsoft Office 365.
- **Community Cloud:** Im Rahmen einer Community Cloud wird die Infrastruktur von mehreren Organisationen geteilt, die aus der gleichen Branche kommen und ähnliche Interessen haben. Der Betrieb kann durch eine dieser Organisation oder durch einen Dritten erfolgen.
- **Hybrid Cloud:** Von einer Hybrid Cloud spricht man die Mischform der beiden Private Cloud und Public Cloud (manchmal auch mit der Community Cloud), indem bestimmte, nicht geschäftskritische Dienste von öffentlichen Anbietern über das Internet zugegriffen werden, während kritische und sensitive Anwendungen und Daten im Unternehmen über Intranet und Extranet (z.B. für die Kunden, Lieferanten und Geschäftspartner des Unternehmens) betrieben werden.

Im Rahmen des standardisierten Architekturmodells der Cloud gibt es noch verschiedene Ausprägungen. Allerdings beinhalten alle Meinungen immer drei Schichten Anwendung, Plattform und Infrastruktur, die die Grundbausteine der Cloud Architektur bilden [Xan10]. Aus diesem Grund werden grundsätzlich drei verschiedene Servicemodelle von Cloud Computing aus Kundensicht angeboten.

- **Software as a Service (SaaS):** In dieser Form verkaufen die Hersteller Anwendungen über den Browser an Benutzer. Auf der einen Seite kann ein Benutzer seine Ausgaben senken, die vorher für die Einrichtung der Server und den Kauf zahlreicher Software notwendig sind. Auf der anderen Seite profitiert ein Dienst-Anbieter auch, da der Aufwand auf die Verwaltung einzelner Programme reduziert ist. SaaS steht häufig in den Anwendungen von Personalmanagement und ERP-Anwendungsprogramms zur Verfügung.
- **Platform as a Service (PaaS):** Es geht mehr um Middleware auf die der Benutzer seine Anwendungen installiert/deployed. Bei den Anwendungen selbst kann es sich dann z.B. um Entwicklungsumgebungen (IDEs) handeln. Mit dieser Hilfe von den Cloud-Anbieten entwickelt ein Unternehmen eigene Programme und danach setzt sie in der Herstellungs- und Managementprozess seiner Benutzern ein.
- **Infrastructure as a Service (IaaS):** Die Benutzer kontrollieren in diesem Fall nicht die grundlegende Infrastruktur, sondern die anderen Komponente wie Betriebssysteme, Speicher und Anwendungen, und verfügen möglicherweise auch über eine begrenzte Kontrolle über die Netzwerk-Komponenten (z.B. Host-Firewalls). Das bekannteste Produkt von IaaS ist wohl Amazon EC2.

2.1.2 Sonstige Eigenschaften und Vorteile

Nach der Cloud Security Alliance (CSA) hat Cloud Computing neben der oben erwähnten fünf Charakteristika noch zwei erweiterten Kriterien [Inf]:

- **Mandantenfähigkeit:** Mandantenfähigkeit gilt als eines der wichtigsten Merkmale der Cloud-Dienste. Sie bezieht sich auf das Prinzip in der IT-Infrastruktur, bei dem derselbe Server, dasselbe Softwaresystem oder eine einzelne Anwendung mehrere Kunden bedienen kann. In diesem Fall können die Kosten für Softwareentwicklung und -wartung zwischen den Mandanten geteilt werden, um die Infrastruktur wirtschaftlicher zu planen und betreiben. Dazu wird der Dienst-Anbieter jedem Mandanten eine mandantenspezifische Instanz des Systems zuordnen, die für jeden Mandanten individuell eingestellt wird, z.B. Service Level Agreement (SLA) der einzelnen Mandanten können untereinander unterschiedlich sein, damit auch die Dienstflexibilität garantiert werden kann.
- **Pay-per-Use Modell:** Der Cloud-Benutzer muss sich nicht mehr selbst um seine IT-Infrastruktur kümmern und wird nur für die tatsächlich verbrauchten IT-Ressourcen bezahlen. Insbesondere aus Sicht eines Unternehmens müssen die IT-Ressourcen wie Rechnerkapazitäten, Datenspeicher und Anwendungen nicht mehr in vollem Umfang

von ihm vorgehalten, betrieben und gewartet werden. Dadurch werden Betriebskosten gespart und die Wirtschaftlichkeit erhöht.

2.1.3 Klassifizierung von Daten in der Cloud

Bevor ein Cloud-Benutzer seine Daten auf einen Dienst eines Cloud-Anbieters überträgt, sollte er in erster Linie seine Daten klassifizieren und festlegen, welche Daten davon geeignet sind, bei einem Cloud-Dienst abgelegt zu werden.

- **Anwendungsdaten:** Die Anwendungsdaten bestehen aus allen Ressourcen bezüglich der Dienstinstanz. Einerseits enthalten sie die zur Bereitstellung und Verwaltung einer Dienstinstanz benötigten Daten, andererseits umfassen sie auch relevante Informationen, die wegen der Installation und der Konfiguration der Dienstinstanz sowie für das Backup-und Log-protokoll entstehen [TK12].
- **Nutzdaten:** Die von den Dienstinstanzen für die Cloud-Benutzer in der Cloud verarbeiteten und gespeicherten Daten werden als Nutzdaten beschrieben. Falls die Cloud-Benutzer eine Dienstinstanz wiederum einer anderen Benutzergruppe übertragen, werden dabei die generierten Daten von dieser Benutzergruppe auch als Nutzdaten genannt [TK12]. Teile von den Nutzdaten sind personenbezogen und auf deren Verarbeitung muss besonderes geachtet werden.
- **Kundendaten:** Im Rahmen der Kundendaten werden die Informationen bezeichnet, die der Cloud-Anbieter für die Aufbewahrung der Kundenunterlagen und Verwaltung bzw. Begrenzung des Kundenverhaltens speichert. Es enthält z.B. Kundenanmeldungsdaten, Zugangsdaten, Abrechnungsdaten, Berechtigungsverteilung, Authentisierungsdaten usw. [TK12]. Normalerweise sind die Kundendaten personenbezogene Daten und für die Verwaltung solcher Daten muss das Bundesdatenschutzgesetz beachtet werden.

2.2 DBMS in Cloud Diensten

Das Datenbankmanagementsystem ist ein wichtiger Baustein in Cloud Diensten, indem es sich um Datenverwaltungen kümmert.

2.2.1 Relationale DBMS

Das DBMS stellt unterschiedliche Werkzeuge bereit, mit welchen eine oder mehrere Datenbanken erstellt, mit Daten gefüllt und verwaltet werden können. Ein DBMS verfügt in der Regel über mindestens eine Benutzerschnittstelle und kann eine oder mehrere Datenbanken nur aufeinanderfolgend oder parallel verwalten. Ferner gibt es - explizit oder implizit - Regeln, wer zu welchem Zeitpunkt Datenzugriff und Änderungsrecht an den Daten hat. Schließlich können Sicherungsstrategien / Backup-Verfahren definiert und regelmäßig ausgeführt werden [Aue13].

Wegen der Gewährleistung der Atomicity-Consistency-Isolation-Durability (ACID) Eigenschaften für Transaktionen und der Unterstützung bzgl. Ad-hoc Abfragen, verwenden Geschäftsanwendungen typischerweise relationale DBMS für die Datenverwaltung.

Und um die Funktionalitäten des traditionalen DBMSs an die Cloud anzupassen, entwickeln die Datenbankanbieter auch viele neue Produkte. Beispielsweise steht bei der relationalen Datenbank SQL Azure [Sch], die auf der MS SQL Servertechnologie beruht, gewohnte Konzepte wie Tabellen, Views, Joins, Stored Procedures etc. zur Verfügung, diese werden durch z. B. verteilte Abfragen erweitert [BT11].

2.2.2 Datenmanagementverfahren in der Cloud

Wenn das DBMS in die Cloud übernommen wird, werden verschiedene Datenmanagementverfahren gemäß den drei Servicemodellen eingestellt.

Wie in Abbildung 2.1 (a) gezeigt, stellt der Einsatz von traditionellen relationalen Datenbanksystemen (RDBMS) im Rahmen eines IaaS Servicemodells die einfachste Möglichkeit dar. Im diesen Fall wird für jede virtuelle Maschine (VM) ein einzelnes DBMS installiert, das die Datenverarbeitung und -speicherung für die darauf installierte Dienstinstanz übernimmt. Beispielsweise bietet Amazon EC2 [Bey] solche Infrastrukturen, indem die verschiedene VMs parallel auf der gemeinsamen Hardware eines IaaS-Anbieters laufen.

Da IaaS eine sehr hohe Skalierbarkeit anbieten kann, haben die Benutzer dabei die größten Freiräume, die DBMS selbst zu administrieren. Im Gegensatz dazu müssen die Benutzer auch eigenständig für die Verwaltung und Pflege des DBMSs verantwortlich sein [SK12]. Aus diesem Grund erhöht sich neben den Betriebskosten auch das Sicherheitsrisiko. Um dieses Dilemma zu beseitigen, haben die Cloud-Anbieter wie Amazon und Microsoft einige Produkte weiter entwickelt. Beispielsweise hat das Amazon Relational Database Service (RDS) nicht nur die notwendigen Infrastruktur zum Betreiben einer MySQL-Datenbankinstanz bereitgestellt, sondern auch einige administrative Aufgaben (z.B. automatische Erzeugen von Backups) von Benutzer übernommen, damit die Benutzer von Amazon RDS von erforderlichen Verwaltung- und Routineaufgaben im Vergleich zum Eigenbetrieb einer entsprechenden MySQL-Datenbank befreit werden [BT11].

Die Datenspeicherung bei PaaS ist typischerweise stark mit der umgebenden Plattform verwoben [SK12], im Vergleich zu IaaS wird daher das RDBMS und die administrative Datenbanktätigkeiten an den Dienstleister ausgelagert, und über die vom Cloud-Anbieter angebotene Schnittstellen können die plattformbasierten Anwendungen auf die Datenbank zugreifen.

Bei Abb. 2.1 (c) werden die Benutzer üblicherweise mittels eines Thin-Client, z.B. Webbrowser, auf die Anwendungen zugreifen. Und die Datenmanipulation erfolgt dann über den Zugriff der Anwendung. Das bedeutet, dass der Cloud-Anbieter für das komplette Datenmanagement von aller seiner Benutzer (auch Mandanten genannt) zentral zuständig ist. Aus diesem Grund muss ein SaaS-RDBMS neben dem Ergreifen der allgemeinen Sicherheitsmaßnahmen

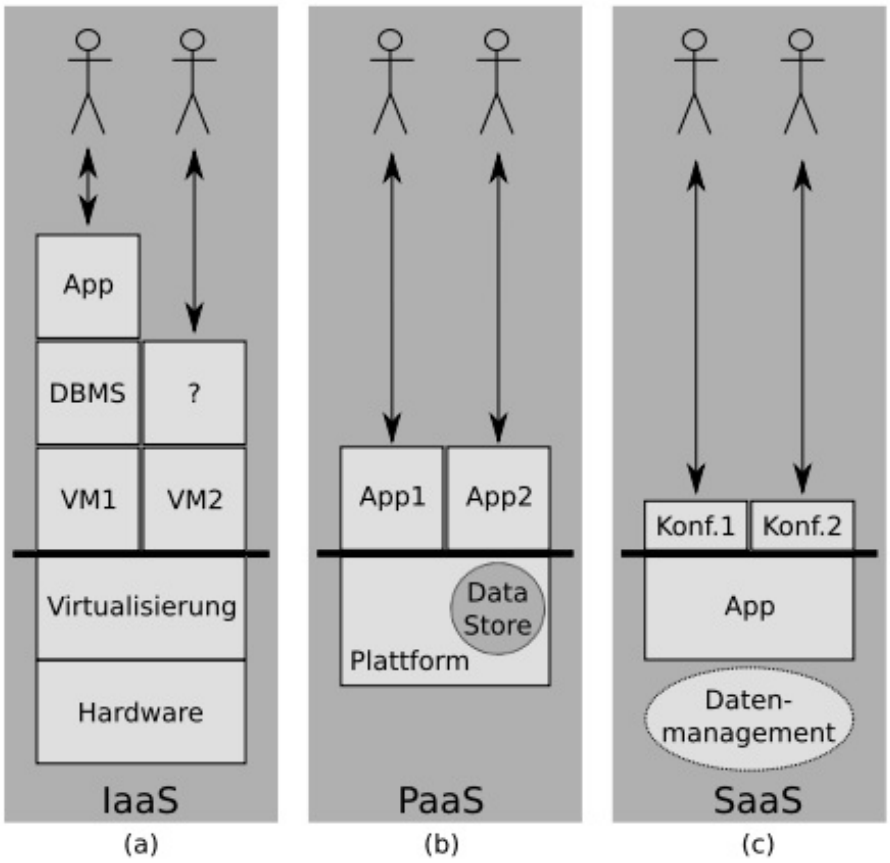

Abbildung 2.1: DBMS bei verschiedenen Service Modellen [SK12]

bei derartig logischer Trennung von Mandantendaten noch die Fähigkeit haben, vor mandantenübergreifenden Fremdzugriffen zu schützen. Üblicherweise verwenden die Cloud-Anbieter Maßnahmen wie Authentifizierung, Daten und Speicherplatz-Verschlüsselung [Gö10], um solche Sicherheitskriterien zu erfüllen. Zudem kann auch Discretionary Access Control (DAC) Modell (z.B. durch ein Befehl GRANT zur Vergabe von Privilegien) [KE11] und Mandatory Access Control (MAC) Modell eingesetzt werden, um die Zugriffskontrolle für Tabellenzeilen und -spalten zu realisieren. MAC wird von vielen RDBMS in Form von Label Security unterstützt [Gö10].

2.3 Datensicherheitsproblem

Der folgende Abschnitt spricht über das Sicherheitsproblem. Im Unterabschnitt 2.3.1 werden die allgemeinen Cloud-Sicherheitsbedrohungen kurz erläutert. Darüber hinaus werden die Gründe für das Datenbank-zentrale Sicherheitsproblem im Unterabschnitt 2.3.2 aufgezählt.

2.3.1 Allgemeines Sicherheitsproblem beim Cloud Computing

Wenn wir über die Sicherheitsbedrohung bei den Cloud-Diensten diskutieren, soll betont werden, dass viele Begriffe, die man als Schattenseite der Cloud betrachtet, grundsätzlich nicht nur bei der Cloud sondern auch in allen internetübergreifenden Bereichen vorkommen. Die typischen Beispiele umfassen z.B. den Datenmissbrauch, downtime, Phishing [Kno09], Internet-Betrug usw.. Derartige Probleme richten sich vielmehr auf die Netzwerksicherheit und die Datenverwaltung. Im Gegensatz dazu existieren auch einige Cloud-spezifische Sicherheitsherausforderungen. Die Analyse wird generell unter folgenden drei Aspekten erfolgen.

1. **Der Ursprung der Bedrohung liegt innerhalb der Cloud**

   Wenn ein Benutzer seine kritischen Daten in der Cloud speichert, lagert er auch die Sicherheitsaufgaben aus. Aber gleichzeitig könnte insbesondere bei PaaS und SaaS der Systemadministrator eines Cloud-Anbieters sehr leicht auf die bei ihm gespeicherten Daten zugreifen. Falls zureichende Vereinbarungen bei den SLAs zwischen dem Service-Anbieter und dem Kunden fehlen, wäre es für einen Unternehmen vornehmlich sehr gefährlich, seine sensitiven Geschäftsgeheimnisse möglicherweise über den interessensorientierten Verkauf des Cloud-Anbieters in die Hände seiner Wettbewerber fallen zu lassen. Deswegen ist ein zuverlässiger Cloud-Anbieter mit vollständigen SLAs die Voraussetzung für den ordentlichen Betrieb der Cloud-Dienste.

2. **Der Ursprung der Bedrohung kommt außerhalb der Cloud**

   Aufgrund der Mandantenfähigkeit und Virtualisierung werden die Soft- und Hardware-Komponenten des Cloud-Betreibers in der Regel von mehreren Benutzern gemeinsam genutzt und nur durch die Partitionierung von virtuellen Maschinen (VM) logisch getrennt. Daher könnte ein Angreifer beispielsweise seine bösartige VM auf der gleichen physischen Maschine wie sein Target (auch als co-residence genannt) einstellen und dann über die Cross-VM Angriffe (z.B. den Seitenkanalangriff) das Password des Targets entziffern, und sich damit die gespeicherte kritische Informationen aneignen [RTSS09].

3. **Reputation fate-Sharing**

   Während die Cloud-Benutzer potenziell von fate-Sharing profitieren, bringt es gleichzeitig auch unkalkulierbares Risiko, dass nur eine kleine Störung bei dem Cloud-Anbieter eventuell zahlreichere Benutzer beeinflusst. Beispielsweise hat ein Spammer Amazon EC2 angegriffen und einer der Cloud-Server ist zum Versenden von größeren Mengen Spam missbraucht worden. Dies verursacht, dass einen großer Teil der IP-Adressen von EC2 auf eine schwarze Liste vom Spamhaus gesetzt werden. Im diesen Fall werden sogar viele Dienste unterbrochen. Falls ein Benutzer eine Email aus EC2 senden wollte, musste er zuerst ein Formular (`http://aws.amazon.com/contact-us/ec2-email-limit-request/`) ausfüllen und dann eine Liste der EC2 Adresse für Authorization anbieten, um die Adresse aus der schwarzen Liste zu entfernen [CPK10].

2.3.2 Datenbankspezifische Sicherheitsprobleme

Der Lebenszyklus von Daten besteht aus Erzeugung, Speicherung, Bearbeitung, Übertragung und Zerstörung der Daten. Das Thema Datensicherheit umfasst alle Schutzmechanismen gegen Fehler und Missbrauch rund um den Daten-Lebenszyklus.

Auf der anderen Seite lassen sich die Cloud-spezifischen Datenschutz-Risiken laut European-Privacy-Seal (eine Initiative des Unabhängigen Landeszentrums für Datenschutz (ULD)) in zwei Hauptgruppen unterteilen: „solche, die einen Kontrollverlust über personenbezogene Daten bei der Nutzung von Cloud-Diensten mit sich bringen und solche, die sich auf eine unzureichende Information über das ‚Wie', ‚Wo' und ‚Durch wen' der Verarbeitung personenbezogener Daten beziehen" [Sea12].

Solche Cloud-spezifischen Risiken ergeben sich aus den folgenden Charakteristiken der Daten in der Cloud:

- In der Cloud werden die Daten auf einem Server gespeichert, der sich möglicherweise in einer anderen Region sogar in einem anderen Land als der Benutzer befindet, und ein Benutzer werden normalerweise auch nicht mitgeteilt, wo seine Daten gespeichert worden sind. In diesem Fall hat ein Benutzer kaum die Möglichkeit, seine Daten nachzuvollziehen, und er verlieret damit die Kontrolle über seine Daten.
- Um die Fehlertoleranz und die Verfügbarkeit eines Diensts zu garantieren, bemühen sich die Cloud-Anbieter um viele Schutzmaßnahmen gegen Knoten-Ausfall. Hauptsächlich werden die Daten in der Cloud redundant gespeichert, wodurch der Dienst seine benötigten Daten bei einem Knoten-Ausfall von anderen Knoten rechtzeitig erhalten und noch problemlos betreiben kann. Da solche Datenkopien auf verschiedenen Knoten gespeichert werden, die sich oft über große geographische Distanzen verteilen, bringt die Synchronisation der Replikate Probleme mit sich: sollen alle Knoten die aktuellen Daten nach write „letztendlich" lesen (Eventual Consistency), oder „sofort" lesen (Strong Consistency) [Tho11])?
- Die Rechenleistung einer Cloud ist elastisch, nur dann wenn die Aufgaben parallel bearbeitet werden können (shared-nothing Architektur).
- Daten könnten bei einigen spezifischen Operationen (z.B. Übertragung, Kopieren und ggf. Aggregation [Aba09]) in der Cloud direkt bearbeitet werden, obwohl sie schon verschlüsselt sind, um die Bearbeitungsperformance zu erhöhen. Da es für eine Anwendung bei solchen Operationen nicht mehr nötig ist, die verschlüsselten Daten zuerst von dem Cloud herunterzuladen und sie nach der Entschlüsselung und Bearbeitung wiederum in die Cloud hochzuladen. Aber im Gegenzug steigt das Sicherheitsrisiko mit der eventuellen Verarbeitung der verschlüsselten Daten.

2.4 Topology and Orchestration Specification for Cloud Applications (TOSCA)

Um die Kompatibilität von Cloud-Diensten bei verschiedenen Cloud-Anbietern zu ermöglichen, definiert OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) ein standardisiertes Verfahren [Wei12], indem die Service Komponenten, ihre Beziehungen und die Managementvorgänge während des ganzen Service-Lebenszyklus eines Dienstes von TOSCA in einer spezifischen Sprache normiert werden können. Die in Kapitel 3 dieser Arbeit erstellten Sicherheitsrichtlinien können später als Policies in TOSCA umgesetzt werden und für alle Cloud-Dienste gelten.

2.4.1 TOSCA Grundlagen

Da TOSCA alle bestehende Service Komponenten und ihr Management in eine modulare und portable Form beschreibt [BBLS12], werden Service Templates als grundsätzliche Bestandteile definiert. Und solche Service Templates werden schließlich in einer TOSCA-tauglichen Umgebung interpretiert, die die Cloud-Dienste betriebe und ihre Instanzen verwalte.

Wie in Abbildung 2.2 dargestellt, besteht ein standardisiertes TOSCA Service Template aus Topology Template, seine operationalen Aspekte (z.B. sein Deployment usw.) und Plänen. Das Topology Template beschreibt die Topologie eines modellierten Cloud-Dienstes, welche aus den Knoten Elementen besteht, die die verschiedene Service Komponenten darstellen, und den Beziehung Elementen, welche die Beziehungen zwischen Knoten definieren. Sowohl ein Knoten als auch eine Beziehung besitzt eine Reihe von Eigenschaften, die entsprechend als Typ festgelegt werden. Im Rahmen eines Plans werden verschiedene mögliche Workflows gezeigt und jeder Workflow kann durch eine spezifische Workflow Sprache wie Business Process Model and Notation (BPMN) spezifiziert werden.

2.4.2 Vorteile

Durch die Realisierung der Portabilität der Cloud-Diensten und die Abschwächung der Anbieter „lock-in“ Situation [KBBL12] ermöglicht TOSCA [Com13a]:

- Portable und unabhängige Deployment auf irgendeine kompatible Cloud
- Problemlose und einfachere Migration existierender Anwendungen in eine andere Cloud
- Flexible Lastspitzenskalierung (nach Benutzerwunsch)
- Dynamische Anwendungen, die auf einer durch mehr als ein Cloud-Anbieter unterstützten Plattform betrieben werden können

Davon können nicht nur die Cloud-Anbieter und ihre Entwickler sondern auch die Cloud-Benutzer profitieren.

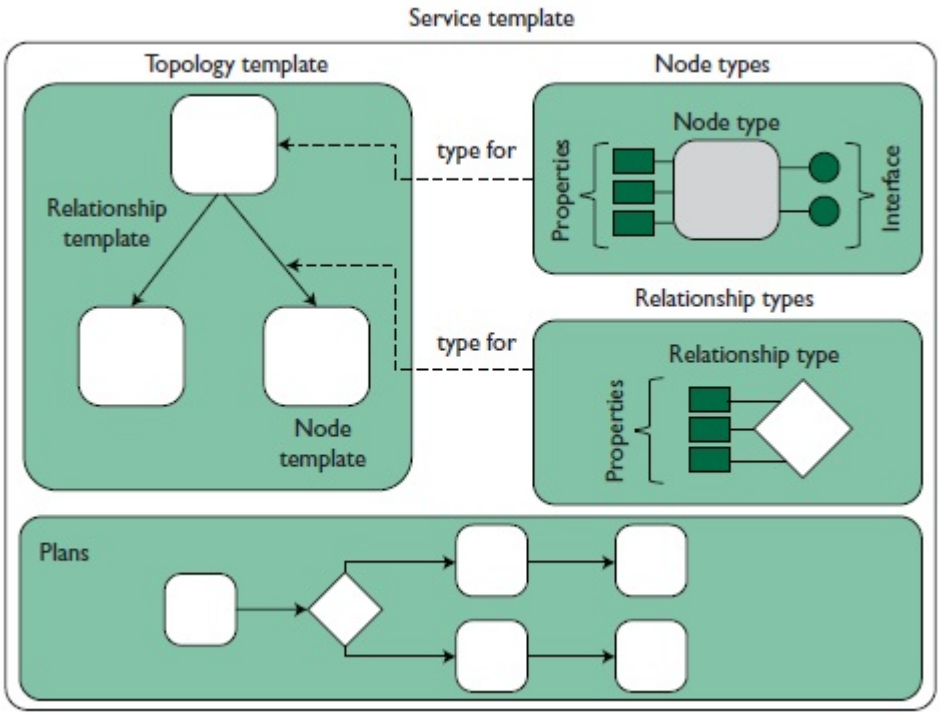

Abbildung 2.2: TOSCA Service Template [BBLS12]

2.4.3 TOSCA Policy Sprache

Alle nicht-funktionale Anforderungen und Quality-of-Services (QoS) werden in TOSCA als Richtlinien (engl. Policy) definiert [Com13b].

Ein Knoten Template (Service Komponente) wird mit einer Reihe von Richtlinien verbunden, die die nicht-funktionalen Anforderungen für diesen Knoten darstellen.

Die in einer Anforderung beinhalteten Bedingungen bzw. Eigenschaften werden entsprechend in einem Policy Type vordefiniert. Beispielsweise werden alle Zahlungsbedingungen wie Zahlungsmittel, Zahlungsfrist und Zahlungsbetrag im Rahmen eines Policy Types der Richtlinie Zahlungsinformationen bestimmt. Und nach der Festlegung dieses Policy Types bieten die entsprechenden Policy Templates konkrete Werte für die in diesem Policy Type bestimmten Bedingungen, z.B. „US$“ für das „Zahlungsmittel“ und „ein Monat“ für die „Zahlungsfrist“. Und die Bedingung „Zahlungsbetrag“ bleibt zuerst offen und danach bei verschiedenen Knoten Templates konkretisiert [Com13b].

Die konkreten Policy Type und Policy Templates für die Richtlinien der Datensicherheit werden in Kapitel 5 gezeigt.

3 Sicherheitsrichtlinien und Maßnahmen

In diesem Kapitel soll dem Leser zuerst ein grober Überblick verschafft werden, welche Richtlinien in dieser Arbeit festgelegt werden. Dazu wurden einige Richtlinien aus [TK12] übernommen. Danach wird jede Richtlinie mit den dazu eventuell eingesetzten Maßnahmen in ihrem Unterkapitel genauer erklärt. Zum Ende des Kapitels werden einige von verschiedenen Softwareanbietern entwickelten Datenverschlüsselungstechnologien vorgestellt, die zurzeit häufig zum Einsatz kommen.

3.1 Richtlinien mit den evtl. eingesetzten Technologien bzw. Maßnahmen

3.1.1 Transparenz	RTP1: Log Daten Nachvollziehbarkeit
3.1.2 Datenvermeidbarkeit	RDM1: Klartext Vermeidbarkeit
3.1.3 Datensparsamkeit	RDS1:Aufbewahrungsfrist
	RDS2: Datenartenabhängigkeit
3.1.4 Zweckbindung	RZB1:Mitteilungspflicht
	RZB2:Ausschließlichkeit
3.1.5 Vertraulichkeit	RVT1:Zugriffsberechtigung
3.1.6 Integrität	RIG1:Manipulationen Check
	RIG2:Datenintegrität
	RIG3:SchlüsselIntegrität
3.1.7 Verfügbarkeit	RVF1:Ressourcenverfügbarkeit
	RVF2:Unbeeinträchtigter Zugriff
	RVF3:Anwendungsverfügbarkeit
3.1.8 Authentizität	RAT1:Authentizität der Daten
	RAT2:Authentizität des Kommunikationspartners
3.1.9 Pseudonymität	RPD1:Pseudonym Eindeutigkeit
	RPD2:Pseudonym Einloggen
	RPD3:Informationanzeige
3.1.10 Angemessenes Datenschutzniveau	RSN1:Datenauslagerung im Ausland
3.1.11 Überprüfung von Benutzereingaben	RÜE1:Http-Anfrage Überprüfung
3.1.12 Referenzen direkt auf interne Objekte	RRI1:Referenzen Vermeidbarkeit

3.1.13 Einsatz von hochmodernen Hypervisoren	REH:Virtual Machine Monitor
3.1.14 Regelmäßige Datensicherungen	RRS1:Offline Backups
	RRS2:Datenwiederherstellungsfähigkeit
	RRS3:Online Backups
	RRS4:Speicherung der Back-up Daten
	RRS5:Log Daten Nachvollziehbarkeit
	RRS6:Speicherungsfrist
3.1.15 Standard-Datenformat	RSF1:Format Standardisierung
3.1.16 Isolierung	RII1: Isolierung der Dienstinstanzen
	RII2:Isolierung der Kunden-Daten
3.1.17 Löschung der Anwendungsdaten und Nutzdaten	RLD1:Löschung der Anwendungsdaten
	RLD2:Löschung der Nutzdaten
3.1.18 Schlüsselaufbewahrung und -management	RSA1:Schlüssel Isolierung
	RSA2:Datei Verschlüsselung
	RSA3:Schlüssel Zweckverbindung
3.1.19 Überwachung von Datenbankaktivitäten	RÜD1:Datenaktivität Monitoring
3.1.20 Konsistenz (engl. Concurreny) nach der Synchronisation	RKS1:Datensynchronisierung
	RKS2:Sperrung während der Datensynchronisierung

Tabelle 3.1: Richtlinien Liste

3.1.1 Transparenz

Das Prinzip „Transparenz“ beschreibt, dass ein Cloud-Anbieter jedem Betroffen mitteilen muss, welche Daten von ihm für wie lange und aus welchem Grund genutzt werden.

Richtlinien

- **RTP1**: Log Daten (wann, wo, von wem und wie) für die Erstellung, die Verarbeitung und das Löschen der kritischen und personenbezogenen Daten müssen gespeichert werden, kontrollierbar und nachvollziehbar sein [TK12].

Evtl. entsprechende Maßnahme

- Diese Anforderung könnte mithilfe von sicheren Logging-Frameworks (z.B. log4j für Java, ObjectGuy Framework, Simple Log, TraceTool usw.) verwirklicht werden. log4j-Ansatz ist eine der bekanntesten Logging-Framework-Implementierung, die die Logging-Information möglichst effizient filtern und ausgeben kann. Wenn log4j mit log4view zusammen ins Spiel kommen, sind die Entwickler noch in der Lage, die Log-Meldungen parallel an mehrere Ziele (z.B. an eine Datei, an das Netzwerk oder an eine Datenbank) oder an bestimmte Empfänger zu schicken [Gmb13]. Darüber hinaus gibt es noch Wrapper Tools, um Logging-APIs zu kapseln. Wrapper um Logging-Frameworks bieten die Möglichkeit an, dass der Endbenutzer noch konfigurieren kann, welches Logging-Framework benutzt werden soll [Tor11].

3.1.2 Datenvermeidbarkeit

Richtlinien

- **RDM1**: Daten sollten so oft wie möglich nicht im Klartext erscheinen [TK12].

Evtl. entsprechende Maßnahme

- Geschäftskritische und personenbezogene Daten werden verschlüsselt. Die konkreten Datenverschlüsselungsverfahren werden später im Unterkapitel 3.2 ausführlich erläutert.

3.1.3 Datensparsamkeit

Richtlinien

- **RDS1**: Daten sollen nicht für unbegrenzte Zeit aufbewahrt werden. Das bedeutet, dass nicht mehr zur Erreichung eines rechtmäßigen Zwecks benötigte Daten gelöscht oder mindestens gesperrt werden sollen [TK12].
- **RDS2**: Nach unterschiedlichen Datenkategorien sollen unterschiedlich lange Aufbewahrungsfristen festgestellt werden.

Evtl. entsprechende Maßnahme

- Es soll ein bestimmter Zeitraum so eingerichtet werden, dass die Daten gelöscht oder gesperrt werden müssen, wenn sie während dieser Frist doch niemals bearbeitet bzw. abgerufen geworden sind. Und das grundsätzliche Kriterium lautet: So kurz wie möglich, so lange wie nötig [Dat].

3.1.4 Zweckbindung

Richtlinien

- **RZB1**: Bevor die Cloud-Anbieter die personenbezogenen Daten erheben oder verarbeiten, müssen die damit verbundenen Zwecke dem Betroffenen mitgeteilt werden [TK12].
- **RZB2**: Jeder Datenverarbeitung muss ein bestimmter Zweck zugrunde liegen Die empfangende Stelle ist darauf hinzuweisen, dass die Daten nur zu diesen Zwecken verarbeitet werden dürfen, für die sie übermittelt wurden [LNW00] (Ausnahme liegt nur beim Fall einer im Voraus erteilten freiwilligen Einwilligung des Betroffenen).

Evtl. entsprechende Maßnahme

- Die Manipulation bestimmter kritischen Spalten in den Datenbanktabellen würde der Cloud-Anbieter den betroffenen Benutzern z.B. per Email benachrichtigen. Kritische Spalte bedeutet, dass diese Spalte benutzerbezogen kritischen Daten enthalten. Dazu gehören z.B. Personalausweisnummer, Bankkontonummer, Geschäftsgeheimnis usw. Die Vereinbarung über die Benachrichtigungsart können der Cloud-Anbieter und die Kunden über SLA treffen.

3.1.5 Vertraulichkeit

In Cloud-Umgebung sind die Daten häufig in Bewegung, da Anbieter von Cloud-Ressourcen zur Optimierung ihrer Infrastrukturkapazität und Gewährleistung der Performanz die Daten auf von ihnen festgestellten Rechnern speichern können (die Landesgrenzen werden oft überschritten) und diese Daten auch duplizieren dürfen müssen. Wenn die Rechner sich in einem Staat befinden, in dem der Datenschutz nicht denselben Stellenwert wie in Deutschland besitzt, führt zu Vertraulichkeitsproblemen.

Richtlinien

- **RVT1**: Weder unberechtigter Zugriff auf gespeicherte Daten noch unautorisierte Informationsgewinnung ist erlaubt.

Evtl. entsprechende Maßnahme

- Es überlassen dem Kunden selbst, Daten über primäre Geschäftsprozesses im Unternehmen, Behörden- und Verwaltungsdaten, personenbezogene Daten und gesetzlich geregelte Daten immer auf dem Speichermedium zu verschlüsseln (homomorphe Verschlüsselung) oder in der Datenbank vorzulegen.
- Die Übertragung solcher Daten über ein Netzwerk sollte gemäß vorgegebenen Berechtigungen ausgeführt werden und kontrollierbar sein.
- Im Fall einer bilateralen Cloud-Szenario (ein Konsument und ein Anbieter) [Rup10] wird die Vertraulichkeit durch bestehende Verfahren wie beispielsweise SSL/TLS zur sicheren Datenübertragung zugesichert.
- Die Fragmentierung der Informationen und die Verteilung der Daten auf verschiedene Clouds (Beispielsweise kann man sensiblere Daten in einer vertrauenswürdigen Private Cloud (z.B. Amanzon Virtual Private Cloud mit der Einrichtung von mehreren Sicherheitsebenen [Ama]) und übrige Daten in Public Cloud verarbeiten lassen) bieten eine weitere Lösung.

3.1.6 Integrität

Richtlinien

- **RIG1**: Daten, die in der Cloud auf virtuellen, gemeinsam genutzten Festplattenspeichern abgelegt sind, müssen vor unbemerkten, nicht autorisierten Manipulationen (z.B. cross-VM side-channel attack) geschützt werden.[TK12].
- **RIG2**: Während der Verarbeitung sollten die Daten vollständig und unversehrt sein.
- **RIG3**: Die (Kunden-) Schlüsselverteilung muss integer sein.

Evtl. entsprechende Maßnahme

- Schutz vor unautorisierten Manipulationen: Prüfsummen oder Signaturen können über die Dateien mitgespeichert werden, um Veränderungen aufdeckbar zu machen [Han12].

3.1.7 Verfügbarkeit

Verfügbarkeit wird durch die DIN 40042 als Wahrscheinlichkeit definiert [Rup10]:

„Die Verfügbarkeit ist die Wahrscheinlichkeit, ein System zu einem gegebenen Zeitpunkt in einem funktionsfähigen Zustand anzutreffen“.

Richtlinien

- **RVF1**: Ein genehmigter Zugriff auf die benötigten Ressourcen soll zu jeder Zeit möglich sein.
- **RVF2**: Solche Zugriffe sollten nicht durch unautorisierte Aktionen oder gezielte Angriffe externer Akteure beeinträchtigt werden [SR09].
- **RVF3**: Der Administrator von einem Dienstanbieter muss jederzeit sicherstellen, dass eine Anwendung eines Kunden auf Basis des Back-ups wiederherstellbar ist. [TK12].

Evtl. entsprechende Maßnahme

- Duplizierte Daten sollten auf mindestens zwei örtlich getrennten Datenbanken abgelegt werden, die sogar in verschiedenen Ländern liegen und mit verschiedenen Schlüsseln bzw. Verschlüsselungstechnologien verschlüsselt werden.
- Die Cloud-Rechenzentren sollten redundant vernetzt werden.
- Checkpoint-und-Recovery-Mechanismen.
- Begrenzung der Ressourcen für einen einzelnen Benutzer zur Vermeidung der verteilten Denial-of-Service-Angriffe [SR09].

3.1.8 Authentizität

Richtlinien

- **RAT1**: Die Sicherstellung der Authentizität enthält die Anforderung, kritische Informationen jederzeit ihrem ursprünglichen Sender sicher zuordnen zu können und dass sie nach der Erstellung und dem Versand nicht mehr verändert worden sind.
- **RAT2**: Die bilateralen Kommunikationspartner sollten identifiziert sein und die (Kunden-) Schlüssel müssen authentifiziert verteilt werden.

Evtl. entsprechende Maßnahme

- Die Authentifizierung zwischen dem Cloud-Benutzer und dem Cloud-Anbieter kann anhand des Austauschs von Authentifizierungsdaten (wie z.B. digitale Signaturen, Sicherheitstoken) erfolgreich durchgeführt werden, die getrennt und unabhängig von der Übertragung der tatsächlichen benötigten Informationen eingesetzt wird.

3.1.9 Pseudonymität

Anonymität ist zwar hilfreich für die Datensicherheit, kann aber für Cloud-Dienste nur eingeschränkt angewendet werden, da für die Abrechnung der verbrauchten Ressourcen wie Rechnerkapazitäten und Speicherplatz detaillierte Benutzerprofil erfolgen müssen. Deswegen sollte in Cloud-Computing-Systemen Pseudonyme vorgezogen werden [SR09].

Richtlinien

- **RPD1**: Das Pseudonym sollte eindeutig und identifizierbar sein, um die Abrechnungsfunktion aufzudecken.
- **RPD2**: Öffentlich werden bei einem Cloud-Dienst nur pseudonymisierte ID gezeigt. Und ohne Login kann niemand auf die Benutzerinformationen zugreifen.
- **RPD3**: Durch das Einloggen über das Pseudonym mit Passwort darf nur der korrekt authentifizierte Dienste-Benutzer seine Rechnungsinformation prüfen und sein Benutzerprofil einstellen oder verändern.

Evtl. entsprechende Maßnahme

- Im RDBMS kann diese Pseudonym-ID in einer Tabelle wie z.B. Benutzerprofil als Primärschluss gespeichert werden, und in referenzierenden Tabellen wie Rechnung als Fremdschlüssel verwiesen werden. Dadurch wird auch die referentielle Integrität der Daten gewährleistet.

3.1.10 Angemessenes Datenschutzniveau

Die folgende Abbildung 3.1 hat gezeigt, dass einerseits nur etwa jeder zweite Cloud-Anbieter seine Infrastruktur (Server und Rechenzentren) in Deutschland gelegt haben, wo in der Regel ein höheres Sicherheitsniveau garantiert wird. Andererseits wissen lediglich zwei Drittel der Cloud-Benutzer (siehe Abb. 3.2), in welchem Land sich die Infrastruktur befindet, in der ihre ausgelagerten Daten abgelegt werden [VG11]. Das bedeutet, dass vielen Kunden die Informationen über den Speicherstandort ihrer Daten fehlen.

Richtlinien

- **RSN1**: Übermittlung benutzerrelevanter Daten an Stellen außerhalb der Staaten des Europäischen Wirtschaftsraums (EWR) ist nur zulässig, wenn dort ein angemessenes Datenschutzniveau gewährleistet ist [LNW00, TK12], oder wenn bestimmten Voraussetzungen erfüllt werden, z. B. bei Verwendung sogenannter Standardvertragsklauseln oder verbindlicher Unternehmensregelungen [DSH12].

Abbildung 3.1: Lokation der Daten [VG10]

Abbildung 3.2: Kenntnis der befragten Nutzer über den Standort der Rechenzentren, in dem ihre Daten gespeichert werden [VG11]

Evtl. entsprechende Maßnahme

- Wenn ein Cloud-Anbieter Daten in mehreren Ländern speichert, sollte er ihren Kunden die Entscheidung darüber überlassen, in welchem Land ihre Daten gespeichert werden können und ob die Daten ausschließlich in Deutschland gespeichert werden sollen [VG10].

3.1.11 Überprüfung von Benutzereingaben

SQL Injektion Angriffe beschreiben das Angriffsverfahren in Zusammenhang mit der relationalen Datenbank, die durch fehlende Überprüfungen von Schade-Code in Benutzereingaben

ermöglicht (siehe Abb. 3.3). Das bedeutet, dass der Angreifer über das Benutzereingabefeld eines Dienstes versucht, das die Zugriffsmöglichkeit auf die Datenbank bietet, eigene schädliche Datenbankbefehle einzuschleusen. Sobald diese Eingaben ungeprüft in eine SQL-Abfrage eingefügt werden, könnte er durch solche SQL-Befehle die Datenbank auf seinem Wunsch manipulieren und sogar Kontrolle über den Server erhalten. Beispielsweise können folgende Fälle durch einen erfolgreichen SQL- Injektion-Angriff erreicht werden [Kaco8]:

- Hinzufügen, Ändern, Löschen von Datensätzen, Tabellen und Datenbanken
- Auslesen und Stehlen von Datensätzen
- Erzeugen von Dateien auf dem Dateisystem der Anwendung

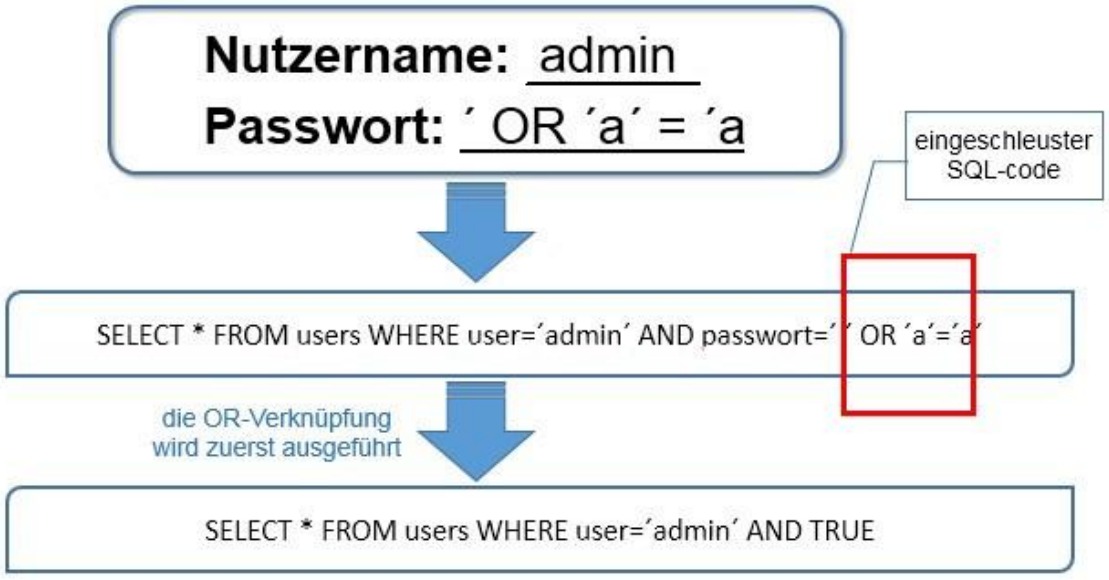

Abbildung 3.3: SQL Injektion Angriffe [Kaco8]

Richtlinien

- **RÜE1**: Bei SaaS-Anwendungen sollten die Eingabedaten, die in einer HTTP-Anfrage übermittelt werden, überprüft werden, ob sie ausführbaren Code mit Schadcode enthalten und ggf. SQL Injektion Attacke verursachen können.

Evtl. entsprechende Maßnahme

- Die Cloud-Anbieter sollten für die korrekte Überprüfung der Eingabedaten ihrer Web-basierten Anwendungen verantwortlich sein, vor allem die Metazeichen des betreffenden Datenbanksystems entsprechend zu maskieren.

- Die Eingabedaten sollten möglicherweise basierend auf die Eigenschaften der Werte weiter überprüft werden. So bestehen Telefonnummer und Postleitzahlen beispielsweise nur aus Ziffern.
- Für verschiedene Programmierungssprachen können auch jeweils verschiedene Methoden verwendet werden, um solche Angriffe abzuwehren. Beispielsweise wird in Java die PreparedStatement-Klasse (JDBC-Technologie) zu diesem Zweck verwendet und die Daten unsicherer Herkunft werden als getrennte Parameter übergeben.
- Viele Datenbankanbieter haben auch verschiedene Produkte gegen den Angriff entwickelt. Ein Beispiel dafür ist die Oracle Database Firewall, die durch effiziente Echtzeit-Analyse des SQL einen Schutz vor SQL Injektion Angriffen gewährt [Fab12].

3.1.12 Referenzen direkt auf interne Objekte

Richtlinien

- **RRI1**: Wenn ein Cloud-Dienst mit Referenzen auf interne Objekte verbindet, z.B. Primär- oder Fremdschlüssel, dürfen diese Referenzen öffentlich nicht zugegriffen werden [TK12].

Evtl. entsprechende Maßnahme

- Wenn eine externe Referenz nicht vermieden werden könnte, darf der Zugriff auf interne Objekte nur kontrolliert (z.B. mithilfe der Zuweisung der Zugriffsrechte zu bestimmten Rollen) erfolgen [TK12].

3.1.13 Einsatz von hochmodernen Hypervisoren

Um die Virtualisierungsumgebung eines Cloud-Dienstes zentral und effizient zu verwalten, wird der virtuelle Maschinen Monitor (Hypervisor) zum Einsatz kommen.

Richtlinien

- **REH1**: Zertifizierte und gehärtete Hypervisoren müssen insbesondere eingesetzt werden, wenn der Cloud-Dienst einen hohen Anspruch auf Vertraulichkeit und Verfügbarkeit hätte [TK12].
- **REH2**: Im Rahmen einer IaaS-Anwendung müssen sowohl die erfolgreichen als auch die abgelehnten Anmeldungen am Hypervisor geloggt werden [TK12].

Evtl. entsprechende Maßnahme

- Oracle VM, Red Hat, Storage-Hypervisor SANsymphony-V können eingesetzt werden.

3.1.14 Regelmäßige Datensicherungen

Die Datensicherung ist immer einer der wichtigsten Aspekte bei der Computersicherheit. Backup für die Public Cloud, Modelle zur Cloud-Subskription, Disaster Recovery auf Cloud-Basis und die Integration von Deduplizierung und Datenschutz – Sie sind nach Ansicht des Security-Anbieters Quantum vier wegweisende Trends für 2013 [Kur13].

Die Datensicherung kann das Hot Backup während des laufenden Betriebs der Datenbank parallel durchgeführt, ohne dass Arbeitsprozesse zu unterbrechen. Im Gegensatz dazu findet ein Cold Backup außerhalb der üblichen Arbeitszeiten statt.

Richtlinien

- **RRS1**: Die Offline Backups sollten in regelmäßigen Zeitabständen erfolgen und vollständig sein.
- **RRS2**: Die Tests der Datenwiederherstellung sollten auch regelmäßig innerhalb eines vordefinierten, angemessenen Zeitraums durchgeführt werden.
- **RRS3**: Wenn der Dienst hohen Anspruch an Verfügbarkeit hat, sollten auch online Backups eingesetzt werden, um die Konsequenz eines Datenausfalls zu entschärfen.
- **RRS4**: Die Back-up Daten sollten redundant und an örtlich separaten Standorten gespeichert werden.
- **RRS5**: Datensicherungen (d.h. deren Umfang, Speicherinternalle, Speicherzeitpunkte und Speicherdauer) müssen nachvollziehbar eingehalten werden [TK12].
- **RRS6**: Nicht mehr benötigte Datensicherungen sollten gelöscht werden.

Evtl. entsprechende Maßnahme

- Für die Private Cloud könnte der Unternehmen die regelmäßigen Datensicherungen selbst durchführen.
- Für die Public Cloud gestaltet sich dies schwieriger, da die Datenkontrollaufgabe nicht mehr gegeben ist, sondern von dem Cloud-Betreiber übernommen wird. Deswegen muss der Cloud-Betreiber sich mit dem Benutzer über die regelmäßige Durchführung der Back-ups absprechen.

3.1.15 Standard-Datenformat

Richtlinien

- **RSF1**: Nicht nur strukturierte sondern auch unstrukturierte Daten wie Back-ups sollten in einem etablierten portablen, standardisierten und nicht proprietären Format bereitgestellt werden. Dadurch wird die mögliche Übertragung von Daten nach Vertragsende zu einem neuen Cloud-Anbieter erleichtert [TK12].

Evtl. entsprechende Maßnahme

- Beispielsweise stehen SQL-Standards ISO/IEC 9075 SQL-92(SQL 2), SQL-99(SQL 3) und SQL-2003 [MNS+07, Info4] zur Verfügung, um Anwendungen so entwickeln zu können, dass sie von ihren verwendeten DBMS unabhängig sind.

3.1.16 Isolierung

Richtlinien

- **RII1**: Isolierung der Dienstinstanzen:
  - Eine Anwendung darf nicht unberechtigt auf die Daten einer anderen Anwendung zugreifen.
  - Für die Durchführung eines Cloud-Diensts muss der Zugriff auf die Daten nur auf diesen Kunden und auf diese Anwendung des Kunden beschränkt sein.
- **RII2**: Isolierung der Kunden-Daten: Der Cloud-Anbieter (insbesondere bei SaaS) muss eine sichere Trennung der Daten der unterschiedlichen Kunden gewährleisten und jeder Mandant bleibt immer für die anderen Mandanten unsichtbar.

Evtl. entsprechende Maßnahme

- Anwendung von Sandboxing
  Eine sicherere Privatsphäre profitiert eventuell davon, wenn die Sandbox- Technologien eingesetzt werden können, indem jede VM bzw. Anwendung eines Kunden als separate Sandbox verwaltet werden und ihre Maßnahmen sich nicht auf die außer andere VMs auswirken. So bleibt der Zugriff isoliert von anderen VMs und fügt keinen Schaden zu.
- Ein tauglicher Grad (siehe Unterkapitel 2.1.2 Sonstige Eigenschaften und Vorteile) der Mandantenfähigkeit sollte basierend auf die Workload des Anwendungsbereiches gewählt werden (z.B. Höhere Stufen passen am besten zu branchenübergreifenden utilitaristischen Workloads wie Sales Force Management [Bla12] und erfordert entsprechend eine höhere Sicherheitsmaßnahmen)

3.1.17 Löschung der Anwendungsdaten und Nutzdaten

Nach Beendigung des Vertragsverhältnisses darf der Cloud-Anbieter weder die Anwendungs- und Kundendaten noch die Nutzdaten aufbewahren.

Richtlinien

- **RLD1**: Für Anwendungsdaten: Nachdem das Vertragsverhältnis zwischen Cloud-Anbieter und ihren Kunden beendet wird, müssen der Cloud-Anbieter alle den Kunden betreffenden Daten von den Datenbanken gelöscht werden. Die maximal mögliche Aufbewahrungsfrist nach dem Vertragsende sollte vorher entweder im Vertrag oder über SLA festgelegt werden [TK12].
- **RLD2**: Für Nutzdaten: Nach dem Vertragsende haben die Kunden die Wahlmöglichkeit, die Nutzdaten bezüglich ihrer Kundenanwendung vollständig zu löschen oder auf die Datenbank des neuen Cloud-Anbieters zu übertragen. Die gelöschten Nutzdaten dürfen nie mehr wiederhergestellt werden [TK12].

Evtl. entsprechende Maßnahme

- Für RLD2: Der Betroffene ist die Art der Daten, die von den Dienstinstanzen für die Cloud-Benutzer in der Cloud-Umgebung verarbeiteten und gespeicherten werden, z.B.alle Back-ups, Log-Daten usw.

3.1.18 Schlüsselaufbewahrung und -management

Da die Bestimmung des Verschlüsselungsalgorithmuses für einen Hacker relativ einfach ist, wird der Schlüssel selbst die Sicherheitsaufgaben von verschlüsselten Informationen übernehmen. Darum ist auch die Aufbewahrung der Schlüssel absolut entscheidend.

Richtlinien

- **RSA1**: Schlüssel sollten außerhalb der sie eingesetzten Datenbanken getrennt aufbewahrt werden.
- **RSA2**: Eine Datei, in der ein Schlüssel abgelegt wird, muss selbst auch verschlüsselt sein.
- **RSA3**: Jeder kryptographische Schlüssel sollte mit einem bestimmten Zweck verbinden. Für die verschiedene Sicherungszwecke müssen jeweils unterschiedliche Schlüssel verwendet werden.

Evtl. entsprechende Maßnahme

- Eine mögliche Maßnahme für RSA2 ist das Password-Locking-Verfahren, indem ein kryptographisch starker Schlüssel mithilfe eines Schlüsselableitungsalgorithmus aus einem eingegebenen Passwort erzeugt werden kann, um die in RSA2 erwähnte Datei zu verschlüsseln [TK12].

3.1.19 Überwachung von Datenbankaktivitäten

Richtlinien

- **RÜD1**: Die Datenbankaktivitäten sollten während des Cloud-Betriebs stets überwacht werden, um die ungewöhnlichen Aktivitäten, die auf Angriffe oder Missbrauchsversuche hinweisen können, rechtzeitig zu erkennen.

Evtl. entsprechende Maßnahme

- Um diese Richtlinie erfolgreich einzuhalten, sollte ein Cloud-Anbieter nicht nur die Datenbank-herstellerspezifischen Built-in Auditmechanismen wie Oracle Audit, SQL Server Audit und IBM DB2-UDB Audit verwenden, sondern auch die herstellerneutralen (Third Party) Database Activity Monitoring Database Activity Monitoring (DAM) Tools auf seinem DBMS einsetzen [jes11].

3.1.20 Konsistenz (engl. Concurreny) nach der Synchronisation

Wie im vorhergehenden Abschnitt 2.3.2 beschrieben, werden die Daten in der Cloud redundant gespeichert, um auf sie bei einem Knoten-Ausfall noch problemlos zuzugreifen. Weil Solche Datenkopien in der Cloud auf verschiedenen Knoten gespeichert werden, die sich oft über große geographische Distanzen befinden, bringt das Synchronisationsproblem mit sich.

Richtlinien

- **RKS1**: Nach einer Datenveränderung sollten alle betroffenen Knoten, die eine Kopie von diesen Daten gespeichert haben, ihre Datenbank sofort synchronisieren.
- **RKS2**: Falls ein Knoten seine Datensynchronisation noch nicht fertiggestellt hat, ist ein Zugriff auf diese synchronisierten Daten des Knotens verboten.

Evtl. entsprechende Maßnahme

- Die betroffene Zeile, Spalten oder die ganze Datenbank werden während der Synchronisation gesperrt. Die Realisierung einer Sperrung dieses kritischen Abschnitts könnte durch den Einsatz von Mutex-Verfahren(engl. Mutual Exclusion) [AKH03] erfolgen, das häufig für die Koordination von nebenläufigen und auf gemeinsam genutzte Daten zugreifenden Prozesse/Threads verwendet werden.

3.2 Verschlüsselungstechnologien

Für die relationalen Datenbankprodukte verschiedener Softwareanbieter gibt es jeweils unterschiedliche Verschlüsselungsverfahren. Die Auswahl einer fortschrittlichen Verschlüsselungstechnologie ist die Basis für die Realisierung der meisten Sicherheitsrichtlinien. In diesem Unterkapitel werden die Verschlüsselungsverfahren von Oracle, MySQL und DB2 erläutert.

3.2.1 Oracle: Transparente Datenverschlüsselung

In Oracle Datenbank 10g wurde Transparent Data Encryption (TDE) zum ersten Mal eingesetzt, um Tabellenspalten verschlüsseln zu können. Allerdings gibt es dabei noch eine ganze Reihe von Einschränkungen für die betroffene Datentypen, Indizes, usw. Außerdem ist es auch problematisch bei der Verschlüsselung der Fremdschlüsselspalten [HWF].

In Oracle Datenbank 11g (z.B. Oracle Database Cloud Service nutzt Oracle DB 11g Release 2 [Koo12]) kann man den ganzen Tabellenraum ohne Einschränkungen verschlüsseln, damit es eine sicherere Performanz bieten kann. Jedoch kann ein Tabellenraum nicht nachträglich verschlüsselt werden, sondern immer nur als verschlüsselt angelegt werden [HWF]. In diesem Fall müsste die Verschiebung vorhandener Daten in einen verschlüsselten Tabellenraum mit einem CREATE TABLE AS SELECT oder mit einem Export/Import [HWF] durchgeführt werden.

Schlüsselverwaltung: Zweistufige Verfahren: geeignet sowohl für die Spaltenverschlüsselung als auch für die Tabellenraumverschlüsselung [HWF, Brö09] (siehe Abb. 3.4).

1) Die Verschlüsselungen und Entschlüsselungen der dazugehörigen Daten jeder Tabelle und jedes Tabellenraums werden mit einem eigenen Schlüssel durchgeführt, der von Oracle automatisch erzeugt und innerhalb der Tabelle bzw. des Tabellenraums gespeichert wird.
2) Danach wird ein neuer Schlüssel, der sogenannte Master Key, von Oracle erzeugt, mit dem die Schlüssel der Tabellen und Tabellenraums ver- und entschlüsselt werden können. Und dieser Master Key wird außerhalb der Datenbank, in einem gesicherten Bereich abgelegt.

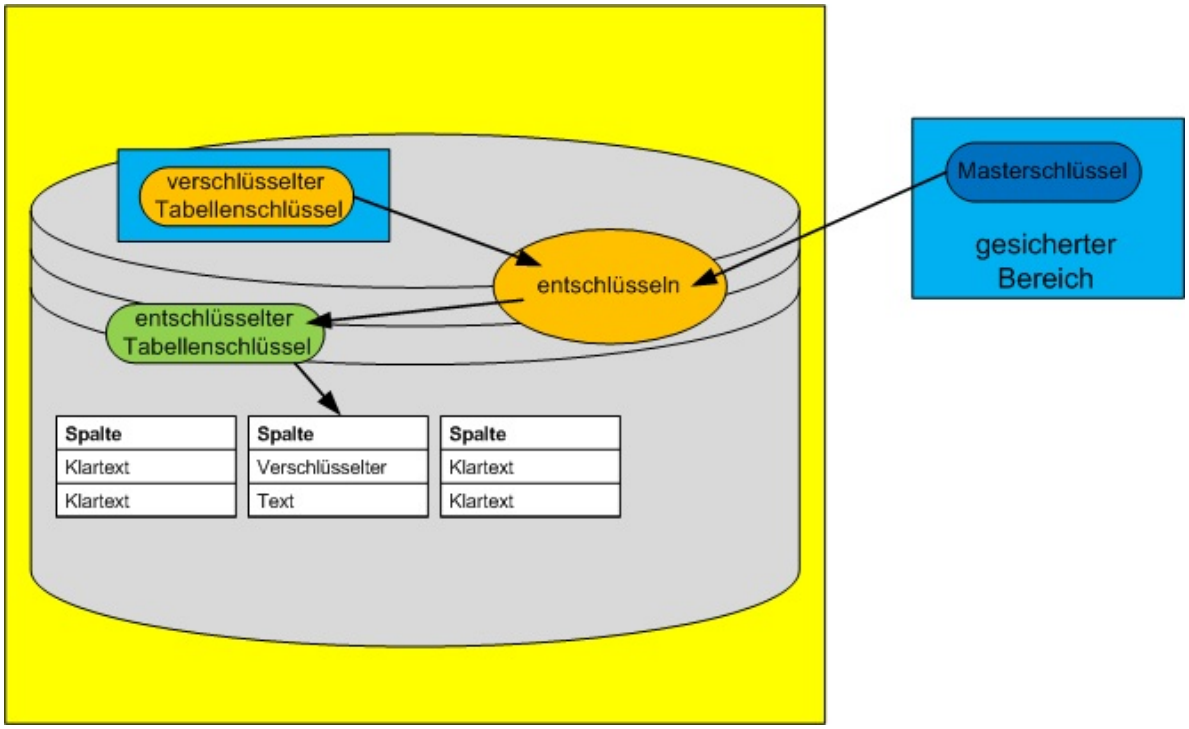

Abbildung 3.4: Transparente Datenverschlüsselung [Brö09]

3.2.2 MySQL

Die Verschlüsselung bei MySQL ist im Gegensatz zur Oracle-Lösung nicht transparent.

Bei jedem Datenbankzugriff, der auf eine verschlüsselte Datenbankspalte zugreift, werden die Befehle ENCRYPT() bzw. DECRYPT() benutzt.

Bei der ENCRYPT-Funktion (verschlüsselt einen String und gibt einen Binär-String zurück [MyS13a]) werden beide, der zu verschlüsselnde Klartext und der Schlüssel übergeben, während bei DECRYPT-Funktion (entschlüsselt den verschlüsselten String und gibt den Original-String zurück [MyS13a]) der verschlüsselte Datensatz und der Schlüssel übergeben werden. Und was wichtiger ist, dass die DECRYPT-Funktion nur funktioniert, wenn die Verbindungen zwischen MySQL-Clients und dem Server unter Verwendung des SSL-Protokolls (Secure Sockets Layer) konfiguriert wurde [MyS13a, MyS13b].

Für die Verschlüsselungsalgorithmen werden normalerweise AES und Triple-DES verwendet. Bei AES wird meistens ein Schlüssel mit 128-bit Länge benutzt, da diese Schlüssellänge für eine schnellere Ver- und Entschlüsselung besorgt und ausreichend Sicherheit für die meisten Anwendungsfälle bieten kann.

Der folgende Beispielcode zeigt, dass der Schlüssel bei MySQL im Gegensatz zur Oracle TDE nicht automatisch von der Datenbank erzeugt wird , sondern muss der Administrator ihn explizit angeben.

```
INSERT INTO Studenten
(Matrikelnummer, Vorname, Nachname)
VALUES(
'2433510',
AES_ENCRYPT('Max', 'meinschluessel')
AES_ENCRYPT('Mustermann', 'meinschluessel')
);

SELECT
Matrikelnummer,
AES_DECRYPT(Vorname, 'meinschluessel')
AES_DECRYPT(Nachname, 'meinschluessel')
FROM Studenten;
```

3.2.3 IBM DB2

DB2 ist eine kommerzielle relationale Datenbank, die von IBM entwickelt ist. Gleich wie Oracle bietet IBM ihren Datenbanken auch viele transparente Verschlüsselungsvorgänge. Mit der Verschlüsselungshardware von IBM System Z werden sensible Daten auf DB2-Zeilenebene geschützt [Cor13]. Als Beispiel wird folgendermaßen das IBM Werkzeug InfoSphere Guardium Data Encryption vorgestellt, bei dem AES 128, 192 und 256 Bit Schlüssellänge verwendet werden können.

Der Verschlüsselungsvorgang läuft für DB2 (siehe Abb.3.5) wie folgend ab [Cor13]:

1. Das Anwendungsprogramm übergibt eine Zeile an DB2.
2. DB2 lädt die Verschlüsselungsmethode (als DB2 edit procedure exit routine genannt), die durch einen entsprechenden SQL CREATE TABLEBE Befehl spezifiziert werden. Danach überträgt DB2 die Zeile auf diese DB2-Routine.
3. Zudem ruft die Edit-Routine das Integrated Kryptographisch Service Facility (ICSF) Service auf, um die Ziele zu verschlüsseln.
4. Nach der erfolgreichen Verschlüsselung gibt die ICSF Service der DB2-Routine die verschlüsselte Zeile zurück.
5. Diese verschlüsselte Zeile wird von DB2 in einer Tabelle der Datenbank gespeichert.

Wie in der Abbildung 3.6 dargestellt ist, wird der Entschlüsselungsvorgang fast umgekehrt vorgegangen [Cor13]:

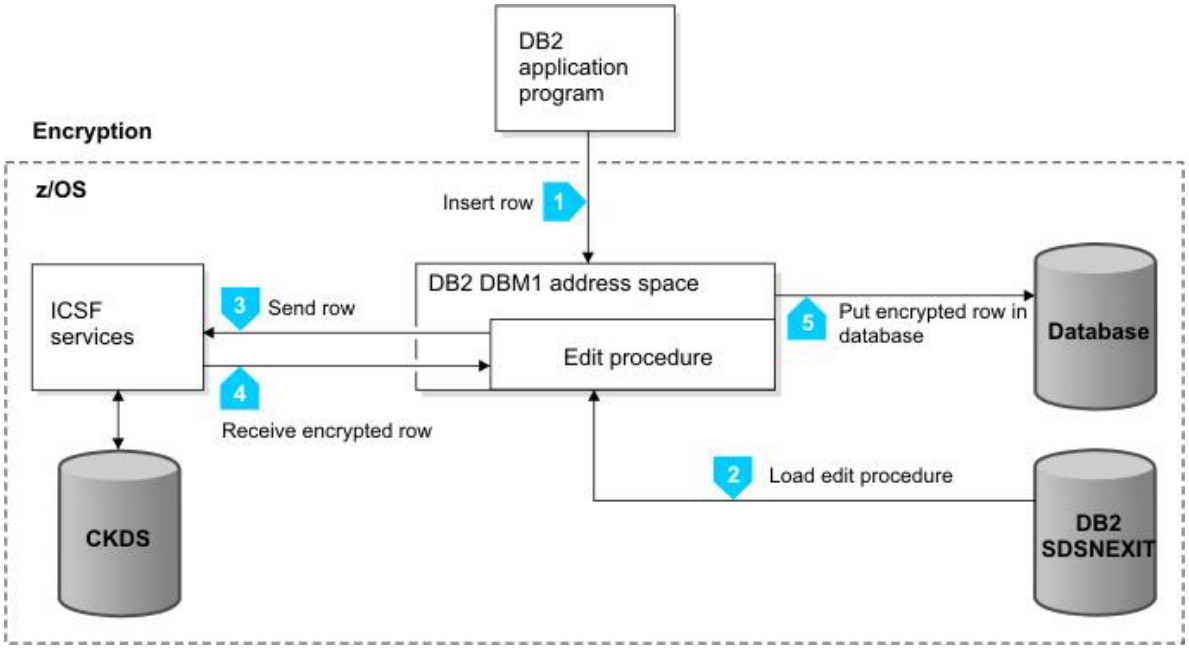

Abbildung 3.5: Datenverschlüsselung in DB2 [Cor13]

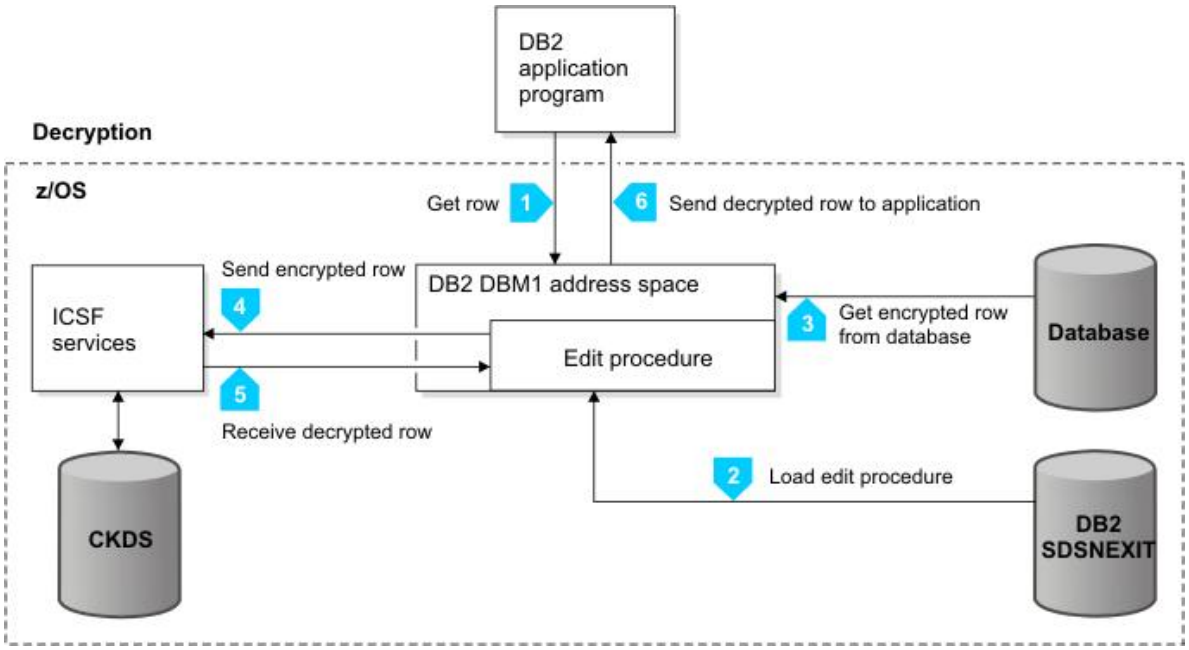

Abbildung 3.6: Datenentschlüsselung in DB2 [Cor13]

1. Das Anwendungsprogramm sendet eine Anfrage an DB2, um die entsprechende Zeile/Daten zu erhalten.
2. DB2 lädt die Verschlüsselungsmethode DB2-Routine.
3. DB2 holt die verschlüsselte Zeile von der Tabelle ab, bei der die Zeile vorher gespeichert worden ist.
4. Die DB2-Routine wird das ICSF Service wieder aufrufen, um die verschlüsselte Ziele zu entschlüsseln.
5. Nach der erfolgreichen Entschlüsselung gibt die ICSF Service der DB2-Routine die Zeile in Klartext zurück.
6. Die entschlüsselte Zeile wird von DB2 an der Anwendung zurückgeben.

Zwar sind die meistens DB2 Verschlüsselungsvorgänge transparent und flexibel, trotzdem hat InfoSphere Guardium Data Encryption noch Einschränkungen. Zum Beispiel kann weder der Index einer Tabelle noch Tabellen mit ROWIDs oder LOBs verschlüsselt werden. Darüber hinaus kann eine DB2 Tabelle ausschließlich mit einem Bearbeitungsverfahren verschlüsselt werden [Cor13].

4 Taxonomie nach verschiedenen Kriterien

Im diesen Kapitel werden Taxonomien anhand der in Kapitel 3 dargestellten Richtlinien abgeleitet. Jede Taxonomie klassifiziert die Sicherheitsrichtlinien hinsichtlich einer bestimmten Perspektive. Die folgende Tabelle gibt zuerst einen Überblick über alle Taxonomien jeweils mit ihrem Klassifikationskatalog. Danach werden bei jedem nachfolgenden Unterkapitel eine konkrete Taxonomie und die nach dieser Taxonomie klassifizierte Richtlinientabelle erläutert.

4.1 Service-Modelle der Cloud Architekturen	Paas
	IaaS
	SaaS
4.2 Deployment-Modelle	Private Cloud
	Public Cloud
	Community Cloud
	Hybrid Cloud
4.3 Zeitpunkt der Durchsetzung einer Sicherheitsrichtlinie	By Design
	Deployment
	Laufzeit
4.4 Arten des RDBMS	Transaktionale DB
	Analytische DB
4.5 Granularität	Spalte/Zeile
	Eine Tabelle
	Verteile DB
	Ganze DBMS
4.6 Durchführungstypen	Automatisch
	Manuell

Tabelle 4.1: Taxonomie Übersicht

4.1 Service-Modelle der Cloud Architekturen

Es entsteht gemäß den drei Servicemodellen verschiedene Datenmanagementverfahren, bei denen der Dienst-Anbieter unterschiedliche Granularität vom Kontrolleinfluss für Datenbanksicherheit anbieten kann. Deswegen können auch die Sicherheitsrichtlinien nach diesen

drei Servicemodellen klassifiziert werden. Dadurch können die aufgrund verschiedenen Servicemodellen entwickelnden Dienst-Anbieter die geeigneteren Richtlinien einhalten.

Richtlinien		PaaS	IaaS	SaaS
3.1.1 Transparenz	RTP1	Muss	Muss	Muss
3.1.2 Datenvermeidbarkeit	RDM1	Muss	Muss	Muss
3.1.3 Datensparsamkeit	RDS1	Muss	Muss	Muss
	RDS2	Muss	Muss	Muss
3.1.4 Zweckbindung	RZB1	Muss	Muss	Muss
	RZB2	Muss	Muss	Muss
3.1.5 Vertraulichkeit	RVT1	Muss	Muss	Muss
3.1.6 Integrität	RIG1	Muss	Muss	Muss
	RIG2	Muss	Muss	Muss
	RIG3	Muss	Muss	Muss
3.1.7 Verfügbarkeit	RVF1	Muss	Muss	Muss
	RVF2	Muss	Muss	Muss
	RVF3	Muss	Muss	Muss
3.1.8 Authentizität	RAT1	Muss	Muss	Muss
	RAT2	Muss	Muss	Muss
3.1.9 Pseudonymität	RPD1	Muss	Muss	Muss
	RPD2	Muss	Muss	Muss
	RPD3	Muss	Muss	Muss
3.1.10 Angemessenes Datenschutzniveau	RSN1	Muss	Muss	Muss
3.1.11 Überprüfung von Benutzereingaben	RÜE1	Sollte	Sollte	Muss
3.1.12 Referenzen direkt auf interne Objekte	RRI1	Muss	Muss	Muss
3.1.13 Einsatz von hochmodernen Hypervisoren	REH1	Sollte	Muss	Sollte
	REH2	Sollte	Muss	Sollte
3.1.14 Regelmäßige Datensicherungen	RRS1	Muss	Muss	Muss
	RRS2	Muss	Muss	Muss
	RRS3	Muss	Muss	Muss
	RRS4	Muss	Muss	Muss
	RRS5	Muss	Muss	Muss
	RRS6	Muss	Muss	Muss
3.1.15 Standard-Datenformat	RSF1	Sollte	Sollte	Sollte
3.1.16 Isolierung	RII1	Muss	Muss	Muss
	RII2	Muss	Muss	Muss
3.1.17 Löschung der Anwendungsdaten und Nutzdaten	RLD1	Muss	Muss	Muss
	RLD2	Muss	Muss	Muss
3.1.18 Schlüsselaufbewahrung und -management	RSA1	Muss	Muss	Muss

	RSA2	Muss	Muss	Muss
	RSA3	Muss	Muss	Muss
3.1.19 Überwachung von Datenbankaktivitäten	RÜD1	Muss	Muss	Muss
3.1.20 Konsistenz (engl. Concurreny) nach der Synchronisation	RKS1	Muss	Muss	Muss
	RKS2	Muss	Muss	Muss

Tabelle 4.2: Taxonomie: Service Modelle

4.2 Deployment-Modelle

Die Nutzergruppe nach den verschiedenen Cloud-Deployment Modellen erweist sich auch unterschiedliches Niveau von Sicherheitsbedürfnissen. Typischerweise scheinen die Sicherheitsbedrohungen bei einer Private Cloud nicht so dringend wie bei einer Public Cloud, da der Kunde und der Cloud-Betreiber in diesem Fall derselben Organisation angehören, und die ganze Cloud gegen äußere Gefahren abgesichert ist. Aus diesem Grund sind die Richtlinien hinsichtlich der verschiedenen Deployment Modelle auch klassifizierbar, damit die mit verschiedenartigen Zwecke verbundenen Clouds die entsprechenden Richtlinien beachten.

Richtlinien		Privat Cloud	Public Cloud	Community Cloud	Hybrid Cloud
3.1.1 Transparenz	RTP1	Muss	Muss	Muss	Muss
3.1.2 Datenvermeidbarkeit	RDM1	Sollte	Muss	Muss	Muss
3.1.3 Datensparsamkeit	RDS1	Sollte	Muss	Muss	Muss
	RDS2	Muss	Muss	Muss	Muss
3.1.4 Zweckbindung	RZB1	Sollte	Muss	Muss	Muss
	RZB2	Muss	Muss	Muss	Muss
3.1.5 Vertraulichkeit	RVT1	Muss	Muss	Muss	Muss
3.1.6 Integrität	RIG1	Muss	Muss	Muss	Muss
	RIG2	Muss	Muss	Muss	Muss
	RIG3	Muss	Muss	Muss	Muss
3.1.7 Verfügbarkeit	RVF1	Muss	Muss	Muss	Muss
	RVF2	Muss	Muss	Muss	Muss
	RVF3	Muss	Muss	Muss	Muss
3.1.8 Authentizität	RAT1	Muss	Muss	Muss	Muss
	RAT2	Muss	Muss	Muss	Muss

3.1.9 Pseudonymität	RPD1	Sollte	Muss	Muss	Muss
	RPD2	Muss	Muss	Muss	Muss
	RPD3	Muss	Muss	Muss	Muss
3.1.10 Angemessenes Datenschutzniveau	RSN1	Muss	Muss	Muss	Muss
3.1.11 Überprüfung von Benutzereingaben	RÜE1	Muss	Muss	Muss	Muss
3.1.12 Referenzen direkt auf interne Objekte	RRI1	Muss	Muss	Muss	Muss
3.1.13 Einsatz von hochmodernen Hypervisoren	REH1	Muss	Muss	Muss	Muss
	REH2	Muss	Muss	Muss	Muss
3.1.14 Regelmäßige Datensicherungen	RRS1	Muss	Muss	Muss	Muss
	RRS2	Sollte	Sollte	Sollte	Sollte
	RRS3	Muss	Muss	Muss	Muss
	RRS4	Sollte	Sollte	Sollte	Sollte
	RRS5	Muss	Muss	Muss	Muss
	RRS6	Muss	Muss	Muss	Muss
3.1.15 Standard-Datenformat	RSF1	sollte	Muss	sollte	Muss
3.1.16 Isolierung	RII1	Muss	Muss	Muss	Muss
	RII2	Muss	Muss	Muss	Muss
3.1.17 Löschung der Anwendungsdaten und Nutzdaten	RLD1	k.A.	Muss	Muss	Muss
	RLD2	Sollte	Muss	Muss	Muss
3.1.18 Schlüsselaufbewahrung und -management	RSA1	Muss	Muss	Muss	Muss
	RSA2	Muss	Muss	Muss	Muss
	RSA3	Muss	Muss	Muss	Muss
3.1.19 Überwachung von Datenbankaktivitäten	RÜD1	Sollte	Muss	Muss	Muss
3.1.20 Konsistenz (engl. Concurreny) nach der Synchronisation	RKS1	Muss	Muss	Muss	Muss
	RKS2	Muss	Muss	Muss	Muss

Tabelle 4.3: Taxonomie: Deployment Modelle

4.3 Zeitpunkt der Durchsetzung einer Sicherheitsrichtlinie

Da jede Richtlinie auf die verschiedene Phase des Lebenszyklus einer Cloud wirkt, hängt dieser Taxonomie von den durchgesetzten Zeitpunkten einer Richtlinie ab. Hierbei werden die Zeitpunkte der Durchsetzungen in drei aufeinanderfolgende Phasen gegliedert.

- **By Design:** Während des Designs des Cloud-Dienstes muss diese Richtlinie berücksichtigt werden, und deren Einhaltung soll ständig geprüft werden. In der Zukunft könnte der Einsatz eines Cloud-Dienstes gegebenenfalls nur dann möglich sein, wenn es alle Zertifikate für By Design Richtlinien erfolgreich erworben hat.
- **Deployment:** Im Rahmen des Deployments fokussiert sich eine Richtlinie auf die Einstellung eines Cloud-Dienstes vor seinem Betrieb.
- **Laufzeit:** Die Funktion derjenigen Richtlinien ist während der ganzen Laufzeit eines Cloud-Dienstes zu überwachen und zu kontrollieren.

Richtlinien		By Design	Depolyment	Laufzeit
3.1.1 Transparenz	RTP1	✓		✓
3.1.2 Datenvermeidbarkeit	RDM1	✓		✓
3.1.3 Datensparsamkeit	RDS1	✓		
	RDS2		✓	
3.1.4 Zweckbindung	RZB1			✓
	RZB2			✓
3.1.5 Vertraulichkeit	RVT1	✓		✓
3.1.6 Integrität	RIG1			✓
	RIG2	✓		✓
	RIG3	✓		✓
3.1.7 Verfügbarkeit	RVF1	✓		✓
	RVF2	✓		✓
	RVF3	✓		✓
3.1.8 Authentizität	RAT1	✓		✓
	RAT2		✓	✓
3.1.9 Pseudonymität	RPD1	✓	✓	
	RPD2	✓		✓
	RPD3	✓		✓
3.1.10 Angemessenes Datenschutzniveau	RSN1	✓	✓	
3.1.11 Überprüfung von Benutzereingaben	RÜE1			✓
3.1.12 Referenzen direkt auf interne Objekte	RRI1	✓		
3.1.13 Einsatz von hochmodernen Hypervisoren	REH1	✓		✓
	REH2			✓

3.1.14 Regelmäßige Datensicherungen	RRS1	✓	✓	✓
	RRS2	✓		✓
	RRS3	✓	✓	✓
	RRS4	✓	✓	
	RRS5	✓		✓
	RRS6	✓	✓	
3.1.15 Standard-Datenformat	RSF1	✓	✓	
3.1.16 Isolierung	RII1	✓	✓	
	RII2	✓	✓	
3.1.17 Löschung der Anwendungsdaten und Nutzdaten	RLD1		✓	✓
	RLD2		✓	✓
3.1.18 Schlüsselaufbewahrung und -management	RSA1	✓		
	RSA2	✓		✓
	RSA3	✓		✓
3.1.19 Überwachung von Datenbankaktivitäten	RÜD1			✓
3.1.20 Konsistenz (engl. Concurreny) nach der Synchronisation	RKS1	✓		
	RKS2	✓		

Tabelle 4.4: Taxonomie: Zeitpunkt der Durchsetzung

4.4 Arten des RDBMS

Die Unternehmen, die umfangreich strukturierten Daten für die Ausführung von Geschäftsprozessen verarbeiten müssen, besitzt ein Großteil der Kundenressourcen von Cloud-Diensten. Solche Daten werden einerseits für die Bearbeitung von Geschäftsvorfällen (als Online Transaction Processing (OLTP) genannt) benötigt, andererseits für Auswertungen verwendet, die die zukünftigen Entscheidungen unterstützen können (Online Analytical Processing (OLAP)) [Boe10]. Aus dieser Sicht können die geschäftsorientierten RDBMS in analytische Datenbanken und transaktionale Datenbanken unterteilt werden. Der Unterschied liegt darin, dass die Datenverarbeitung in einer transaktionalen Datenbank die ACID Eigenschaften strikt befolgt, während die Daten in einer analytischen Datenbank nicht so häufig geändert werden, sondern meistens nur gelesen und miteinander verglichen werden.

Wenn man über die jeweiligen Merkmale dieser zwei Arten des RDBMS diskutiert, kann man sich vorstellen, dass jede Richtlinien ihren Nachdruck auf verschiedene Arten des RDBMS legen. Beispielsweise denken wir an ein analytisches RDBMS, bei dem die meisten durchgeführte Operation „Daten Lesen“ ist. Dabei spielen die Richtlinien Vertraulichkeit und Zweckverbindung eine wichtige Rolle, um den unberechtigten Lesezugriff zu verhindern und das Lesen zu prüfen, ob es nur für die vereinbarten Zwecke genutzt wird. Im Gegensatz dazu sind die Richtlinien z.B. die Verfügbarkeit und Integrität wichtiger für eine transaktionsorientierte Datenbank.

Richtlinien		Transaktionale DB	Analytische DB
3.1.1 Transparenz	RTP1	Muss	Muss
3.1.2 Datenvermeidbarkeit	RDM1	Muss	Muss
3.1.3 Datensparsamkeit	RDS1	Muss	Sollte
	RDS2	Muss	Sollte
3.1.4 Zweckbindung	RZB1	Muss	Muss
	RZB2	Muss	Muss
3.1.5 Vertraulichkeit	RVT1	Muss	Muss
3.1.6 Integrität	RIG1	Muss	Muss
	RIG2	Muss	Sollte
	RIG3	Muss	Sollte
3.1.7 Verfügbarkeit	RVF1	Muss	Muss
	RVF2	Muss	Muss
	RVF3	Muss	Sollte
3.1.8 Authentizität	RAT1	Muss	Muss
	RAT2	Muss	Sollte
3.1.9 Pseudonymität	RPD1	Muss	Muss
	RPD2	Muss	Muss
	RPD3	Muss	Muss
3.1.10 Angemessenes Datenschutzniveau	RSN1	Muss	Muss
3.1.11 Überprüfung von Benutzereingaben	RÜE1	Muss	Muss
3.1.12 Referenzen direkt auf interne Objekte	RRI1	Muss	Muss
3.1.13 Einsatz von hochmodernen Hypervisoren	REH1	Muss	Muss
	REH2	Muss	Muss
3.1.14 Regelmäßige Datensicherungen	RRS1	Muss	Muss
	RRS2	Muss	Muss
	RRS3	Muss	Sollte
	RRS4	Muss	Sollte
	RRS5	Muss	Muss
	RRS6	Muss	Muss

3.1.15 Standard-Datenformat	RSF1	Sollte	Sollte
3.1.16 Isolierung	RII1	Muss	Muss
	RII2	Muss	Muss
3.1.17 Löschung der Anwendungsdaten und Nutzdaten	RLD1	Muss	Muss
	RLD2	Muss	Muss
3.1.18 Schlüsselaufbewahrung und -management	RSA1	Muss	Muss
	RSA2	Muss	Muss
	RSA3	Muss	Muss
3.1.19 Überwachung von Datenbankaktivitäten	RÜD1	Muss	Sollte
3.1.20 Konsistenz (engl. Concurreny) nach der Synchronisation	RKS1	Muss	Muss
	RKS2	Muss	Muss

Tabelle 4.5: Taxonomie: Arten des DBMS

4.5 Granularität

Hinsichtlich der Perspektive Granularität können die Richtlinien nach der Größe ihrer Geltungsbereiche eingeordnet werden. Je nach der Größe kann eine Richtlinie auf Spalten-/Zeilenebene, Tabellenebene, verteilte Datenbankebene oder das ganze RDBMS einwirken.

Richtlinien		Spalte / Zeile	Eine Tabelle	Verteile DB	Ganze DBMS
3.1.1 Transparenz	RTP1			✓	✓
3.1.2 Datenvermeidbarkeit	RDM1	✓	✓	✓	✓
3.1.3 Datensparsamkeit	RDS1				✓
	RDS2				✓
3.1.4 Zweckbindung	RZB1				✓
	RZB2				✓
3.1.5 Vertraulichkeit	RVT1		✓		
3.1.6 Integrität	RIG1		✓		
	RIG2		✓		✓
	RIG3		✓		
3.1.7 Verfügbarkeit	RVF1				✓

	RVF2		✓		
	RVF3				✓
3.1.8 Authentizität	RAT1		✓		
	RAT2				✓
3.1.9 Pseudonymität	RPD1		✓	✓	
	RPD2		✓	✓	
	RPD3		✓	✓	
3.1.10 Angemessenes Datenschutzniveau	RSN1			✓	
3.1.11 Überprüfung von Benutzereingaben	RÜE1		✓		
3.1.12 Referenzen direkt auf interne Objekte	RRI1		✓		
3.1.13 Einsatz von hochmodernen Hypervisoren	REH1			✓	✓
	REH2			✓	✓
3.1.14 Regelmäßige Datensicherungen	RRS1				✓
	RRS2				✓
	RRS3				✓
	RRS4			✓	
	RRS5				✓
	RRS6				✓
3.1.15 Standard-Datenformat	RSF1	✓	✓	✓	
3.1.16 Isolierung	RII1				✓
	RII2				✓
3.1.17 Löschung der Anwendungsdaten und Nutzdaten	RLD1				✓
	RLD2				✓
3.1.18 Schlüsselaufbewahrung und -management	RSA1		✓	✓	
	RSA2			✓	
	RSA3			✓	
3.1.19 Überwachung von Datenbankaktivitäten	RÜD1			✓	✓
3.1.20 Konsistenz (engl. Concurreny) nach der Synchronisation	RKS1			✓	✓
	RKS2			✓	✓

Tabelle 4.6: Taxonomie: Granularität

4.6 Durchführungstypen

Richtlinien können entweder während des Betriebs automatisch durchgeführt werden, oder je nach Wunsch manuell fokussiert werden. Ein gemischtes Verfahren ist auch möglich. Ein Beispiel dafür ist die Richtlinie Datensparsamkeit RDS1, dass die nicht mehr zur Erreichung eines rechtmäßigen Zwecks benötigten Daten nach einer bestimmten Frist automatisch gelöscht werden sollten. Und dieser Zeitraum kann vor dem Betrieb je nach dem Datentyp manuell eingestellt werden.

Richtlinien		Automatisch	Manuell
3.1.1 Transparenz	RTP1	✓	
3.1.2 Datenvermeidbarkeit	RDM1	✓	
3.1.3 Datensparsamkeit	RDS1	✓	✓
	RDS2		✓
3.1.4 Zweckbindung	RZB1	✓	✓
	RZB2	✓	
3.1.5 Vertraulichkeit	RVT1	✓	
3.1.6 Integrität	RIG1	✓	
	RIG2	✓	
	RIG3	✓	
3.1.7 Verfügbarkeit	RVF1	✓	
	RVF2	✓	
	RVF3	✓	
3.1.8 Authentizität	RAT1	✓	
	RAT2	✓	
3.1.9 Pseudonymität	RPD1	✓	
	RPD2	✓	
	RPD3	✓	
3.1.10 Angemessenes Datenschutzniveau	RSN1	✓	✓
3.1.11 Überprüfung von Benutzereingaben	RÜE1	✓	✓
3.1.12 Referenzen direkt auf interne Objekte	RRI1	✓	
3.1.13 Einsatz von hochmodernen Hypervisoren	REH1	✓	
	REH2		✓
3.1.14 Regelmäßige Datensicherungen	RRS1	✓	✓
	RRS2		✓
	RRS3	✓	✓
	RRS4	✓	
	RRS5	✓	
	RRS6		✓

3.1.15 Standard-Datenformat	RSF1	✓	✓
3.1.16 Isolierung	RII1	✓	
	RII2	✓	
3.1.17 Löschung der Anwendungsdaten und Nutzdaten	RLD1	✓	✓
	RLD2	✓	✓
3.1.18 Schlüsselaufbewahrung und -management	RSA1		✓
	RSA2		✓
	RSA3		✓
3.1.19 Überwachung von Datenbankaktivitäten	RÜD1	✓	✓
3.1.20 Konsistenz (engl. Concurreny) nach der Synchronisation	RKS1	✓	
	RKS2	✓	

Tabelle 4.7: Taxonomie: Durchführungstypen

5 Implementierung

In folgendem Kapitel wird die Umsetzung der Sicherheitsrichtlinien mit der TOSCA Sprache 3 durchgeführt.

5.1 Auswahl einer Policy Sprache

Um die als Text dargestellten Richtlinienspezifikation umzusetzen, können entweder eine existierenden Policy Sprache oder in TOSCA selbst definierte Policy Sprache eingesetzt werden (können auch gemischt verwendet).

Ein prominentes Beispiel der Policy Sprache ist die im Zuge der WebService-Spezifikationen des W3C entworfene Sprache WS-Policy, die in [SR12] von Renner et. al die höchste Noten zur Auswertung verschiedener Policy Sprachen bekam (siehe Abb. 5.1), um Dienst-Anbietern die Möglichkeit zu beschaffen, ihre Dienste an Qualitäts- und Sicherheitsrichtlinien zu heften. Hinsichtlich der Ergebnisse der Fachstudie von Renner et. Al [SR12] erfüllt allerdings keine Sprache die Anforderungen des Projektes CloudCycle vollständig, da die mitgebrachten Mittel der Sprache meistens für domänenspezifische Beschreibungsaufgaben nicht ausreichen, und die Sprache muss in diesem Fall noch erweitert werden (z.B. bei Policy Sprache KAos wäre es durch ein entsprechende Ontologie möglich).

Im Vergleich dazu ist die Umsetzung mit der TOSCA Sprache, die basierend auf XML Schema 1.0 und einen Extensionsmechanismus für die Erweiterung der Anbieter- und Domain-spezifische Definition anbietet [Com13b] relativ einfach. Darüber hinaus ist es mit der TOSCA Spezifikation auch möglich, den Aufbau und die Verwaltung eines Cloud-Dienstes portabel zu beschreiben [SR12]. Deswegen wird die TOSCA Sprache für die folgende Beschreibung ausgewählt.

5.2 XML Schema für die Richtliniendefinition

In diesem Unterkapitel wird das XML Schema Dokument präsentiert, welches die Datentypen und Einschränkungen der Property-Elemente von der jeweilige Richtlinie vordefiniert hat. Die Property-Elemente werden danach im entsprechenden Policy Typ Dokument referenziert.

Das Schema Dokument ist nicht kompakt geschrieben, beispielsweise steht jetzt bei vielen Properties wie bei den Transparency Properties nur ein Element darin und man bräuchte tatsächlich keinen komplexen Typ für die Typdefinition, sondern könnte dieses Element

Platz:	Sprache:	Punkte:
1.	WS-Policy	9,38
2.	Rei	8,83
3.	XACML	8,12
4.	PERMIS	7,57
5.	SAML	7,40
6.	RuleML	5,26
7.	P3P	4,80
8.	UDDI	4,61
9.	SecPAL	4,61
10.	ODRL	4,48
11.	IBM EPAL	4,03

Abbildung 5.1: Endergebnis der Auswertung mit gleich-gewichteten Kriterien [SR12]

direkt als simplen Typ definieren. Allerdings könnte es sehr wohl möglich sein, dass man später z.B. die Richtlinien Transparenz noch vervollständigen und erweitern wird, indem mehre Elemente bei der Property-Definition hinzugefügt werden, um ein Property besser zu begrenzen. Im diesen Fall reicht der simple Typ nicht aus und wäre es für die weitere Arbeit besser, wenn schon der Container „TransparenceProperties“ da ist. Deswegen habe ich das Schema so aufgebaut, um Erweiterbarkeit für die zukünftige Verbesserungsarbeit zu unterstützen.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
        targetNamespace="http://www.example.com/tosca/Types/PolicyProperties"
        xmlns="http://www.example.com/tosca/Types/PolicyProperties">

        <xs:element name="TransparencyProperties"
            type="tTransparencyProperties" />

        <xs:element name="DataAvoidabilityProperties"
            type="tDataAvoidabilityProperties" />

        <xs:element name="DataEconomyProperties" type="tDataEconomyProperties"
            />

        <xs:element name="LimitedUseProperties" type="tLimitedUseProperties" />
```

```
<xs:element name="ConfidentialityProperties"
    type="tConfidentialityProperties" />

<xs:element name="IntegrityProperties" type="tIntegrityProperties" />

<xs:element name="AvailabilityProperties"
    type="tAvailabilityProperties" />

<xs:element name="AuthenticityProperties"
    type="tAuthenticityProperties" />

<xs:element name="PseudonymityProperties"
    type="tPseudonymityProperties" />

<xs:element name="DataProtectionLevelProperties"
    type="tDataProtectionLevelProperties" />

<xs:element name="InputValidationProperties"
    type="tInputValidationProperties" />

<xs:element name="ReferenceInternalObjectProperties"
    type="tReferenceInternalObjectProperties" />

<xs:element name="VMMonitorProperties" type="tVMMonitorProperties" />

<xs:element name="BackupProperties" type="tBackupProperties" />

<xs:element name="StandardDataFormatProperties"
    type="tStandardDataFormatProperties" />

<xs:element name="IsolationProperties" type="tIsolationProperties" />

<xs:element name="DataDeletionProperties"
    type="tDataDeletionProperties" />

<xs:element name="KeyStorageProperties" type="tKeyStorageProperties" />

<xs:element name="DatabaseMonitorProperties"
    type="tDatabaseMonitorProperties" />

<xs:element name="ConcurrencyProperties" type="tConcurrencyProperties"
    />

<xs:complexType name="tTransparencyProperties">
       <xs:sequence>
```

```
                <xs:element name="LogDataRequest" type="xs:boolean"
                    fixed="true" />
        </xs:sequence>
</xs:complexType>

<xs:complexType name="tDataAvoidabilityProperties">
        <xs:sequence>
                <xs:element name="EncryptionRequest" type="xs:boolean"
                        fixed="true" />
        </xs:sequence>
</xs:complexType>

<xs:complexType name="tDataEconomyProperties">
        <xs:sequence>
                <xs:element name="TimeWindow" type="xs:duration" />
                <xs:element name="Operation"
                    type="tDataEconomyOperation" />
        </xs:sequence>
</xs:complexType>

<xs:simpleType name="tDataEconomyOperation">
        <xs:restriction base="xs:string">
                <xs:enumeration value="delete" />
                <xs:enumeration value="lock" />
        </xs:restriction>
</xs:simpleType>


<xs:complexType name="tLimitedUseProperties">
        <xs:sequence>
                <xs:element name="Purpose" type="xs:string" />
                <xs:element name="FreeUse" type="xs:boolean" />
        </xs:sequence>
</xs:complexType>

<xs:complexType name="tConfidentialityProperties">
        <xs:sequence>
                <xs:element name="AccessAuthority" type="xs:boolean" />
        </xs:sequence>
</xs:complexType>

<xs:complexType name="tIntegrityProperties">
        <xs:sequence>
                <xs:element name="DataProtectionRequest"
                    type="xs:boolean" fixed="true" />
```

```
                <xs:element name="DataIntegrityRequest"
                    type="xs:boolean" fixed="true" />
                <xs:element name="KeyIntegrityRequest"
                    type="xs:boolean" fixed="true" />
        </xs:sequence>
</xs:complexType>

<xs:complexType name="tAvailabilityProperties">
        <xs:sequence>
                <xs:element name="AccessAllowed" type="xs:boolean"
                        fixed="true" />
                <xs:element name="AccessAffect" type="xs:boolean"
                        fixed="false" />
                <xs:element name="InstanceRestorable" type="xs:boolean"
                    fixed="true" />
        </xs:sequence>
</xs:complexType>

<xs:complexType name="tAuthenticityProperties">
        <xs:sequence>
                <xs:element name="Assignable" type="xs:boolean"
                    fixed="true" />
                <xs:element name="Changeable" type="xs:boolean"
                    fixed="false" />
                <xs:element name="PartnerIdentified" type="xs:boolean"
                        fixed="true" />
                <xs:element name="KeyDistributionId" type="xs:boolean"
                        fixed="true" />
        </xs:sequence>
</xs:complexType>

<xs:complexType name="tPseudonymityProperties">
        <xs:sequence>
                <xs:element name="PseudonymIdentifiable"
                    type="xs:boolean"
                        fixed="true" />
                <xs:element name="UnauthorizedAccess" type="xs:boolean"
                    fixed="false" />
                <xs:element name="UserInfoCheckable" type="xs:boolean"
                        fixed="true" />
                <xs:element name="UserProfileEditable" type="xs:boolean"
                        fixed="true" />
        </xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="tDataProtectionLevelProperties">
       <xs:sequence>
              <xs:element name="DataProtectionLevel"
                  type="tDataProtectionLevel" />
       </xs:sequence>
</xs:complexType>

<xs:simpleType name="tDataProtectionLevel">
       <xs:restriction base="xs:string">
              <xs:enumeration value="extremly low" />
              <xs:enumeration value="low" />
              <xs:enumeration value="normal" />
              <xs:enumeration value="high" />
              <xs:enumeration value="greatly high" />
       </xs:restriction>
</xs:simpleType>


<xs:complexType name="tInputValidationProperties">
       <xs:sequence>
              <xs:element name="ValidationRequest" type="xs:boolean"
                     fixed="true" />
       </xs:sequence>
</xs:complexType>

<xs:complexType name="tReferenceInternalObjectProperties">
       <xs:sequence>
              <xs:element name="DirectReferenceInternalObject"
                  type="xs:boolean"
                     fixed="false" />
       </xs:sequence>
</xs:complexType>

<xs:complexType name="tVMMonitorProperties">
       <xs:element name="VMMRequest" type="xs:boolean" fixed="true" />
       <xs:element name="LoginDataLogRequest" type="xs:boolean"
          fixed="true" />
</xs:complexType>

<xs:complexType name="tBackupProperties">
       <xs:sequence>
              <xs:element name="OfflineBackupPeriod"
                  type="xs:duration" />
              <xs:element name="OfflineBackupTime" type="xs:time" />
```

```
                <xs:element name="BackupTestPeriod" type="xs:duration"
                    />
                <xs:element name="OnlineBackup" type="xs:boolean" />
                <xs:element name="RedundantBackup" type="xs:boolean"
                        fixed="true" />
                <xs:element name="DeleteExpiredBackup"
                    type="xs:duration" />
        </xs:sequence>
</xs:complexType>

<xs:complexType name="tStandardDataFormatProperties">
        <xs:sequence>
                <xs:element name="DataFormat" type="tDataFormat" />
        </xs:sequence>
</xs:complexType>

<xs:simpleType name="tDataFormat">
        <xs:restriction base="xs:string">
                <xs:enumeration value="SQL" />
                <xs:enumeration value="JSON" />
                <xs:enumeration value="XML" />
        </xs:restriction>
</xs:simpleType>

<xs:complexType name="tIsolationProperties">
        <xs:sequence>
                <xs:element name="DataAccessAuthorizeRequest"
                    type="xs:boolean"
                        fixed="true" />
                <xs:element name="ExternDataAccessible"
                    type="xs:boolean"
                        fixed="false" />
                <xs:element name="UserDataInvisible" type="xs:boolean"
                        fixed="true" />
        </xs:sequence>
</xs:complexType>

<xs:complexType name="tDataDeletionProperties">
        <xs:sequence>
                <xs:element name="DeleteApplicationData"
                    type="xs:boolean"
                        fixed="true" />
                <xs:element name="DeleteCustomerData" type="xs:boolean"
                        fixed="true" />
```

```
                <xs:element name="UserDataOperation"
                    type="tUserDataOperation" />
        </xs:sequence>
</xs:complexType>

<xs:simpleType name="tUserDataOperation">
        <xs:restriction base="xs:string">
                <xs:enumeration value="delete" />
                <xs:enumeration value="transmit" />
        </xs:restriction>
</xs:simpleType>

<xs:complexType name="tKeyStorageProperties">
        <xs:sequence>
                <xs:element name="SeparateStorageRequest"
                    type="xs:boolean"
                        fixed="true" />
                <xs:element name="KeyFileEncryptRequest"
                    type="xs:boolean"
                        fixed="true" />
                <xs:elememt name="EncryptionPurpose"
                    type="tEncryptionPurpose" />
        </xs:sequence>
</xs:complexType>

<xs:simpleType name="tEncryptionPurpose">
        <xs:restriction base="xs:string">
                <xs:enumeration value="key for encryption of data" />
                <xs:enumeration value="key for encryption of keys" />
                <xs:enumeration value="key for authentication" />
                <xs:enumeration value="key for the creation of
                    electronic signatures" />
        </xs:restriction>
</xs:simpleType>

<xs:complexType name="tDatabaseMonitorProperties">
        <xs:sequence>
                <xs:element name="ActivityMonitorRequest"
                    type="xs:boolean"
                        fixed="true" />
        </xs:sequence>
</xs:complexType>

<xs:complexType name="tConcurrencyProperties">
        <xs:sequence>
```

```
                    <xs:element name="SynchronizationRequest"
                        type="xs:boolean"
                            fixed="true" />
                    <xs:element name="AccessAccepted" type="xs:boolean"
                            fixed="true" />
                </xs:sequence>
        </xs:complexType>

</xs:schema>
```

5.3 Policy Type und Policy Template Beschreibung

Nach der Festlegung aller Property-Elemente in dem Schema Dokument wird jede Policy Type Definition auf das für sie passende Property-Element referenzieren. Wie bereits im vorangegangenen Abschnitt 2.4.1 –TOSCA Grundlagen erwähnt, werden die geeigneten Werte für die Property-Elemente je nach der Vereinbarung zwischen den Cloud-Anbieter und Kunden (in Form eines Vertrags oder über SLA) zudem in den Policy Template zugewiesen.

Hinweise: xmlns:spp="http://www.example.com/tosca/Types/PolicyProperties"

- **Richtlinie: Transparenz**

```
<PolicyType name="TransparencyPolicyType">
        <PropertiesDefinition element="spp:TransparencyProperties" />
</PolicyType>

<PolicyTemplate type="TransparencyPolicyType" id="MyTransparencyPolicy">
        <Properties>
                <spp:TransparencyProperties>
<!-- Whether the Log Data should be saved or not. Default value is TRUE.
    -->
                        <LogDataRequest>true</LogDataRequest>
                </spp:TransparencyProperties>
        </Properties>
</PolicyTemplate>
```

- **Richtlinie: Datenvermeidbarkeit**

```
<PolicyType name="DataAvoidabilityPolicyType">
        <PropertiesDefinition element="spp:DataAvoidabilityProperties" />
</PolicyType>

<PolicyTemplate type="DataAvoidabilityPolicyType"
    id="MyDataAvoidabilityPolicy">
```

```
        <Properties>
                <spp:DataAvoidabilityProperties>
<!-- Whether the Data should be encrypted or not. Default value is TRUE.
    -->
<!-- For the critical business und personal data, the value of this
    property schoud be always TRUE. -->
                        <EncryptionRequest>true</EncryptionRequest>
                </spp:DataAvoidabilityProperties>
        </Properties>
</PolicyTemplate>
```

- **Richtlinie: Datensparsamkeit**

```
<PolicyType name="DataEconomyPolicyType">
        <PropertiesDefinition element="spp:DataEconomyProperties" />
</PolicyType>

<PolicyTemplate type="DataEconomyPolicyType" id="MyDataEconomyPolicy">
        <Properties>
                <spp:DataEconomyProperties>
<!-- If the data has not yet been used during the period, which is set
    by property TimeWindow, then it should be deleted or locked, just
    depending on the property Operation. -->
                        <TimeWindow>P10DT12H</TimeWindow>
                        <Operation>delete</Operation>
                </spp:DataEconomyProperties>
        </Properties>
</PolicyTemplate>
```

- **Richtlinie: Zweckbindung**

```
<PolicyType name="LimitedUsePolicyType">
        <PropertiesDefinition element="spp:LimitedUseProperties" />
</PolicyType>

<PolicyTemplate type="LimitedUsePolicyType" id="MyLimitedUsePolicy">
        <Properties>
                <spp:LimitedUseProperties>
<!-- Each processing of personal data shoud be associated to a certain
    purpose, which must be informed in advance to the related customer.
    This will be describted in the Purpose property. -->
                        <Purpose>validation</Purpose>
<!-- Whether the Data could be freely used without informing the related
    person or not. Default value is FALSE. -->
                        <FreeUse>false</FreeUse>
                </spp:LimitedUseProperties>
```

```
        </Properties>
</PolicyTemplate>
```

- **Richtlinie: Vertraulichkeit**

```
<PolicyType name="ConfidentialityPolicyType">
        <PropertiesDefinition element="spp:ConfidentialityProperties" />
</PolicyType>

<PolicyTemplate type="ConfidentialityPolicyType"
    id="MyConfidentialityPolicy">
        <Properties>
                <spp:ConfidentialityProperties>
<!-- Whether an unauthorized access to Data should be allowed or not.
    Default value is FALSE. -->
                        <AccessAuthority>true</AccessAuthority>
                </spp:ConfidentialityProperties>
        </Properties>
</PolicyTemplate>
```

- **Richtlinie: Integrität**

```
<PolicyType name="IntegrityPolicyType">
        <PropertiesDefinition element="spp:IntegrityProperties" />
</PolicyType>

<PolicyTemplate type="IntegrityPolicyType" id="MyIntegrityPolicy">
        <Properties>
                <spp:IntegrityProperties>
<!-- Whether a protection against unautorized manipulation of data is
    requested. The value of this property must be always TRUE. -->
                   <DataProtectionRequest>true</DataProtectionRequest>
<!-- Whether a validation of data Integrity during the data processing
    is requested. The value of this property must be always TRUE. -->
                   <DataIntegrityRequest>true</DataIntegrityRequest>
<!-- A key should be integerity after distribution. -->
                   <KeyIntegrityRequest>true</KeyIntegrityRequest>
                </spp:IntegrityProperties>
        </Properties>
</PolicyTemplate>
```

- **Richtlinie: Verfügbarkeit**

```
<PolicyType name="AvailabilityPolicyType">
        <PropertiesDefinition element="spp:AvailabilityProperties" />
</PolicyType>
```

```
<PolicyTemplate type="AvailabilityPolicyType" id="MyAvailabilityPolicy">
       <Properties>
              <spp:AvailabilityProperties>
<!-- Access to the required resource should be allowed at all times, if
   this access is authorized. The value of this property should be
   fixedly TRUE. -->
                     <AccessAllowed>true</AccessAllowed>
<!-- Access should not be affected by attacks. The value of this
   property should be fixedly FALSE. -->
                     <AccessAffected>false</AccessAffected>
<!-- Every instance of application should be restorable for the
   customer. Restoration is based on the back-up files. The value of
   this property should be fixedly TRUE. -->
                     <InstanceRestorable>true</InstanceRestorable>
              </spp:AvailabilityProperties>
       </Properties>
</PolicyTemplate>
```

- **Richtlinie: Authentizität**

```
<PolicyType name="AuthenticityPolicyType">
       <PropertiesDefinition element="spp:AuthenticityProperties" />
</PolicyType>

<PolicyTemplate type="AuthenticityPolicyType" id="MyAuthenticityPolicy">
       <Properties>
              <spp:AuthenticityProperties>
<!-- Whether all of the critical data could be assigned to their
   original owners at any time. And whether these data could be modified
   by anyone else after departure from their owners. -->
                  <Assignable>true</Assignable>
                  <Changeable>false</Changeable>
<!-- Whether both of the communication partners are identified or not.
   The default value of this property is TRUE.-->
                  <PartnerIdentified>true</PartnerIdentified>
<!-- Whether the key distribution process is identified or not. The
   default value of this property is TRUE.-->
                  <KeyDistributionId>true</KeyDistributionId>
              </spp:AuthenticityProperties>
       </Properties>
</PolicyTemplate>
```

- **Richtlinie: Pseudonymität**

```
<PolicyType name="PseudonymityPolicyType">
```

```
        <PropertiesDefinition element="spp:PseudonymityProperties" />
</PolicyType>

<PolicyTemplate type="PseudonymityPolicyType" id="MyPseudonymityPolicy">
        <Properties>
                <spp:PseudonymityProperties>
<!--The pseudonym should be unique and identifiable. The value of this
    property should be fixedly TRUE.-->
                    <PseudonymIdentifiable>true</PseudonymIdentifiable>
<!-- Whether anyone can access to critical data without login or not.
    The value of this property should be fixedly FALSE.-->
                    <UnauthorizedAccess>false</UnauthorizedAccess>
                    <UserInfoCheckable>true</UserInfoCheckable>
                    <UserProfileEditable>true</UserProfileEditable>
                </spp:PseudonymityProperties>
        </Properties>
</PolicyTemplate>
```

- **Richtlinie: Angemessenes Datenschutzniveau**

```
<PolicyType name="DataProtectionLevelPolicyType">
     <PropertiesDefinition element="spp:DataProtectionLevelProperties" />
</PolicyType>

<PolicyTemplate type="DataProtectionLevelPolicyType"
    id="CriticalDataProtectionLevelPolicy">
        <Properties>
                <spp:DataProtectionLevelProperties>
<!-- The permission of a critical data transmission outside EWR depends
    on the local protection level. This would be set in the property of
    DataProtectionLevel. -->
                        <DataProtectionLevel>high</DataProtectionLevel>
                </spp:DataProtectionLevelProperties>
        </Properties>
</PolicyTemplate>
```

- **Richtlinie: Überprüfung von Benutzereingaben**

```
<PolicyType name="InputValidationPolicyType">
        <PropertiesDefinition element="spp:InputValidationProperties" />
</PolicyType>

<PolicyTemplate type="InputValidationPolicyType"
    id="SaaSInputValidationPolicy">
        <Properties>
```

```
                <spp:InputValidationProperties>
<!-- Whether the input data should be verified against SQL injection
    attack or not. The value of this property by SaaS service model
    should be fixedly TRUE. -->
                        <ValidationRequest>true</ValidationRequest>
                </spp:InputValidationProperties>
        </Properties>
</PolicyTemplate>
```

- **Richtlinie: Referenzen direkt auf interne Objekte**

```
<PolicyType name="ReferenceInternalObjectPolicyType">
        <PropertiesDefinition
            element="spp:ReferenceInternalObjectProperties" />
</PolicyType>

<PolicyTemplate type="ReferenceInternalObjectPolicyType"
    id="MyReferenceInternalObjectPolicy">
        <Properties>
            <spp:ReferenceInternalObjectProperties>
<!-- Whether a direct reference to internal object is allowed or not.
    The defaul value of this property is FALSE. -->
              <DirectReferenceInternalObject>false</DirectReferenceInternalObject>
            </spp:ReferenceInternalObjectProperties>
        </Properties>
</PolicyTemplate>
```

- **Richtlinie: Einsatz von hochmodernen Hypervisoren**

```
<PolicyType name="VMMonitorPolicyType">
        <PropertiesDefinition element="spp:VMMonitorProperties" />
</PolicyType>

<PolicyTemplate type="VMMonitorPolicyType" id="IaaSVMMonitorPolicy">
        <Properties>
                <spp:VMMonitorProperties>
<!-- Whether a certified and hardened VM monitor should be used or not.
    The defaul value of this property is TRUE. -->
                    <VMMRequestRequest>true</VMMRequestRequest>
<!-- Whether the information of login action to the VM monitor should be
    logged or not. The value of this property by IaaS service model
    should be fixedly TRUE. -->
                    <LoginDataLogRequest>true</LoginDataLogRequest>
                </spp:VMMonitorProperties>
        </Properties>
</PolicyTemplate>
```

- **Richtlinie: Regelmäßige Datensicherungen**

```
<PolicyType name="BackupPolicyType">
       <PropertiesDefinition element="spp:BackupProperties" />
</PolicyType>

<PolicyTemplate type="BackupPolicyType" id="MyBackupPolicy">
   <Properties>
        <spp:BackupProperties>
<!-- The OfflineBackupPeriod property sets the time interval, how often
    the data should be offline backed up. Value P1M means every one
    month. -->
            <OfflineBackupPeriod>P1M</OfflineBackupPeriod>
<!-- The OfflineBackupTime property sets the exact time of the offline
    backup. -->
            <OfflineBackupTime>23:59:59Z</OfflineBackupTime>
<!-- The BackupTestPeriod property sets the time interval, how often the
    system should be checked, if it can be restored based on the backup
    data. Value P1D means every 10 days. -->
            <BackupTestPeriod>P10D</BackupTestPeriod>
<!-- Whether a online backup is needed or not. The value of this
    property should be TRUE when high availability is requested by this
    cloud service. -->
            <OnlineBackup>false</OnlineBackup>
<!-- Whether the data should be redundant saved or not. The value of
    this property should be fixedly TRUE. -->
            <RedundantBackup>true</RedundantBackup>
<!-- No longer needed backup data should be deleted after a predefined
    time period, which will be set by the DeleteExpiredBackup property.
    Value P12M means every 12 months. -->
            <DeleteExpiredBackup>P12M</DeleteExpiredBackup>
        </spp:BackupProperties>
   </Properties>
</PolicyTemplate>
```

- **Richtlinie: Standard-Datenformat**

```
<PolicyType name="StandardDataFormatPolicyType">
       <PropertiesDefinition element="spp:StandardDataFormatProperties"
          />
</PolicyType>
<!-- Different policy templates could be carried out, which is depend on
    the different values of a property. -->
```

```
<PolicyTemplate type="StandardDataFormatPolicyType"
    id="SQL92StandardDataFormatPolicy">
        <Properties>
               <spp:StandardDataFormatProperties>
                      <DataFormat>SQL-92</DataFormat>
               </spp:StandardDataFormatProperties>
        </Properties>
</PolicyTemplate>

<PolicyTemplate type="StandardDataFormatPolicyType"
    id="SQL2003StandardDataFormatPolicy">
        <Properties>
               <spp:StandardDataFormatProperties>
                      <DataFormat>SQL-2003</DataFormat>
               </spp:StandardDataFormatProperties>
        </Properties>
</PolicyTemplate>
```

- **Richtlinie: Isolierung**

```
<PolicyType name="IsolationPolicyType">
       <PropertiesDefinition element="spp:IsolationProperties" />
</PolicyType>

<PolicyTemplate type="IsolationPolicyType" id="MyIsolationPolicy">
    <Properties>
         <spp:IsolationProperties>
<!-- Data of a application instance could be accessed exclusive by its
    user. For the others this instance is not accessible. So the value of
    the ExternDataAccessible property should be fixedly FALSE. -->
            <DataAccessAuthorizeRequest>true</DataAccessAuthorizeRequest>
            <ExternDataAccessible>false</ExternDataAccessible>
<!-- Whether the data of a user is visible to the other users, who also
    owns a virtual machine on the same server. -->
            <UserDataInvisible>true</UserDataInvisible>
         </spp:IsolationProperties>
    </Properties>
</PolicyTemplate>
```

- **Richtlinie: Löschung der Anwendungsdaten und Nutzdaten**

```
<PolicyType name="DataDeletionPolicyType">
       <PropertiesDefinition element="spp:DataDeletionProperties" />
</PolicyType>

<PolicyTemplate type="DataDeletionPolicyType" id="MyDataDeletionPolicy">
```

```
    <Properties>
        <spp:DataDeletionProperties>
<!-- Whether the user instance and the user's personal informations that
    belong to this instance should be deleted by the cloud vendor after
    ending the contract or not. The default value of both these
    properties are TRUE. -->
            <DeleteApplicationData>true</DeleteApplicationData>
            <DeleteCustomerData>true</DeleteCustomerData>
<!-- After ending the contract, the user data should be either deleted
    directly by the old vendor or transmitted to the new cloud vendor. -->
            <UserDataOperation>transmit</UserDataOperation>
        </spp:DataDeletionProperties>
    </Properties>
</PolicyTemplate>
```

- **Richtlinie: Schlüsselaufbewahrung und -management**

```
<PolicyType name="KeyStoragePolicyType">
       <PropertiesDefinition element="spp:KeyStorageProperties" />
</PolicyType>

<!-- e.g. Keys with different purposes could be classified into
    different templates with the corresponding id. -->

<PolicyTemplate type="KeyStoragePolicyType" id="DoubleKeyStoragePolicy">
    <Properties>
          <spp:KeyStorageProperties>
<!-- Whether the key should be stored separately from its encrypted data
    or not. The default value of this property is TRUE. -->
            <SeparateStorageRequest>true</SeparateStorageRequest>
<!-- Whether the file, in which the key has been stored, should be also
    encrypted or not. The default value of this property is TRUE. -->
            <KeyFileEncryptRequest>true</KeyFileEncryptRequest>
<!-- This property shows the purpose of this kind of keys. -->
            <EncryptionPurpose>key for encryption of
                keys</EncryptionPurpose>
          </spp:KeyStorageProperties>
    </Properties>
</PolicyTemplate>

<PolicyTemplate type="KeyStoragePolicyType"
    id="AuthenticationKeyStoragePolicy">
    <Properties>
        <spp:KeyStorageProperties>
            <SeparateStorageRequest>true</SeparateStorageRequest>
```

```
            <KeyFileEncryptRequest>true</KeyFileEncryptRequest>
            <EncryptionPurpose>key for authentication</EncryptionPurpose>
        </spp:KeyStorageProperties>
    </Properties>
</PolicyTemplate>
```

- **Richtlinie: Überwachung von Datenbankaktivitäten**

```
<PolicyType name="DatabaseMonitorPolicyType">
       <PropertiesDefinition element="spp:DatabaseMonitorProperties" />
</PolicyType>

<PolicyTemplate type="DatabaseMonitorPolicyType"
    id="MyDatabaseMonitorPolicy">
    <Properties>
        <spp:DatabaseMonitorProperties>
<!-- Whether a database activity monitor is needed or not. The value of
    this property should be fixedly TRUE.-->
            <ActivityMonitorRequest>true</ActivityMonitorRequest>
        </spp:DatabaseMonitorProperties>
    </Properties>
</PolicyTemplate>
```

- **Richtlinie: Konsistenz (engl. Concurreny) nach der Synchronisation**

```
<PolicyType name="ConcurrencyPolicyType">
       <PropertiesDefinition element="spp:ConcurrencyProperties" />
</PolicyType>

<PolicyTemplate type="ConcurrencyPolicyType" id="MyConcurrencyPolicy">
    <Properties>
        <spp:ConcurrencyProperties>
<!-- Whether a sychronization is requested, when the original data have
    been modified. The value of this property should be fixedly TRUE. -->
            <SynchronizationRequest>true</SynchronizationRequest>
<!-- Whether an access to the data is accepted or not, when the
    synchronization is not complete yet. The value of this property
    should be fixedly FALSE. -->
            <AccessAccepted>false</AccessAccepted>
        </spp:ConcurrencyProperties>
    </Properties>
</PolicyTemplate>
```

6 Zusammenfassung und Ausblick

Abschließend werden in diesem Kapitel die Ergebnisse dieser Diplomarbeit zusammengefasst. Zuerst wird ein Fazit gezogen, wie die Ziele der Aufgabenstellung erreicht werden. Weiterhin befinden sich die offen geblieben Fragen und die möglichen Verbesserungen in Unterkapitel 6.2.

6.1 Zusammenfassung

Die nächste Evolutionsstufe des Cloud-Computing hat bereits begonnen. Nach Speicherdiensten und selbst geschäftskritischen Anwendungen fangen nun die ersten Unternehmen und Organisationen damit an, große Datenmengen in die Cloud zu verschieben und diese dort zu organisieren und bearbeiten [Sch13]. Während mehrere Kunden ihre Daten in die Cloud auslagern, verschärfen sich die Datensicherheitsprobleme aber auch gleichzeitig.

Das Ziel der vorliegenden Diplomarbeit war die Erstellung und Beschreibung von RDBMS-zentrischen Sicherheitsrichtlinien, die beim Betrieb von Cloud-Diensten eingesetzt werden können, damit die Cloud-Dienste in der Zukunft im Rahmen von TOSCA bei garantierter Sicherheit und Compliance bereitgestellt werden könnten.

Nach der grundsätzlichen Einleitung und Aufgabenstellung wurden in Kapitel 2 die Grundlagen kurz erläutert, die zum Verständnis dieser Arbeit besonders notwendig sind. In Kapitel 3 wurde in erster Linie ein Richtlinienkatalog für Datensicherheit erstellt. Des Weiteren wurde auf die verschiedenen Verschlüsselungsverfahren eingegangen, die ein wesentlicher Bestandteil der Datensicherheit sind. Ferner wurden in Kapitel 4 Taxonomien anhand dieses Richtlinienkatalogs abgeleitet, die meist auf bestimmten Merkmale vom Cloud Computing und relationalen DBMS basieren. Abschließend beschäftigt sich das fünfte Kapitel mit der in der Implementierung verwendeten TOSCA Sprache sowie der Art und Weise der Implementierung.

6.2 Ausblick

Eine in dieser Diplomarbeit entwickelte Taxonomie beschränkt sich auf die existierenden Service-Modelle der Cloud Architekturen: PaaS, IaaS und Saas. Um die Datensicherheit besser zu steuern, hat man seit kurzem auch die Möglichkeit die Database as a Service (DbaaS) [SK12] in der Cloud zu nutzen. Diese ermöglicht neben der Bereitstellung grundlegender

Funktionalität eines relationalen DBMS die Authentifizierungs- und Autorisierungsfunktionen, um den Datenzugriff feingranular zu steuern [SK12, HV10]. Hierfür ist in der Zukunft die Erstellung der Sicherheitsrichtlinien insbesondere im Hinblick auf diesen Bereich notwendig.

Um den Richtlinienkatalog der Datensicherheit zu vervollständigen und zu erweitern, können die vorgeschlagenen Verbesserungen aus Unterkapitel 5.2 implementiert werden, indem mehre Elemente bei der Property-Definition hinzugefügt werden, um ein Richtlinie-Property besser zu begrenzen.

Ein weiteres Problem, das wir in der Zukunft beheben müssen, liegt darin: was sollte man tun, falls zwei widersprüchliche Richtlinien mit einem gleichen Knoten bzw. einer gleichen Service-Komponente in TOSCA annotiert sind? Eine mögliche Lösung wäre, jeder Richtlinie vor dem Einsatz eine Priorität zu geben. Dadurch können die Gewichte der beiden Richtlinien vergleichen und davon die wichtigere auf diese Service-Komponente angewandt werden.

Da die in dieser Arbeit ausgearbeiteten Richtlinien nur auf wissenschaftlicher Literatur basieren, reicht dies jedoch im Allgemeinen nicht, um über Praxistauglichkeit des Konzeptes eine Aussage zu treffen. Aus diesem Grund werden weitere Erfahrungen aus der Praxis benötigt, um solche Richtlinien zu beurteilen und gegebenfalls zu verbessern.

Abkürzungsverzeichnis

ACID Atomicity-Consistency-Isolation-Durability

BPMN Business Process Model and Notation

DAC Discretionary Access Control

DAM Database Activity Monitoring

DbaaS Database as a Service

EWR Staaten des Europäischen Wirtschaftsraums

IaaS Infrastructure as a Service

ICSF Integrated Kryptographisch Service Facility

MAC Mandatory Access Control

NIST National Institute of Standards and Technology

OLAP Online Analytical Processing

OLTP Online Transaction Processing

PaaS Platform as a Service

QoS Quality-of-Services

RDBMS Relational Database Management System

RDS Relational Database Service

SaaS Software as a Service

SLA Service Level Agreement

TDE Transparent Data Encryption

TOSCA Topology and Orchestration Specification for Cloud Applications

VM virtuelle Maschine

Literaturverzeichnis

[Aba09] D. J. Abadi. Data management in the cloud: Limitations and opportunities. *IEEE Data Eng. Bull*, 32(1):3–12, 2009. (Zitiert auf den Seiten 9 und 18)

[AKH03] J. H. Anderson, Y.-J. Kim, T. Herman. Shared-memory mutual exclusion: major research trends since 1986. *Distributed Computing*, 16(2-3):75–110, 2003. (Zitiert auf Seite 35)

[Ama] Amazon. Amazon Virtual Private Cloud (Amazon VPC). http://aws.amazon.com/de/vpc/. (Zitiert auf Seite 25)

[Aue13] J. Auer. Grundbegriffe und Konzepte von Datenbank-Systemen. http://www.sql-und-xml.de/sql-tutorial/datenbank-grundbegriffe.html, 2013. (Zitiert auf Seite 14)

[BBLS12] T. Binz, G. Breiter, F. Leyman, T. Spatzier. Portable cloud services using tosca. *Internet Computing, IEEE*, 16(3):80–85, 2012. (Zitiert auf den Seiten 5, 9, 19 und 20)

[Bey] R. Beyer. Cloud-Infrastrukturen. (Zitiert auf den Seiten 9 und 15)

[BIT11] BITKOM. Cloud Computing ist erneut IT-Trend des Jahres. *Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V.*, S. 4, 2011. (Zitiert auf Seite 7)

[Bla12] R. Blaisdell. Mandantenfähigkeit in der Cloud: Die Vorteile verstehen. http://www.enterprisecioforum.com/de/blogs/rickblaisdell/mandantenf%C3%A4higkeit-der-cloud-die-vorteil, 2012. (Zitiert auf Seite 32)

[Boe10] D. Boesswetter. Spaltenorientierte Datenbanken. *Informatik Spektrum*, 2010. (Zitiert auf Seite 46)

[Brö09] B. Bröcker. Verschlüsselung, Anonymisierung und Bereitstellung sicherheitsrelevanter Daten im Projekt Med-On-Mix. 2009. (Zitiert auf den Seiten 5, 35 und 36)

[BT11] R. Bröcker, J. Tiemeyer. Relationale Cloud-Datenbanken, ein aktueller Vergleich. *Informatik-Spektrum*, 34(1):90–98, 2011. (Zitiert auf Seite 15)

[Bus12] D. M. Busch. Definition: Cloud Computing. http://blog.team-neusta.de/index.php/spektrum/definition-cloud-computing/, 2012. (Zitiert auf Seite 11)

[Com13a] O. T. Committee. OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) Overview. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca, 2013. (Zitiert auf den Seiten 9 und 19)

[Com13b] O. T. Committee. Topology and Orchestration Specification for Cloud Applications Version 1.0. http://docs.oasis-open.org/tosca/TOSCA/v1.0/cs01/TOSCA-v1.0-cs01.pdf, 2013. (Zitiert auf den Seiten 9, 20 und 53)

[Cor13] I. Corporation. DB2 encryption and decryptionIBM Corporation. http://pic.dhe.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.imstools.dec.doc.ug/topics/decucon_db2-oview.htm, 2013. (Zitiert auf den Seiten 5, 9, 37, 38 und 39)

[CPK10] Y. Chen, V. Paxson, R. H. Katz. What's new about cloud computing security? *University of California, Berkeley Report No. UCB/EECS-2010-5 January*, 20(2010):2010–5, 2010. (Zitiert auf den Seiten 9 und 17)

[Dat] Datenschutzbeauftragter. Die sieben Grundprinzipien im Datenschutz. http://www.datenschutzbeauftragter-info.de/fachbeitraege/die-sieben-grundprinzipien-im-datenschutz/. (Zitiert auf den Seiten 9 und 23)

[DSH12] U. L. für Datenschutz Schleswig-Holstein. ULD: „Datenschutzkonformes Cloud Computing ist möglich". https://www.datenschutzzentrum.de/presse/20120713-datenschutzkonformes-cloud-computing.htm, 2012. (Zitiert auf Seite 27)

[Fab12] H.-W. Fabry. Mehr Sicherheit für Netzwerkverbindungen. http://www.oracle.com/webfolder/technetwork/de/community/dbadmin/tipps/net/index.html, 2012. (Zitiert auf Seite 30)

[Gö10] A. Göbel. Anforderungen von Cloud-Anwendungen an Datenbanksysteme. In *Workshop Database as a Service*. 2010. (Zitiert auf den Seiten 9 und 16)

[Gab11] M. Gabel. AES-Verschlüsselung in MySQL. http://blog.blueend.com/2011/06/aes-verschlusselung-in-mysql/, 2011. (Zitiert auf Seite 9)

[Gmb13] P. GmbH. Log4View Logging Frameworks. http://www.log4view.de/support/logging-frameworks/, 2013. (Zitiert auf Seite 23)

[Han12] M. Hansen. Vertraulichkeit und Integrität von Daten und IT-Systemen im Cloud-Zeitalter. *Datenschutz und Datensicherheit-DuD*, 36(6):407–412, 2012. (Zitiert auf den Seiten 9 und 25)

[HV10] T. Haselmann, G. Vossen. Database-as-a-Service für kleine und mittlere Unternehmen. Technischer Bericht, Working Paper (3), Institut für Wirtschaftsinformatik, Westfälische Wilhelms-Universität Münster, Münster, 2010. (Zitiert auf Seite 72)

[HWF] O. D. G. Heinz-Wilhelm Fabry. Daten verschluesseln mit Transparent Daten Encryption(TDE). http://www.oracle.com/webfolder/technetwork/de/community/dbadmin/tipps/tde/index.html. (Zitiert auf den Seiten 9 und 35)

[Inf] B. für Sicherheit in der Informationstechnik. Cloud Computing Grundlagen. https://www.bsi.bund.de/DE/Themen/CloudComputing/Grundlagen/Grundlagen_node.html. (Zitiert auf den Seiten 9, 11, 12 und 13)

[Info04] M.-P.-I. für Informatik. Die Datenbanksprache SQL. http://www.mpi-inf.mpg.de/departments/d5/teaching/ss04/is04/skripte/kap7.pdf, 2004. (Zitiert auf Seite 32)

[jes11] jesser. Database Activity Monitoring Lösungen. http://www.it-experts.edilog.de/2011/09/08/database-activity-monitoring-losungen/, 2011. (Zitiert auf Seite 34)

[Kö11] I. Köpke. Cloud im Mittelstand: Transparenz der Angebote ist gefragt. http://www.cloud-practice.de/news/cloud-im-mittelstand-transparenz-der-angebote-ist-gefragt, 2011. (Zitiert auf Seite 7)

[Kac08] E. Kachel. SQL-Injections - eine Analyse an PHP & MySQL. http://www.erich-kachel.de/?p=223, 2008. (Zitiert auf den Seiten 5 und 29)

[KBBL12] O. Kopp, T. Binz, U. Breitenbuecher, F. Leymann. BPMN4TOSCA: A Domain-Specific Language to Model Management Plans for Composite Applications. In *Business Process Model and Notation*, S. 38–52. Springer, 2012. (Zitiert auf den Seiten 9 und 19)

[KE11] A. Kemper, A. Eickler. *Datenbanksysteme: Eine Einführung*. Oldenbourg Verlag, 2011. (Zitiert auf Seite 16)

[Kno09] E. Knorr. Gmail follies and Google's enterprise pitch. *InfoWorld. September*, 8, 2009. (Zitiert auf Seite 17)

[Koo12] A. Koop. Oracle Java and Database Cloud Services in Action. http://multikoop.blogspot.de/2012/11/oracle-java-and-database-cloud-services.html, 2012. (Zitiert auf Seite 35)

[Kur13] W. Kurzlechner. Backup aus der Public Cloud. http://www.computerwoche.de/a/backup-aus-der-public-cloud,2532959, 2013. (Zitiert auf Seite 31)

[LNW00] D. I. des Landes Nordrhein-Westfalen. Gesetz zum Schutz personenbezogener Daten (Datenschutzgesetz Nordrhein-Westfalen). 2000. (Zitiert auf den Seiten 24 und 27)

[MG11] P. Mell, T. Grance. The NIST definition of cloud computing (draft). *NIST special publication*, 800:145, 2011. (Zitiert auf Seite 11)

[MNS+07] C. Mückschel, J. Nieschulze, B. Sloboda, C. Weist, W. Köhler. Herausforderungen, Probleme und Lösungsansätze im Datenmanagement von Sonderforschungsbereichen. *Gesellschaft für Land-, Forst-und Ernährungswirtschaft eV eZAI (2)*, S. 1–16, 2007. (Zitiert auf Seite 32)

[MyS13a] MySQL. MySQL 5.1 Referenzhandbuch 12.10.2. Verschlüsselungs- und Kompressionsfunktionen. http://dev.mysql.com/doc/refman/5.1/de/encryption-functions.html, 2013. (Zitiert auf den Seiten 9 und 36)

[MyS13b] MySQL. MySQL 5.1 Referenzhandbuch 5.9.7. Verwendung sicherer Verbindungen. http://dev.mysql.com/doc/refman/5.1/de/secure-connections.html, 2013. (Zitiert auf den Seiten 9 und 36)

[RTSS09] T. Ristenpart, E. Tromer, H. Shacham, S. Savage. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *Proceedings of the 16th ACM conference on Computer and communications security*, S. 199–212. ACM, 2009. (Zitiert auf Seite 17)

[Rup10] A. Ruppel. Cloud-Computing-Systeme und ihre Schutzziele. http://www.searchsecurity.de/themenbereiche/applikationssicherheit/web-application-security/articles/249238/, 2010. (Zitiert auf den Seiten 9 und 25)

[Sch] S. Scheuermann. Cloud Lösungen mit Windows Azure. (Zitiert auf Seite 15)

[Sch13] G. Schukat. IT-Cloud und big data für big business im Mittelstand. *Weltmeister Magazin, Beilage der VDI Nachrichten*, 2:11–12, 2013. (Zitiert auf Seite 71)

[Sea12] E. P. Seal. Datenschutzrechtliche Anforderungen an Cloud Computing. https://www.european-privacy-seal.eu/results/fact-sheets/Cloud%20Computing%20FS-201207-DE.pdf, 2012. (Zitiert auf Seite 18)

[SK12] M. Seibold, A. Kemper. Database as a Service. *Datenbank-Spektrum*, 12(1):59–62, 2012. (Zitiert auf den Seiten 5, 15, 16, 71 und 72)

[Sof] I. Software. Schutz sensibler und persönlicher Daten für DB2- und IMS-Systeme. http://www-03.ibm.com/software/products/de/de/infoguardataencrfordb2andimsdata. (Zitiert auf Seite 9)

[SR09] W. Streitberger, A. Ruppel. *Cloud Computing Sicherheit: Schutzziele, Taxonomie, Marktübersicht*. Fraunhofer-Institut für Sichere Informationstechnologie SIT, 2009. (Zitiert auf den Seiten 26 und 27)

[SR12] A. G. u. H. T. Stefan Renner. Vergleich von Policy Sprachen zur Anwendung im Bereich des Cloud Computings. 2012. (Zitiert auf den Seiten 5, 9, 53 und 54)

[Tho11] A. Thor. Cloud Data Management. http://dbs.uni-leipzig.de/file/CDM_03_DataStores.pdf, 2011. (Zitiert auf Seite 18)

[TK12] U. V. S. V. Thomas Kunz, Annika Selzer. CloudCycle: Katalog funktionaler und nicht-funktionaler Sicherheit- und Compliance-Anforderung. 2012. (Zitiert auf den Seiten 9, 14, 21, 22, 23, 24, 25, 26, 27, 30, 31, 32, 33 und 34)

[Tor11] C. Tornau. *Analyse, Design und Implementierung einer aspektorientierten Erweiterung der Programmiersprache nesC im Besonderen fuer das Logging in Sensornetzen*. GRIN Verlag, 2011. (Zitiert auf Seite 23)

[VG10] M. Vehlow, C. Golkowsky. Cloud Computing-Navigation in der Wolke. *PricewaterhouseCoopers AG, Frankfurt am Main*, 2010. (Zitiert auf den Seiten 5, 8 und 28)

[VG11] M. Vehlow, C. Golkowsky. Cloud Computing im Mittelstand: Erfahrungen, Nutzen und Herausforderungen. *PricewaterhouseCoopers AG, Frankfurt am Main*, 2011. (Zitiert auf den Seiten 5, 7, 27 und 28)

[Vio12] G. Viola. Definition Cloud computing -Eigenschaften, Architektur und Modelle der Datenwolken. http://www.egovernment-computing.de/standards/articles/362644, 2012. (Zitiert auf den Seiten 9, 11 und 12)

[Wei12] A. Weiss. Merging of TOSCA Cloud Topology Templates. 2012. (Zitiert auf Seite 19)

[Xan10] M. Xander. Cloud Computing. Technische Grundlagen, Chancen und Probleme des Trends. S. 9, 2010. (Zitiert auf Seite 13)

Alle URLs wurden zuletzt am 20. 05. 2013 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift