Cloud-Applikationen in der Finanzindustrie — Dienstgütemerkmale und deren Überwachung

Ronny Hans, Melanie Holloway, The An Binh Nguyen, Alexander Müller, Marco Seliger, Jürgen Lang
Gemeinsames Whitepaper des Fachgebiets Multimedia Kommunikation (KOM) der Technischen Universität Darmstadt (TUD) und IBM

TECHNISCHE UNIVERSITÄT DARMSTADT

KOM – Multimedia Communications Lab

Cloud-Applikationen in der Finanzindustrie —
Dienstgütemerkmale und deren Überwachung

Ronny Hans, Melanie Holloway, The An Binh Nguyen, Alexander Müller, Marco Seliger, Jürgen Lang

Whitepaper

http://www.kom.tu-darmstadt.de

Erstveröffentlichung: 22.09.2017
Letzte Aktualisierung: 27. Dezember 2017

Technische Universität Darmstadt
Fachbereich für Elektrotechnik und Informationstechnik
Fachbereich für Informatik (Zweitmitglied)

Fachgebiet Multimedia Kommunikation (KOM)
Prof. Dr.-Ing. Ralf Steinmetz

Inhaltsverzeichnis

Abkürzungsverzeichnis

BaFin	Bundesanstalt für Finanzdienstleistungsaufsicht
BDSG	Bundesdatenschutzgesetz
BPaaS	Business Process as a Service
IaaS	Infrastructure as a Service
IT	Informationstechnologie
NIST	National Institute of Standards and Technology
PaaS	Platform as a Service
QoS	Quality of Service
SaaS	Software as a Service
SSL	Secure Sockets Layer
SLA	Service Level Agreement
TOM	technische und/oder organisatorische Maßnahmen
TTR	Time-to-Repair

1 Einführung

Die Informationstechnologie (IT) ist heutzutage ein zentraler Bestandteil der meisten Geschäftsprozesse innerhalb von Unternehmen. Vor allem im Finanzsektor gilt die IT als ein wesentlicher *Enabler*, der Banken erst in die Lage versetzt, Dienstleistungen für ihre Kunden zu erbringen. Zur Kehrseite der Informationstechnologie gehören allerdings die hohen Kosten diese bereitzustellen. Aufgrund des immer größer werden Wettbewerbs im Finanzsektor, sind unter anderem hohe Flexibilität, kurze Produktlebenszyklen und geringe Kosten unerlässlich um ein langfristiges Bestehen am Märkten zu gewährleisten. Dafür muss die IT-Infrastruktur in Unternehmen stets dynamisch den spezifischen Anforderungen angepasst werden können.

Hinsichtlich Kosten, Flexibilität und Skalierbarkeit ist Cloud Computing ein vielversprechendes Bereitstellungsmodell für IT-Dienstleistungen. Dieses Paradigma hat über die letzten 10 Jahre stark an Popularität gewonnen. Laut einer aktuellen Bitkom-Umfrage nutzt mittlerweile die Mehrheit der deutschen Unternehmen Cloud Computing [15]. Bei der Bereitstellung von IT-Dienstleistungen spielt die Zusicherung von Dienstgüteeigenschaften, wie Verfügbarkeit und Zuverlässigkeit, eine wichtige Rolle. Neben der Funktionalität definieren nicht-funktionalen Merkmale wesentlichen Charakteristika von IT-Dienstleistungen. Dies betrifft jede Art von Diensten, sowohl Infrastruktur- als auch Softwaredienste.

Die Finanzindustrie ist eine Branche, die zahlreichen und strengen gesetzlichen Regelungen unterliegt. Dazu gehörten unter anderem das *Bundesdatenschutzgesetz*, die *Mindestanforderungen an das Risikomanagement* und das *IT Sicherheitsgesetz*. Auch in diesem Kontext kommen den Dienstgüteparametern ein wichtige Rolle zu. Vertraglich geregelt sind Dienstgütereigenschaften in Service Level Agreement (SLA)s. Solche Verträge umfassen die Art der Bereitstellung, die Nutzungsbedingungen und nummerische Werte, anhand derer die Erfüllung der zugesicherten Dienstgütemerkmale gemessen werden kann. Fernen können darin Strafzahlungen für die Nichteinhaltung solcher Dienstgütemerkmale vereinbar sein. Besonders herausfordernd ist im Kontext von Cloud Computing die Überwachung der Einhaltung dieser Verträge.

Dieses Whitepaper ist wie folgt aufgebaut. Im Anschluss an diese Einführung werden in Kapitel 2 die Grundlagen und wesentliche Begriffe der traditionellen Diensterbringung und des Cloud Computing kurz dargestellt. Im sich anschließenden Kapitel 3 werden die nicht-funktionalen Anforderungen an Cloud-Dienste dargestellt. Schwerpunkt bilden dabei die Definition und Eigenschaften nicht-funktionaler Anforderungen, die Kategorisierung dieser Merkmale sowie die regulatorischen Anforderungen speziell für den Finanzsektor. In Kapitel 4 wird auf die Überwachung von SLAs eingegangen. Dabei wird die Überwachung traditioneller und Cloud-spezifischer SLAs untersucht. Ferner werden praxisrelevanten Ansätze und Tools vorgestellt. Dieses Whitepaper schließt in Kapitel 5 mit einer Zusammenfassung.

2 Grundlagen der Diensterbringung

2.1 Traditionelle Diensterbringung

Informationstechnologie wird sei jeher genutzt, um Geschäftsprozesse im Unternehmen zu rationalisieren. Dabei erhält die IT-Organisation ihre Anforderungen aus den Fachabteilungen, setzt Lösungsprojekte auf und nimmt die Lösungen in Betrieb.

Die IT ist traditionell häufig anhand der eingesetzten Technologiedomänen organisiert. Es gibt Abteilungen für Mainframe, verteilte Server, Speicher, Middleware, Datenbanken, Netzwerk, Arbeitsplatzbetreuung, etc. Dieses Vorgehen hat den Vorteil, dass es klare Verantwortlichkeiten für die Verfügbarkeit von Systemen, Budgets und Mitarbeitern gibt. Im Sinne der Effizienzsteigerung hat jede Abteilung ihre Abläufe für sich optimiert. Jedoch hat diese Organisationsform auch ihre Schwachstellen. Neben redundanten Aktivitäten rund um Basisaufgaben, wie die Behebung einfacher Störungen oder die Umsetzung einfacher Änderungen, fehlt vor allem ein Gesamtverständnis der Zusammenhänge. Müssen beispielsweise Performance-Engpässe in Anwendungssystemen analysiert werden, sind nicht selten diverse Spezialisten quer über alle Infrastrukturabteilungen im Einsatz. Darüber hinaus macht der fehlende zentrale Einstiegspunkt in die IT-Infrastruktur es nicht leicht für Fachabteilungen oder Anwendungsentwickler, ihre Anforderungen umzusetzen [70]. Jede einzelne Infrastrukturabteilungen stellt ihre Technologien auf Anforderung der Entwicklungsprojekte bereit, häufig ohne eine übergeordenete Koordination oder Abstimmung der Abteilungen untereinander. Das führt zu einer lokalen Optimierung einzelner Lösungsprojekte, geht aber zulasten einer ganzheitlichen Optimierung in der IT.

In den letzten Jahren werden verstärkt geeignete Governance-Mechanismen wie Enterprise Architecture Planungen und Technologie-Roadmaps aufgesetzt, um einen besseren Standardisierungsgrad zu erreichen. Das Etablieren solcher Governance-Mechanismen ist ein notwendiger Schritt, in der Praxis ist jedoch erkennbar, dass nach wie vor eine große Anzahl von Ausnahmen existiert. Standardisierung wird oftmals in den Fachabteilungen nicht akzeptiert [24]. Solange die IT-Abteilungen nicht wie Industrieunternehmen sondern eher wie Handwerksbetriebe oder Manufakturen agieren, ist es relativ leicht, Sonderwünsche aus Fach-Abteilungen zu erfüllen. Dabei geht es nicht um die Automatisierung einzelner Arbeitsschritte, beispielweise durch Scripte, sondern um die Steuerung und Durchführung der Prozesse.

Der Trend zur Digitalisierung bringt neue Anforderungen an die IT mit. Neue Lösungen müssen immer schneller entwickelt und bereitgestellt werden. Die heutigen IT-Organisationen sind in ihrer bisherigen Rolle überwiegend reaktive Dienstleister, die aufgrund ihrer Strukuren, Prozesse und Fähigkeiten kaum in der Lage sind, den neuen Anforderungen gerecht zu werden [53]. Das ist nur zu erreichen, wenn sich die IT-Organisation stärker an den Prinzipien der industriellen Fertigung ausrichtet. Hierzu wird mit den Kunden einerseits ein definiertes, auf Standards basierendes Service-Portfolio abgestimmt. Andererseits ist zu entscheiden, welche der Dienste mit hohem Automationsgrad eigenständig produziert werden und welche als Halbfertigprodukt vom Markt eingekauft und ins eigene Portfolio integriert werden. Dabei ist Hybrid Cloud eine Strategie, um flexibel über die eigene Fertigungstiefe entscheiden zu können und trotzdem die Anforderungen der IT-Kunden zu erfüllen.

2.2 Cloud-basierte Diensterbringung

Der Begriff des Cloud Computing gehört zu den neueren Paradigmen der Technikgeschichte, die zugrundeliegenden Basistechnologien sind jedoch älter. Dazu gehören bereits bekannte Technologien, wie beispielsweise Virtualisierung oder die Nutzung von Web-Diensten [95]. Durch die vielfältigen Einsatzmöglichkeiten von Cloud Computing existieren mittlerweile zahlreiche Definitionen. Eine, die sich als De-Facto-Standard durchgesetzt hat, ist die des National Institute of Standards and Technology (NIST)

[77]. Diese wird im folgenden Abschnitt näher erläutert.

2.2.1 NIST-Definition von Cloud Computing

Die NIST-Definition charakterisiert Cloud Computing anhand von fünf Eigenschaften [72]. So können Cloud-Dienste dem Nutzer bedarfsgerecht zur Verfügung und in Rechnung gestellt werden. Weiterhin ist es bei der Nutzung von Cloud-Diensten möglich, Ressourcen (bspw. Rechenleistung oder Speicher) automatisch und ohne persönliche Interaktion mit dem Dienstanbieter zu beziehen [31, 72]. Bereitgestellte Dienste können über das Internet auf nahezu jedem beliebigen Endgerät genutzt werden. Durch diesen standardisierten Weg erhält der Nutzer einen weltweiten Zugang zu seinen Daten oder Ressourcen [72]. Nachteilig erweist sich dabei, dass die Ressourcen in einem Pool vorliegen und es dem Nutzer damit nicht möglich ist zu erkennen, wo sich gespeicherte Daten oder Ressourcen geografisch befinden und wo diese verarbeitet werden [66, 72].

Abbildung 2.1 zeigt eine Darstellung der in der NIST-Definition genannten Dienstmodelle. Die Abbildung zeigt, dass die Dienstmodelle einerseits aufeinander aufbauen und andererseits hinsichtlich ihres Abstraktionsniveaus und ihrer Einsatzflexibilität unterschiedlich stark ausgeprägt sind [87].

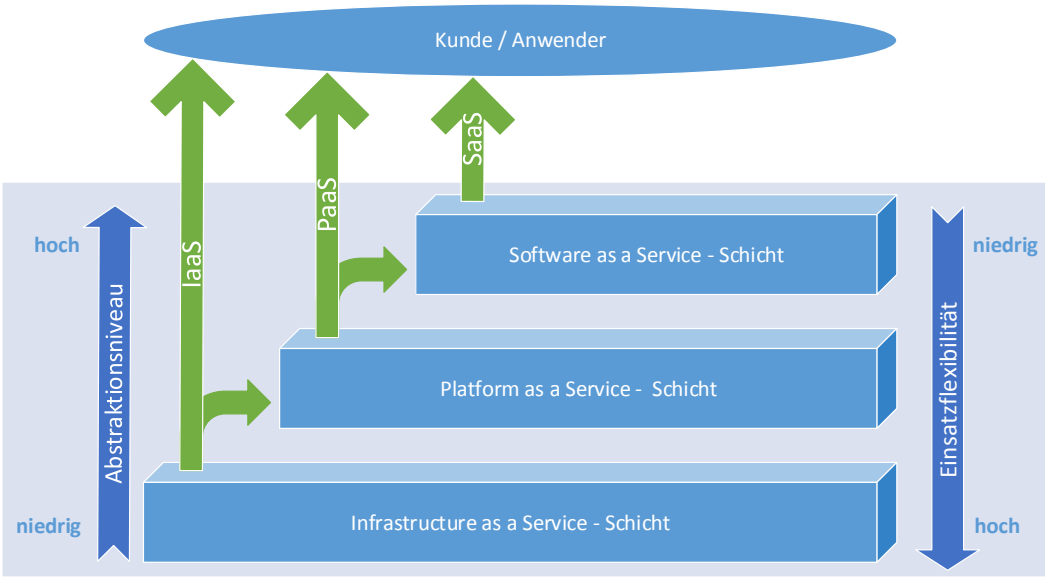

Abbildung 2.1: Abstraktionsniveau und Einsatzflexibilität der Dienstmodelle des Cloud Computings in Anlehnung an Ruppel und Stephanow [87]

Dienstmodelle im Cloud Computing

Die NIST-Definition unterscheidet die drei Dienstmodelle Infrastructure as a Service (IaaS), Platform as a Service (PaaS) und Software as a Service (SaaS). Diese Modelle, die auch Ebenen genannt werden, sind wie folgt definiert:

- **Infrastructure as a Service (IaaS)**: Dieses Dienstmodell stellt dem Nutzer grundlegende Ressourcen zur Verfügung [72], z.B. Rechenleistung, Arbeitsspeicher und Festplattenspeicher [97]. Generell kann es als Zugang zu *reiner* bzw. low-level Computerinfrastruktur und der darin arbeitenden

virtuellen Maschinen bezeichnet werden [31, 79]. Aktuell gibt es eine Vielzahl von IaaS-Anbietern, exemplarisch seien hier Amazon mit der *Elastic Computing Cloud (EC2)* [2] oder IBM mit *Softlayer*[40] genannt.

- **Platform as a Service (PaaS)**: Abstrahiert man von der IaaS-Ebene, das heißt, man gibt die direkte Kontrolle über diese Ressourcen ab, spricht man von einer Plattform [17, 72]. Diese Plattform bietet eine vorkonfigurierte Programmierumgebung zur Unterstützung der Applikationsentwicklung und der Ausführung von Software [30]. Auch hier gibt es mittlerweile eine Vielzahl von Anbietern, z.B. doe Google *App Engine*[36] und IBM *Bluemix Hybrid Cloud Development Platform*[41].

- **Software as a Service (SaaS)**: Abstrahiert man weiter und beschränkt sich auf die Nutzung von Software, spricht man von SaaS. Dieses Bereitstellungsmodell bietet die Möglichkeit, Programme in der Cloud zu nutzen. SaaS kann somit als direkte Schnittstelle zum Endverbraucher angesehen werden [63]. Als Darstellungsmöglichkeit bietet sich der Webbrowser an [18, 97]. Klassische Vertreter von SaaS sind beispielsweise Google mit der Office-Anwendung *Google Docs* [37] oder *Kenaxa* von IBM [42].

Die dargestellte Unterteilung des NIST beschränkt sich auf drei Ebenen. In der Praxis lässt sich allerdings erkennen, dass es zusätzlich hybride layerübergreifende Angebote gibt, die nicht zweifelsfrei einer Ebene zugeordnet werden können. Ein prominentes Beispiel ist “Database as a Service” [57].

Im Laufe der Zeit hat sich durch die enorme Anzahl von “*as a Service* Bereitstellungen” der Begriff *XaaS* etabliert [4]. Für den späteren Verlauf des Whitepapers ist Business Process as a Service (BPaaS) relevant und sei hier deswegen schon als weiterer Vertreter von *XaaS* eingeführt. Bei BPaaS wird ein ganzer Geschäftsprozess durch einen Dienstleister angeboten [107].

Bereitstellungsmodelle im Cloud Computing

Für die genannten Dienstmodelle differenziert die NIST-Definition vier unterschiedliche Bereitstellungsmodelle *Private Cloud*, *Public Cloud*, *Hybrid Cloud* und *Community Cloud*. Diese werden im Folgenden näher erläutert:

- **Private Cloud**: Dieses Bereitstellungsmodell (auch *Exklusive/Enterprise Cloud* genannt) ist nur einer einzigen Organisation zugänglich [14]. Es hat damit große Ähnlichkeiten zu einem internen Rechenzentrum [4]. Dies bedeutet, dass die Private Cloud der Öffentlichkeit nicht zugänglich und der Nutzerkreis dem Unternehmen vollumfänglich bekannt ist [72, 91]. Die Art der Bereitstellung einer Private Cloud kann jedoch variieren. So kann der Besitz, die Verwaltung und die Ausführung von der Organisation selbst durchgeführt werden oder ganz/teilweise an eine dritte Partei ausgelagert werden (vgl. Abbildung 2.2) [72].

- **Public Cloud**: Dieses Bereitstellungsmodell ist eine offene und der Allgemeinheit zugängliche Cloud. Sie kann damit in der Regel keinem Unternehmen singulär zugeordnet werden [4, 7]. Somit sind in den meisten Fällen keine speziellen Vertragsverhandlungen zum Zutritt notwendig [7].

- **Hybrid Cloud**: Dieses Bereitstellungsmodell ist eine Kombination aus Private und Public Cloud [13]. Diese beiden Cloud-Typen werden über eine standardisierte Schnittstelle verbunden, um bspw. Lastspitzen in der Private Cloud mit der Public Cloud abzufangen [7]. Damit gehen die Vor- als auch die Nachteile beider Modelle ineinander über (siehe Kapitel 2.2.2).

- **Community Cloud**: Dieses Bereitstellungsmodell bezeichnet eine Cloud, welche exklusiv für eine Community angeboten wird. Als Community sind hier Unternehmen anzusehen, die ähnliche

Interessen haben. Bereitgestellt und verwaltet wird die Community Cloud dabei von einem Unternehmen der Community oder von einem externen Dienstleister [72]. Ein populäres Beispiel für solch eine Cloud ist die *Frankfurt Cloud*[1], welche die Infrastruktur der „Forschungs-Community“ bereitstellt.

Eine anbieterabhängige Darstellungsform der genannten Bereitstellungs- und Dienstmodelle zeigt Abbildung 2.2. Bezüglich der Bereitstellungsmodelle zeigt diese Grafik den Übergang einer privaten On-Premise Cloud hin zur einer gehosteten Public Cloud, hinsichtlich der Dienstmodelle den Übergang von IaaS hin zu BPaaS.

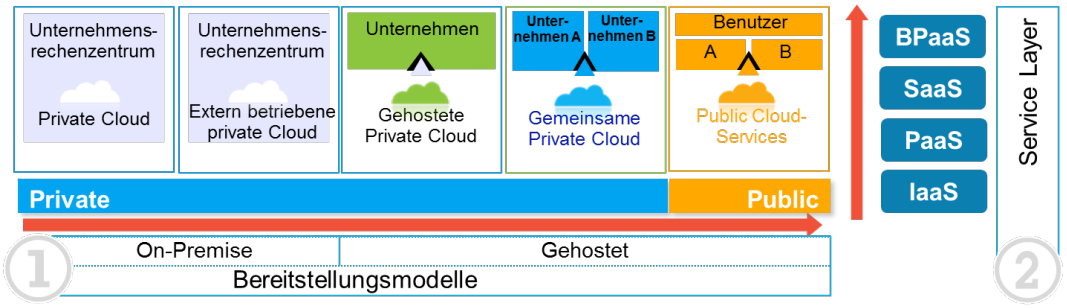

Abbildung 2.2: Bereitstellungsmodelle und Dienstebenen (Quelle IBM)

2.2.2 Vorteile und Risiken von Cloud Computing

Hinsichtlich der Nutzerseite bietet Cloud Computing die Möglichkeit, Fixkosten zu variabilisieren und somit die Investitionskosten zu minimieren.

Auf Nutzerseite verringert sich durch den Einsatz von Cloud Computing das Investitionsrisiko, da bei einigen Preismodellen nur die tatsächlich verbrauchte Leistung vom Cloud-Anbieter in Rechnung gestellt wird [101]. Nähere Informationen zu den Preismodellen und den Kosten allgemein finden sich im Abschnitt 3.2.2. Für den Anbieter ergeben sich realisierbarere Skaleneffekte, aber auch organisatorische Veränderungen, wie das globale Einspielen von Updates bei SaaS-Produkten [13, 101].

Neben den positiven Aspekten lassen sich aber auch potenzielle Nachteile beim Cloud Computing erkennen. So unterliegt eine Public Cloud in der Regel keinem geschlossenem Nutzerkreis. Da die Nutzer von Public Cloud-Lösungen keinen Einfluss auf Handlungsabläufe, Sicherheitsstandards oder auf die Vertragsgestaltung haben, können sich Risiken oder Complianceprobleme zum Beispiel beim Auslagern von personenbezogenen Daten in Drittländer mit geringerem Datenschutzniveau ergeben [14, 56, 80]. Bei der Nutzung von Private Cloud-Lösungen können dem Nutzer durch die potenziell geringere Auslastung der Hardware Skalenerträge entgehen, welche innerhalb einer Public Cloud durch eine höhere Auslastung durch verschiedene Nutzer realisiert werden können [5]. Die hier genannten Punkte der Public Cloud gelten analog auch für die Hybrid Cloud, da diese eine Schnittstelle zur Public Cloud bietet. Damit verschmelzen jeweils die Vor- und Nachteile der Public und Private Cloud für das Mischmodell Hybrid Cloud.

[1] http://www.frankfurt-cloud.com/about-us/

2.3 Fazit

In diesem Kapitel wurden die theoretischen Grundlagen des Cloud Computing näher erläutert. Einhergehend mit der zunehmenden Bereitstellung dieser Dienste, nimmt die ebenfalls die Heterogenität von funktionalen und nicht-funktionalen Anforderungen weiter zu. Die wesentlichen nicht-funktionalen Anforderungen werden in Kapitel 3 ausführlich dargestellt.

3 Nicht-funktionale Anforderungen an Cloud-Dienste

Wie bereits im vergangenen Kapitel beschrieben, ist Cloud Computing ein neueres Bereitstellungsparadigma. Die zugrundeliegende Technik hat sich in den letzten Jahren weiter entwickelt. So existieren Web-Dienste[1] seit nunmehr 15 Jahren in der IT und sind dabei vor allem hinsichtlich des breiten Anwendungsspektrums interessant [81]. An die bereitgestellten Dienste werden verschiedene Anforderungen gestellt, die sich in zwei Arten unterteilen lassen:

- *Funktionale Anforderungen* beschreiben den Zweck eines Dienstes im Allgemeinen. Es wird damit festgelegt, welche Aufgaben der Dienst erfüllen muss [43, 83].
- *Nicht-funktionalen Anforderungen* beschreiben einen Dienst hinsichtlich seiner Merkmale bei der zu erfüllenden Aufgabe [84]. Sie sind also eine Beschreibung, *wie* der Dienst auszuführen ist.

Schwerpunkt dieses Kapitels sind die nicht-funktionalen Anforderungen. Zu Beginn dieses Kapitels erfolgt zunächst eine Begriffsdefinition. Anschließend werden Eigenschaften von nicht-funktionalen Anforderungen definiert und deren Attribute im Folgenden kategorisiert. Im weiteren Verlauf des Kapitels wird die Dienstgüte von Cloud-Diensten untersucht. Abschließend werden die regulatorischen Anforderungen hinsichtlich der Diensterbringung dargestellt. Das Kapitel schließt mit einem Fazit.

3.1 Definition und Eigenschaften nicht-funktionaler Anforderungen

Eine De-Facto-Standard-Definition, ähnlich der NIST-Definition von Cloud Computing, lässt sich basierend auf der wissenschaftlichen Literatur für nicht-funktionale Anforderungen nicht benennen. Glinz gibt jedoch eine Übersicht, die 13 verschiedene Definitionen beinhaltet, und fasst diese in einer eigenen Definition wie folgt zusammen [35]:

Eine nicht-funktionale Anforderung ist ein Attribut oder eine Randbedingung eines Systems. Ein Attribut kann sich wiederum entweder auf die Performanz (bspw. den Durchsatz) oder auf die Qualität (bspw. auf die Verfügbarkeit) des Systems beziehen. Hingegen beziehen sich Randbedingungen bspw. auf physikalische oder gesetzliche Vorschriften, die von dem System zu erfüllen sind.

Basierend auf der Arbeit von Glinz [35] und einer Literaturrecherche konnten folgende zwei Haupt-Charakteristika von nicht-funktionalen Anforderungen abgeleitet werden:

- „Must-have“ Charakteristika beschreiben nicht-funktionale Anforderungen durch die reine Benennung von Eigenschaften. Es wird somit definiert, welche Eigenschaften das System haben muss. Beispiele hierfür sind Benutzerfreundlichkeit, Zuverlässigkeit oder Performanz [35, 47]. Ein mögliches Rahmenwerk zur Festlegung von nicht-funktionalen Anforderungen ist in ISO/IEC 9126 gegeben [47]. Neben der Benennung dieser Attribute ist eine Beschreibung der Dienstgüte ratsam, um zu quantifiziert inwieweit die genannten Attribute erfüllt werden [35, 98].

- „Don’t have“ Charakteristika zeichnen sich durch die Benennung von Beschränkungen aus. So benennen einige Definitionen [44, 46] Entwurfsbeschränkungen und physische Beschränkungen, andere [84] operationale oder rechtliche Aspekte [35]. Gerade compliance-kritische Bereiche unterliegen hohen Anforderungen und so kann es ratsam sein, Beschränkungen, wie beispielsweise ein Ausfuhrverbot von personenbezogenen Daten an sichere Drittländer, zu definieren[105].

[1] engl. Web services

Ferner lassen sich Eigenschaften hinsichtlich ihrer Quanifizierbarkeit unterscheiden. Quantitative Eigenschaften können dabei in messbaren Größen dargestellt werden. Beispiele hierfür sind Performanz oder Verfügbarkeit. Im Gegensatz dazu können Eigenschaften auch qualitativ Natur sein. Das heißt, dass sie keine direkt messbare Einheit besitzen. Sie können beispielsweise mittels Nutzerbefragungen bewertet werden. Beispiele hierfür sind die Reputation und die Sicherheit von Diensten.

Hinsichtlich ihrer Dynamik lassen sich Eigenschaften weiterhin wie folgt unterscheiden:

- Die betrachteten Eigenschaften können *statisch* sein, das heißt, dass diese sich bei der Ausführung eines Dienstes nicht ändern (z.B. Vertraulichkeit eines Dienstes).
- Eigenschaften können aber auch *dynamisch* sein. Dynamische Eigenschaften können sich zur Laufzeit beispielsweise durch hohe Auslastung verändern, was zu höhere Latenz führen kann [104].Eine Herausforderung solcher dynamischer Parameter besteht in deren Überwachung, was fortlaufend geschehen muss.

Hat der Nutzer sich für einen Dienstanbieter entschieden, ist es in der Regel erforderlich, die gewünschten Eigenschaften in Form von einer Dienstgütevereinbarung, einem sogenannten SLA, schriftlich festzuhalten. Ein SLA beschreibt ein Vertragskonstrukt zwischen einem Dienstanbieter und einem Nutzer. Es werden Leistungen festgelegt die der Dienstanbieter dem Nutzer erbringen muss. Dies können beispielsweise Garantien zur Verfügbarkeit oder zu Latenzzeiten sein [10]. In diesem Zusammenhang wird von der Dienstgüte, der sogenannten Quality of Service (QoS), gesprochen. Zur Quantifizierung der Qualitätsanforderungen werden Kennzahlen in Kombination mit Dienstgüteniveaus verwendet. Ein Dienstgüteniveau ist durch einen Wert gekennzeichnet, welcher eine bestimmte Qualität eines Dienstes widerspiegelt. Grundsätzlich werden in der Literatur die folgenden drei Dienstgüteklassen unterschieden:

- *Garantierte Dienste*[2]*:* Kennzeichnen sich durch feste Zusagen bezüglich der Dienstgüte [88, 99].
- *Best mögliche Dienste*[3]*:* Kennzeichnen sich durch das Versprechen, den Dienst bestmöglich auszuführen. Dies impliziert aber, dass oft keine direkten quantitativen Zusagen, wie statische oder dynamische Schwellenwerte, gemacht werden [88, 99].
- *Vorhersagbare Dienste*[4]*:* Kennzeichnen sich durch deren Verhalten in der Vergangenheit [99].

Zusätzlich beinhaltet ein SLA Strafen, die bei einer Nichteinhaltung der Qualitätsanforderungen durch den Anbieter fällig werden [76]. Somit haben Nutzer und Anbieter ein Interesse daran, dass die Qualitätsanforderungen eingehalten werden [73]. Die Nutzer erhalten die Dienste in erwünschter Qualität und die Anbieter ersparen sich zusätzliche Kosten. Im Rahmen dieser Veröffentlichungen liegt der Fokus auf den nicht-funktionalen Anforderungen.

Nachdem in diesem Abschnitt eine Übersicht über Definitionen und igenschaften von nicht-funktionalen Anforderungen gegeben wurde, wird im folgenden Abschnitt die nicht-funktionalen Anforderungen detalliert untersucht und kategorisiert.

3.2 Kategorisierung der nicht-funktionalen Anforderungen

In diesem Abschnitt erfolgt eine Kategorisierung der identifizierten nicht-funktionalen Anforderungen. Generell umfassen Cloud-Dienste ein weites Spektrum an bereitgestellten Ressourcen und Dienstleistungen, so dass der Begriff des Cloud-Dienstes sehr abstrakt. Ferner – im Gegensatz zu traditionellen Geschäftsmodellen mit einmaliger Zahlung – können Cloud-Dienste nutzungsabhängig abgerechnet werden und die Dienste lassen sich entsprechend des Bedarfs dynamisch anpassen.

[2] engl. guaranteed services
[3] engl. best effort services
[4] engl. predictable services

Diese, als Elastizität bezeichnete Eigenschaft, stellt ein Cloud-spezifisches Dienstgütemerkmal dar [72]. Elastizität wird dabei als ein Maß verwendet, um die dynamische Veränderbarkeit eines Cloud-Dienstes zu erfassen [25]. Wie bereits eingangs dargestellt wurde, unterscheidet sich Cloud Computing von anderen Paradigmen hinsichtlich dessen Dienstbereitstellungs- und Abrechnungsmodells. Die Elastizität ist dabei eine von mehreren, neuen Metriken, die dazu genutzt werden kann, Cloud-Dienste untereinander vergleichbar zu machen. SMICloud stellt in diesem Zusammenhang ein Rahmenwerk mit Fokus auf standardisierte Kennzahlen bereit, um eine ganzheitliche Sicht über die Dienstgüte von Cloud-basierten Diensten zu erhalten und einen Vergleich unterschiedlicher Anbieter zu ermöglichen [33]. Auf Basis dieses Rahmenwerks wurde in Anlehnung an ein bestehendes Kategorisierungsschema für traditionelle Dienste (vgl. [9]) ein Kategorisierungsschema für Cloud-Dienste abgeleitet. Abbildung 3.1 zeigt das erarbeitete Kategorisierungsschema. Wie in der Abbildung zu erkennen ist, wird zwischen den Dienstgüteeigenschaften und den Kosten unterschieden. Dementsprechend widmen sich die folgenden Abschnitte diesen beiden Themen.

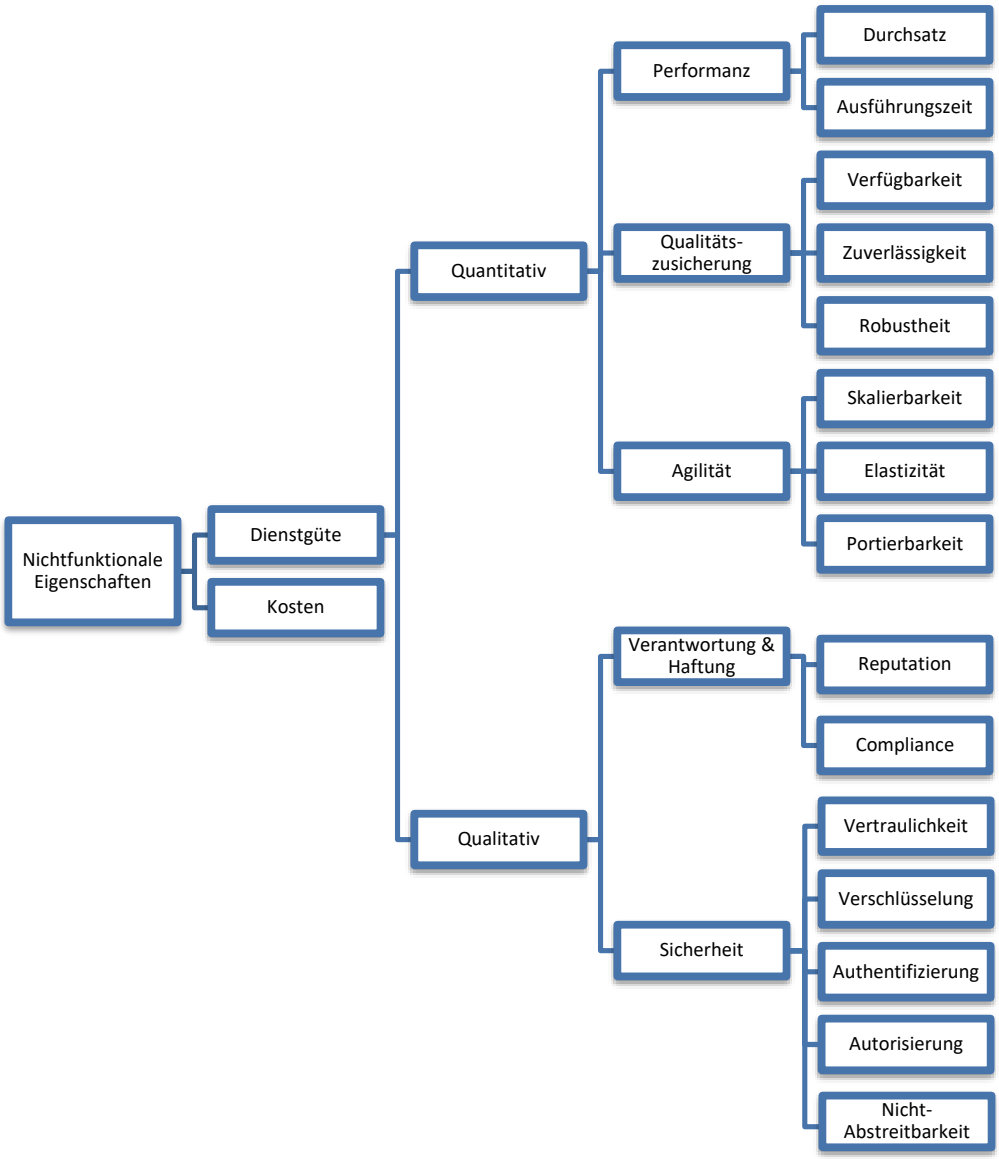

Abbildung 3.1: Kategorisierung von nicht-funktionalen Anforderungen in Anlehnung an [9, 33]

3.2.1 Dienstgütemerkmale

Hinsichtlich der Dienstgüte kann allgmein zwischen quantitativen und qualitativen Merkmalen unterschieden werden. Im Unterschied zu den qualitativen Merkmalen lassen sich quantitative Merkmale in messbaren Größen darstellen. Im Folgenden werden zuerst die quantitativen Merkmale betrachtet, gefolgt von den qualitativen.

Quantitative Eigenschaften

Quantitative Merkmale sind, wie bereits vorrangig beschrieben, dadurch gekennzeichnet, dass sie in numerisch messbaren Größen darstellbar sind. Der folgende Abschnitt beschreibt deren Eigenschaften und gibt mögliche Metriken zur Messung an.

Performanz
Die Performanz als Eigenschaft ergibt sich aus zwei Aspekten, dem Durchsatz und der Ausführungszeit. Diese definieren sich wie folgt:

- *Durchsatz*: Allgemein kann man dabei den Durchsatz als Anzahl von Anfragen, die in einem bestimmten Zeitintervall verarbeitet werden können, beschreiben [9, 75].

- *Ausführungszeit*: Die Ausführungszeit beschreibt die benötigte Zeitspanne eines Dienstes vom Absenden einer Anfrage bis zum Erhalten der Antwort [106]. Dies impliziert folgende Gleichung:

$$Ausführungszeit = t_{\text{Client}} + t_{\text{Netzwerklatenz}} + t_{\text{Server}} [21] \tag{3.1}$$

  Mit:

  t_{Client} — Zeitspanne, die der Client benötigt, um die Anfrage nach der Aktion des Nutzers (Mausbewegung etc.) zu senden und die ankommende Antwort des Servers dem Nutzer anzuzeigen [21].

  $t_{\text{Netzwerklatenz}}$ — Zeitspanne, für den Datentransport zwischen Client-Rechner und Server des Anbieters (Hin- und Rückweg) [21].

  t_{Server} — Zeitspanne zwischen dem Erhalt der Anfrage auf Serverseite und dem Senden der Antwort [21].

  Lampe et al. bezeichnet diese, vom Nutzer wahrgenommene Ausführungszeit als *user-perceived latency* [59]. Es sei angemerkt, dass weitere Quellen die Performanz und vor allem die *Ausführungszeit* averschiedenartig definieren. Beispielsweise definiert Berbner [9] die Ausführungszeit als Zeitspanne, die ein Dienst zum Bearbeiten der Anfrage benötigt. Dies impliziert eine Betrachtung aus Sicht des Dienstanbieters, im Gegensatz zu dieser Definition, die die gesamte Zeitspanne betrachtet und somit eine Beschreibung aus Sicht des Nutzers darstellt.

Qualitätszusicherung
Mittels der Qualitätszusicherung wird eine Metrik angegeben, welche eine Qualität benennt und diese in messbaren Eigenschaften beschreibt. Diese Kategorie lässt sich wie folgt unterteilen:

- **Verfügbarkeit**: Als Verfügbarkeit wird jener Zeitanteil bezeichnet, in dem der Dienst sofort benutzet werden kann. Diese Nutzung impliziert eine funktionstüchtige Ausführung. Die Verfügbarkeit wird in der Regel wie folgt in Prozent angegeben [38, 69, 33]:

$$Verfügbarkeit = \frac{t_{Gesamt} - t_{\text{Nicht-Verfügbar}}}{t_{Gesamt}} \tag{3.2}$$

  In Bezug auf weitere Eigenschaften, die hier benannt werden, hat die Verfügbarkeit eine enge Verwandtschaft mit der Zuverlässigkeit. Die Time-to-Repair Time-to-Repair (TTR) beschreibt ferner die Zeitspanne, welche notwendig ist, um einen Dienst nach dessen Ausfall wieder verfügbar zu machen [9].
- **Zuverlässigkeit**: Die Zuverlässigkeit wird in der Regel in Form eines Intervalls angegeben, in welchem ankommende Anfragen angenommen und *korrekte* Anfragen abgegeben werden [69, 75]. Fehler sind dabei ein Indikator für mangelnde Zuverlässigkeit [9]. Dies wird durch die folgende Formel beschrieben:

$$Zuverlässigkeit = \frac{\text{Anzahl korrekter Antworten}}{\text{Anzahl aller Antworten}} \tag{3.3}$$

- **Robustheit**: Durch ungültige, fehlerhafte oder unvollständige Anfragen sollte ein Web-Dienst weiterhin deterministisch und damit weiterhin funktionsfähig reagieren [9, 81]. Gleichung 3.4 gibt eine Metrik für die Robustheit eines Dienstes an [48, 64].

$$Robustheit = \frac{\text{Anzahl tolerierter Fehler}}{\text{Anzahl aller Fehler}} \tag{3.4}$$

Agilität

Diese Kategorie ist ebenfalls ein übergeordnetes Kriterium, welches sich wie folgt unterteilen lässt:

- **Skalierbarkeit**: Kann man die Kapazität unter Berücksichtigung der erfolgreichen Verarbeitung erhöhen oder verringern, spricht man von der Skalierbarkeit eines Dienstes. Dabei muss der Dienstanbieter in der Lage sein die relevanten Ressourcen erweitern und dynamisch verteilen zu können [9, 81].
- **Elastizität**: Die Elastizität ermöglicht dem Dienstanbieter das dynamische Zu- und Abschalten von Ressourcen in Abhängigkeit vom Bedarf des Nutzers [33, 64]. Sie ist dabei durch zwei Eigenschaften bestimmt, nämlich die maximale Kapazität und die durchschnittliche Dauer $\overline{t}_{Skal}$ eines Skalierungsvorgangs:

$$Kapazität_{max} = \text{Maximale Anzahl an verfügbaren Recheneinheiten unter Spitzenlast} \tag{3.5}$$

$$\overline{t}_{Skal} = \frac{1}{n}\sum_{i=1}^{n} T_{\text{Skalierungsausführung},i} - T_{\text{Skalierungsanfrage},i} \tag{3.6}$$

  Mit:
  $T_{\text{Skalierungsanfrage},i}$ = Zeitpunkt der Skalierung von Anfrage i
  $T_{\text{Skalierungsausführung},i}$ = Zeitpunkt der Skalierungsausführung von Anfrage i
  n = Anzahl der Skalierungsanfragen

- **Portierbarkeit**: Die Portierbarkeit, auch als Interoperabilität (Kompatibilität) bezeichnet, ist ein Maß für die Robustheit eines Dienstes bezüglich unterschiedlicher Standards [69]. Als mögliche Metrik kann Gleichung 3.7 aus [33] genutzt werden:

$$Interoperabilität = \frac{\text{Anzahl Plattformen, die vom Anbieter angeboten werden}}{\text{Anzahl Plattformen, die vom Nutzer zwecks Kompatibilität benötigt werden}} \tag{3.7}$$

Qualitative Eigenschaften

Die qualitativen Eigenschaften lassen sich ebenfalls in verschiedene Kategorien unterteilen. Dabei unterscheiden sich die qualitativen Eigenschaften zu den quantitativen darin, dass es keine direkte numerische Metrik zur Messung dieser gibt. Hierzu müssen entweder die Nutzer befragt werden (z.B. hinsichtlich Reputation) oder es müssen Frameworks zur Messung definiert werden (z.B. bei der Sicherheit). Andere Eigenschaften, wie beispielsweise die Haftbarkeit oder die Nicht-Abstreitbarkeit sind binäre Eigenschaften, das heißt, diese gelten und können garantiert werden, oder sie können nicht garantiert werden. Im Detail lassen sich die qualitativen Eigenschaften wie folgt unterschieden:

Verantwortung und Haftung
Agiert ein Anbieter auf einem Markt, ergibt sich durch sein Handeln eine Bewertung durch die Nutzer. Diese Einschätzung der Nutzer wird im Folgendem als *Reputation* des Anbieters bezeichnet. So ist die Reputation der einzige nicht-funktionale Aspekt, welcher nicht direkt vom Anbieter selbst angegeben werden kann. Sie bildet sich aus Referenzen und Erfahrungen der Vergangenheit [9]. In der Regel wird von einer höheren Bewertung ausgegangen, wenn mit dem Dienst in der Vergangenheit positive Erfahrung gemacht wurde [9, 48].

Sollten Fehler seitens des Dienstanbieters bei der Verarbeitung auftreten, sollte es möglich sein, diesen dafür haftbar zu machen. Um die Haftbarkeit sicherzustellen, müssen Rückverfolgbarkeit, Prüfbarkeit und Nicht-Abstreitbarkeit sichergestellt werden können [9]. Hier sei auf den folgenden Abschnitt *Sicherheit* verwiesen. Weiterhin fällt unter der Verantwortung und Haftung der Punkt Compliance. Allgemein wird darunter das regelkomforme Verhalten beschrieben. Da Abschnitt 3.3 sich explizit auf die regulatorische Anforderungen bezieht, sei an dieser Stelle auf diesen Abschnitt verwiesen.

Sicherheit
Nutzt und verarbeitet ein Dienstanbieter Daten jeglicher Art eines Nutzers, hat jener Nutzer in der Regel einen Wunsch hinsichtlich einer sicheren Verarbeitung. Dies impliziert eine Verwendung der folgenden Mechanismen [9, 27, 48, 75]:

- **Vertraulichkeit**: Der Zustand der Vertraulichkeit garantiert, dass Daten oder Informationen unautorisierten Dritten unzugänglich sind [52]. Vertrauliche Daten sind somit nur einer begrenzten Zahl von Nutzer zugänglich.
  - Zur Sicherstellung der Vertraulichkeit werden in der Regel kryptografische Verfahren und Zugriffskontrollen eingesetzt [8, 9, 101].
  - Die Eigenschaft der Vertraulichkeit wird oft im Rahmen des sogenannten *CIA-Dreieck* genannt. So bilden Vertraulichkeit (Confidentiality), Vollständigkeit (Integrity) und Verfügbarkeit (Availability) die Kanten dieses Dreiecks und stellen dabei die Oberbegriffe von IT-Schutzzielen dar [8].
- **Verschlüsselung**: Werden Daten so verändert, dass deren Inhalt und die Bedeutung von nichtgewollten Empfängern unverständlich sind, spricht man von einer *Verschlüsselung* einer Nachricht [100].
  - Die Verschlüsselung erlaubt es daher nur beabsichtigten Empfängern die Daten des Dienstes zu entschlüsseln und zu verarbeiten [9].
  - Neben der Verschlüsselung der Daten kann auch die Datenübertragung verschlüsselt werden, beispielsweise mit Secure Sockets Layer (SSL) [49].
  - Es existieren eine Vielzahl weiterer Verfahren, die es bspw. erlauben, verschlüsselte Daten ohne Entschlüsselung zu verarbeiten (*Voll homomorphe Verschlüsselung* [34]).

- **Authentifizierung**: Können die Identität des Nutzers oder eines Systems seitens des Dienstanbieters überprüft und identifiziert werden, spricht man von einer erfolgreichen Authentifizierung [8, 9].
- **Autorisierung**: Wurde der Nutzer oder das System durch eine erfolgreiche Authentifizierung identifiziert, ermöglicht es die Autorisierung dem Nutzer definierte Rechte zu garantieren [8, 9].
- **Nicht-Abstreitbarkeit**: Kann ein Nutzer, der beispielsweise einen Web-Dienst aufruft und nutzt, aufgrund einer eindeutigen Protokollierung dies nicht mehr abstreiten, spricht man von der Nicht-Abstreitbarkeit eines Dienstes [9].

3.2.2 Kosten

Generell handelt es sich bei Software um ein digitales Gut [39], bei welchem traditionelle Preismodelle nicht effizient genutzt werden können [86]. Dies betrifft auch Software, die in Form von Cloud-Diensten angeboten wird. Cloud-Provider stehen vor der Herausforderung, bestehende Preismodelle geeignet zu adaptieren, um Einnahmen und Gewinne zu sichern [61].

Im Folgenden werden verschiedene Preismodelle vorgestellt und bewertet. Ziel dieser Modelle ist die faire Abbildung vom Wert der Software für den Kunden und den Bereitstellungskosten des Anbieters [61]. Tabelle 3.1 zeigt die Komponenten eines Preismodells in Form eines Frameworks von Buxmann et al. [19]. Ausgehend von dem Framework von Buxmann et al. wurden durch eine weitere Literatur-

Komponente	Beschreibung
Preisbildung	Parameter für die Preisbildung sind die Preisermittlung (z.B. kostenbasiert oder nachfrageorientiert), und der Integrationsgrad (z.B. einseitig oder interaktiv).
Struktur des Zahlungsstroms	Hierbei wird festgelegt, ob es einen einmaligen, regelmäßig/wiederkehrend oder hybriden Zahlungsstrom seitens des Nutzers gib.
Bemessungsgrundlage	Die Bemessungsgrundlage gibt an, ob der Preis je Nutzer oder Zeit berechnet wird. Anhand der Bemessungsgrundlage entscheiden die meisten Nutzer, ob ein Preismodell als fair gilt.
Preisdifferenzierung	Als Preisdifferenzierung beschreibt man das Anbieten von gleichen Produkten zu unterschiedlichen Preisen. Zweck ist das bessere Abschöpfen der Konsumentenrente. Beispiel sind Mengen- oder Studentenrabatte.
Preisbündelung	Die Preisbündelung gibt an ob nach Angebot (z.B. reine Bündelung), Produktart (z.B. Software, Wartung) oder Integrationsgrad different wird.
Dynamische Preisstrategien	Hierbei lassen sich die Pentrationsstrategie, Follow-the-free-Strategie sowie Skimmingstrategie unterscheiden.

Tabelle 3.1: Übersicht über die Komponenten eines Preismodells [19]

recherche mögliche Kombinationen der Komponenten von Preismodellen identifiziert. Tabelle 3.2 zeigt eine entsprechende Übersicht. Die Tabellen 3.1 und 3.2 sind dabei über die Komponenten Preisbildung und Bemessungsgrundlage miteinander verknüpft.

Ein Hauptproblem bei der genaueren Untersuchung der Preismodelle ist die Transparenz der Angebote. So zeigt eine Untersuchung von Lehmann et al., dass bei einer Untersuchung von 114 SaaS-Anbietern,

Preis-bildung	Bemessungsgrundlage	Beschreibung
Nutzungsunabhängig	Flatrate	Bei diesem Preismodell sind alle Funktionen im kompletten Umfang inklusive [19]. Der Nutzer zahlt dabei in der Regel monatlich ein Nutzungsentgelt.
	Freemium	Dieses Preismodell findet sich oft als Basistarif. Hierbei wird dem Nutzer kostenlos Zugang zum angebotenen Dienst gewährt. Es lassen sich in der Praxis Anbieter finden, die ihr Angebot komplett kostenlos zur Verfügung stellen (Open-Source oder durch Werbung finanziert). Andererseits gibt es gibt es Anbieter, die eine im Funktionsumfang stark reduzierte Version als Freemium bereitstellen [58].
	Einmalzahlung	Weiterhin möglich ist eine Einmalzahlung. Hierbei zahlt der Nutzer einmalig eine Gebühr an den Anbieter und erhält damit Zugang zum gewünschten Dienst. Dieses Modell ist lediglich bei klassischen Offline-Softwarelösungen zu finden, bei dem der Nutzer die Software erwirbt und diese dann unbegrenzt nutzen kann [19].
Nutzungsabhängig	Transaktionsbasiert	Bei dieser Bemessungsgrundlage werden dem Nutzer die tatsächlich verbrauchten Einheiten in Rechnung gestellt. Eine *Transaktion* kann dabei hinsichtlich technischer Ebene (z. B. Anzahl der Dienstaufrufe) als auch inhaltlicher Ebene (z. B. Anzahl bearbeiteter Lieferpositionen) definiert sein [19]. Diese Abrechnung kann dabei gebündelt pro Monat oder direkt on-demand geschehen.
	Speicherplatzbasiert	Bei diesem Preismodell wird der Preis in Abhängigkeit des vom Nutzer genutzten Speicherbedarfs (z. B. in GB) gemessen [19].
	Zeitbasiert	Bei diesem Preismodell wird dem Nutzer eine zeitabhängige Nutzung in Rechnung gestellt. Dabei sind prinzipiell alle Zeitintervalle möglich (z. B.. pro Minute) [19].
Hybrid	Nutzungs-unabhängig kombiniert mit -abhängig	Bei hybriden Modellen werden jeweils Komponenten aus den genannten Modellen kombiniert [61]. So ist es beispielsweise denkbar, eine Softwarelizenz als *Einmalzahlung* zu beziehen und mittels eines Wartungsvertrages einen wiederkehrenden Zahlungsstrom zu erzeugen, der auch *nutzungsabhängige* Komponenten enthält.
Interaktiv	Auktionen	Innerhalb einer Auktion versteigert der Anbieter die Dienste an seine Nutzer. Auktionen sind dadurch charaterisiert, dass sie die Zahlungsbereitschaft des Kunden effektiv abschöpfen [58].
	Reverse Pricing	Auch als *Name-Your-Own-Price* bezeichnet [94]. Hierbei wird seitens des Nutzers ein Gebot abgegeben, das vom Anbieter angenommen wird, sobald eine (für den Nutzer unbekannte) Preisschwelle überschritten wird [11].

Tabelle 3.2: Übersicht der Preismodelle

48% auffindbare Informationen bezüglich des Preismodells und 26% teilweise auffindbare, und weitere 26% keine Informationen zum Preismodell gaben [62]. Weiterhin zeigte diese Untersuchung bezüglich des Zahlungsstroms, dass die Mehrzahl (56 von 84 Anbietern) wiederkehrende Zahlungen nutzten. Exklusive Einmalzahlungen waren bei keinem Anbieter auffindbar, wohl aber eine Kombination und somit eine hybride Form von Einmal- und wiederkehrenden Zahlungen (22 von 84 Anbietern) [62]. Abbildung 3.2 zeigt die Ergebnisse einer Untersuchung von Lehmann et al. unter 84 SaaS-Anbietern hinsichtlich der Ausprägung von nutzungsunabhängigen und nutzungsabhängigen Preismodellen [62]. Wie in der Grafik

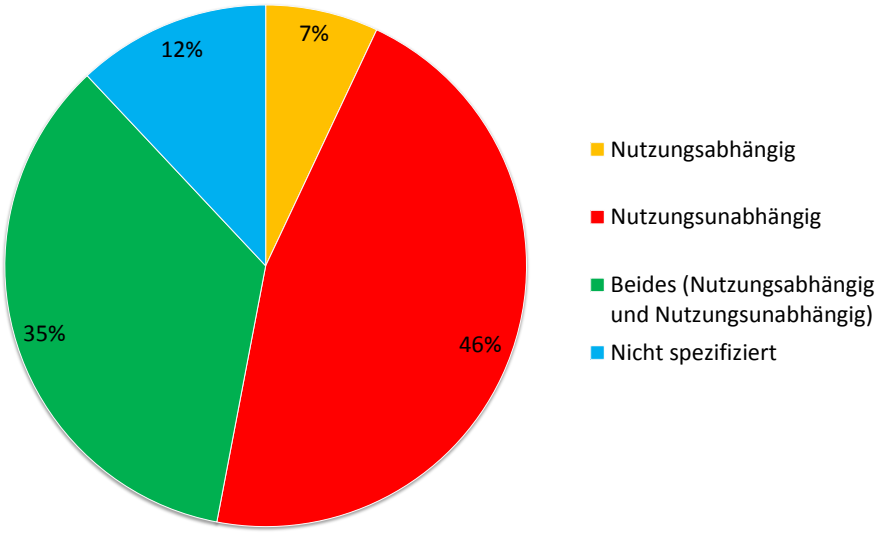

Abbildung 3.2: Ausprägung der Preismodelle (basierend auf [62])

zu erkennen ist, konnten in der Untersuchung keine Angebote von interaktiven Preismodellen aufgefunden werden. Abbildung 3.2 zeigt daher lediglich die nutzungsabhängigen sowie nutzungsunabhängigen Preismodelle.

Bewertung der nutzungsunabhängigen Preismodelle

Die nutzungsunabhängigen Preismodelle charakterisieren sich dadurch, dass der Nutzer eine Gebühr zahlt, die nicht mit dem direkten Nutzungsverhalten korreliert ist. Dabei bieten diese auf Kundenseite eine absolute Planungssicherheit bezüglich der entstehenden Kosten [19, 22]. Weiterhin existieren Freemium-Modelle, welche den Dienst zunächst einmal kostenlos anbieten. Diese finanzieren sich allerdings oft durch Werbung oder werden mit einem nutzungsabhängigen Preismodell kombiniert [19]. Ein klassisches Beispiel für solch ein Preismodell ist *Dropbox*, bei dem 2 GB Speicherplatz kostenlos zur Verfügung gestellt werden und einen größere Speichermenge mit regelmäßigen Gebührungen verbunden ist. Dabei können diese Preismodelle als Motivation zum „Kennenlernen“ der Software dienen. Auf Anbieterseite bieten die nutzungsunabhängigen Preismodelle den Vorteil, dass Kunden allein für die Option der unbegrenzten Nutzung bereit sind mehr zu zahlen [19, 102]. Zudem überschätzen Nutzer ihr Nutzungsverhalten häufig, was als *Flatrate-Bias* bezeichnet wird [55].

In der Praxis ist es schwer, Anbieter zu finden, die ihren Kunden alle Funktionen in unbegrenzter Anzahl zur Verfügung stellen. Aus diesem Grund finden sich jeweils eingeschränkte Flatrate-Modelle, die die Nutzung entweder hinsichtlich der Anwenderzahl, der Zeit oder des Funktionsumfangs beschränken. Die Einmalzahlung als Preismodell ist zudem zwar theoretisch vorhanden, findet allerdings bei SaaS-Lösungen derzeit keine Anwendung. In einigen Fällen werden Einmalzahlungen in hybriden Modellen integriert, z.B. als Einrichtungsgebühren eines Dienstes [19].

Bewertung der nutzungsabhängigen Preismodelle

Neben den nutzungsunabhängigen Preismodellen bieten die nutzungsabhängigen Preismodelle den Vorteil, die tatsächlich verbrauchte Leistung exakt in Rechnung zu stellen. Das oft benannte, nutzungsabhängige Preismodell *Pay-as-you-go* [58] stellt Anbieter vor die Problematik, dass Zahlungsströme schwer vorherzusagen sind [61]. Diese Preismodelle bemessen die Kosten anhand der verbrauchten Einheiten. Nutzungsabhängige Preismodelle erlauben es dem Anbieter, Kunden mit geringerer Nutzungsintensität geringere absolute Kosten in Rechnung zu stellen als sogenannten „Power User“ [61]. Im Gegensatz zu Flatratemodellen kann der Anbieter die tatsächlich verbrauchte Leistung in Rechnung stellen [61]. Demgegenüber steht ein erhöhter administrativer Aufwand auf Anbieterseite, da der Verbrauch in Form eines Monitoringsystems gemessen werden muss [61]. Aus Kundensicht spiegelt dieses Preismodell den Vorteil der Skalierbarkeit im Cloud Computing am besten wider, da dieses Preismodell ermöglicht, den „Konsum“ gänzlich auszuweiten oder zu reduzieren [5].

In der Praxis finden sich transaktionsbasierte, speicherplatzbasierte sowie zeitbasierte Preismodelle. So kann als Beispiel für einen Anbieter mit einem *Preismodell abhängig von der Zahl der Transaktionen* Salesforce genannt werden. Hier wird eine Abrechnung pro Benutzer/Monat angeboten [90]. Ein Beispiel für einen Anbieter mit einem *Preismodell abhängig vom Speicherbedarf* ist der bereits genannte Dienst Dropbox [19].

Bewertung der hybriden und interaktiven Preismodelle

Wie die Untersuchung von Lehmann et al. zeigt, bieten bei der Stichprobe von 84 Unternehmen, 22 eine hybride Form von Einmalzahlungen und wiederkehrenden Zahlungen an [62]. Eine ähnliche Untersuchung aus dem Jahr 2012 zeigt ein vergleichbares Ergebnis. So boten hier 37 von 166 Anbietern hybride SaaS-Zahlungsmodelle an [61]. Diese Modelle finden demnach erkennbar Anwendung in der Praxis. SaaS-Anbieter können durch die Kombination aus nutzungsunabhängigen und nutzungsabhängigen Preiskomponenten einen relativ stabilen Zahlungsstrom sicherstellen [60]. Neben diesem Vorteil verbinden hybride Modelle die Vor- und Nachteile der jeweiligen unterliegenden Preismodelle miteinander.

Die interaktiven Preismodelle sind hier der Vollständigkeit halber genannt. Sie bieten den Vorteil, dass der Nutzer aktiv Einfluss auf den Preis hat. Es lassen sich allerdings keine SaaS-Lösungen finden, die derzeit solch ein Preismodell anbieten. Neben dem Reverse Pricing existieren auktionsbasierte Modelle. Diese gelten allerdings bei digitalen Gütern als ökonomisch weniger sinnvoll, da deren Produktionskosten pro Stück gegen null gehen [19, 92].

3.3 Regulatorische Anforderungen

Bedingt durch die Vielzahl an regulatorischen Anforderungen, die von Banken zu erfüllen sind, kommt auch in diesem Kontext den Dienstgüteparametern bei der Nutzung von Cloud-basierten Angeboten - sofern die Bank diese als „... as a Service“-Lösungen einsetzen - eine wesentliche Bedeutung zu. Dieser Abschnitt gibt einen Überblick, wie sich die regulatorischen Anforderungen an Banken auf die jeweils in der „... as a Service-Lösung“ bereitzustellenden Dienstgüteparameter auswirken. Die regulatorischen Anforderungen an Banken ergeben sich im Wesentlichen aus den nachfolgenden drei Vorgaben und Regelwerken:

1. Technische und organisatorische Maßnahmen nach §9 Bundesdatenschutzgesetz (BDSG) [12]
2. Mindestanforderungen an das Risikomanagement - MaRisk [6] (Rundschreiben 10/2012 der Bundesanstalt für Finanzdienstleistungsaufsicht)
3. IT Sicherheitsgesetz [16]

In Bezug auf die MaRisk liegt den folgenden Betrachtungen die Version 10/2012 zu Grunde. Anpassungen durch den aktuellen Konsultationsentwurf der MaRisk sind hier nicht berücksichtigt.
Auch sind hier nur die Anforderungen betrachtet, die zum Erstellungszeitpunkt dieses Whitepapers bereits verbindlich für Banken in der Bundesrepublik Deutschland anzuwenden sind. So sind zum Beispiel europäische Regelungen wie die EU Datenschutzrichtlinie jeweils in nationales (deutsches) Recht umgesetzt und müssen hier nicht gesondert betrachtet werden. Auch sind die Regelungen zum Datenschutz in der Bundesrepublik Deutschland mit die strengsten in der EU (ein Ausnahme bildet Luxemburg, wo Bankdaten die Landesgrenzen nicht verlassen dürfen) und damit im Rahmen dieser Arbeit als Grundlage als hinreichend umfassend anzusehen.
Auch das Bankgeheimnis ist hier nicht als Anforderung aufgenommen, da es sich um eine, durch den Bankvertrag stillschweigend übernommene Verpflichtung der Bank, keinerlei Informationen über Kunden und deren Geschäftsbeziehungen unbefugt an Dritte weiterzugeben, handelt. Die daraus für die Bank resultierenden Pflichten sind in aller Regel unabhängig von der Art der Diensterbringung [32].
Die ausgewählten regulatorischen Anforderungen stellen den kleinsten gemeinsamen Nenner an Anforderungen dar, die eine XaaS Lösuung erfüllen muss. Darüber hinaus gibt es in vielen Finanzinstituten zusätzlich interne Regeln und Anforderungen die einzuhalten sind um *compliant* aus Sicht der jeweiligen Bank zu sein. Da diese zusätzlichen Anforderungen sich von Bank zu Bank unterscheiden, geht dieses Whitepaper nur auf die, von außen (regulatorisch) vorgegebenen Anforderungen ein.

3.3.1 Technische und organisatorische Maßnahmen - §9 BDSG

Sofern in einer cloud-basierten Lösung auch personenbezogene Daten gespeichert oder verarbeitet werden, sind auch einige Vorgaben aus dem BDSG zu beachten. Dabei gibt es sowohl organisatorische / vertragliche, als auch Anforderungen mit IT Relevanz. Bei der Verarbeitung von personenbezogenen Daten durch einen Dritten spricht man von einer Auftragsdatenverarbeitung (ADV) im Sinne des BDSG. Gemeint ist damit die Erhebung, Verarbeitung oder Nutzung von personenbezogenen Daten durch einen Dienstleister im Auftrag der verantwortlichen Stelle nach §11 BDSG, wobei der Auftraggeber die Bank voll für die Einhaltung der Bestimmungen des BDSG verantwortlich bleibt.

Der Dienstleister ist für die Einhaltung des Datengeheimnisses §5 BDSG und der technischen und organisatorischen Maßnahmen nach §9 BDSG verantwortlich und an die Weisungen des Kunden (im Vertrag geregelt) gebunden. Das gilt auch für die Beauftragung von Unterauftragnehmern durch der Dienstleister.
Gemäß §9 BDSG sind alle Stellen, welche personenbezogene Daten verarbeiten, erheben oder nutzen verpflichtet, technische und/oder organisatorische Maßnahmen (kurz: TOM - vollständige Auflistung in Tabelle 3.3) zu treffen, um zu gewährleisten, dass die Sicherheits- und Schutzanforderungen des BDSG erfüllt sind. Da nach deutschem Recht bzw. der verbreiteten Rechtsauffassung relativ schnell der „Personenbezug“ anzunehmen ist (zum Beispiel für die IP-Adresse) sollen im Folgenden die Anforderungen mit IT-Relevanz aus den technische und/oder organisatorische Maßnahmen (TOM) den Dienstgüteparametern gegenübergestellt werden. Die Spezifizierung dieser Anforderungen ergibt sich aus der Anlage zu §9 Satz 1 BDSG.
Für einen überwiegenden Teil dieser technischen und organisatorischen Maßnahmen ist eine Spiegelung in die in dieser Arbeit definierten Dienstgüteparameter sinnvoll möglich. Betrachtet man nun die TOM und die dahinter stehenden Forderungen genauer, so lassen sich die Zuordnungen zu den Dienstgüteparametern wie in Tabelle 3.4 dargestellt vornehmen.
Keine Berücksichtigung haben die folgenden Dienstgüteparameter erfahren, da diese keine / nur indirekte Relevanz für die Einhaltung der TOM's haben:

- Wirtschaftlichkeit / Kosten
- Skalierbarkeit / Elastizität

technische und/oder organisatorische Maßnahmen	Beschreibung
Zutrittskontrolle	Verhinderung von räumlichen Zutritt - Elektronische Zutrittskontrolle (z. B. durch Ausweisleser)
Zugangskontrolle	Verhinderung der Nutzung von „... as a Service“ Lösungen - Autorisierte Benutzerkennungen und individuelle Passwörter für den Zugang zu Datenverarbeitungsystemen
Zugriffskontrolle	Zugriff nur auf berechtigte Inhalte / kein unbefugtes manipulieren personenbezogener Daten - Abgestufte Zugriffskonzepte mit unterschiedlichen Kennungen und Passwörtern für den Zugriff auf Datenverarbeitungssysteme
Weitergabekontrolle	Verhinderung der unbefugten Weitergabe von Daten beim Transport und der Weitergabe - Technische Maßnahmen, um zu verhindern, dass Kundendaten bei der elektronischen Übertragung oder während ihres Transports unbefugt verarbeitet oder genutzt werden können (z. B. durch Verschlüsselung oder Schutz durch Passwörter)
Eingabekontrolle	Nachvollziehbarkeit der Eingabe / Änderung von Daten - Aufzeichnung von Zugriffen der Mitarbeiter auf Kundendaten in Logfiles bei Verarbeitung auf Systemen
Auftragskontrolle	Verarbeitung der Daten gemäß Weisung des Auftraggebers - Anweisung an Mitarbeiter über Umfang und Inhalt der vom Kunden erteilten Weisungen
Verfügbarkeitskontrolle	Schutz der Daten vor Zerstörung und Verlust - z. B. Maßnahmen zum Brandschutz und bei Stromausfällen in Rechenzentren
Trennungsgebot	Trennung der personenbezogenen Daten abhängig vom Grund der Erhebung und in Bezug auf Umgebungen - Personenbezogene Daten werden physisch getrennt gespeichert

Tabelle 3.3: Technische und organisatorische Maßnahmen nach Anlage zu §9 Satz 1 BDSG

Die Zuordnung der geforderten TOM ist jeweils zu demjenigen Dienstgüteparameter erfolgt, der die Anforderungen am besten abdeckt. Im Umkehrschluss lässt sich folgern, dass die Definition und das

	Dienstgüte-parameter	Verfügbarkeit	Sicherheit	Portierbarkeit	Robustheit
Technische und organisatorische Maßnahmen	Zugangs-kontrolle		Bezug auf Nutzung der Systeme		
	Zugriffs-kontrolle		Nur berechtigter Zugriff auf Daten		
	Weitergabe-kontrolle		Daten sicher / nachvollziehbar übertragen		
	Eingabe-kontrolle		Nachvollziehbarkeit von Änderungen		Nachvollziehbarkeit von Änderungen
	Auftrags-kontrolle		Daten nur gem. Auftrag verarbeiten		
	Verfügbarkeits-kontrolle	Schutz vor Verlust			
	Trennungs-gebot		Trennung der Daten nach Zweck		

Tabelle 3.4: relevante Dienstgüteparameter für technisch organisatorische Maßnahmen

Erfassen der Einhaltung von Dienstgüteparametern eine notwendige Voraussetzung zum Nachweis der, durch den Dienstleistern zu erbringenden Leistung und Einhaltung seiner Verpflichtung darstellt.

3.3.2 Mindestanforderungen an das Risikomanagement - MaRisk (10/2012)

Ein zweiter Anforderungskomplex leitet sich aus den MaRisk ab.

> "Die MaRisk werden als Rundschreiben durch die BaFin herausgegeben und orientieren sich, weitestgehend am § 25a Abs. 1 KWG (Risknews 02/05). Damit werden die Anforderungen aus Basel II bzw. Basel III an das Risikocontrolling von Banken in deutsches Recht umgesetzt. Damit hat die BaFin de facto eine verbindliche Auslegung der Anforderungen an das Risikomanagement für die Banken in Deutschland geschaffen."

Die Anforderungen, die von Banken unter der Aufsicht der ECB[5] Banking Supervision (SSM) zu erfüllen sind werden hier nicht aufgeführt. Dies hat im Wesentlichen zwei Gründe: Zum einen gibt es bislang wenige IT Prüfungen der betroffenen Banken, zum anderen ist dies, basierend auf Äußerungen der ECB auf der Konferenz „IT Aufsicht bei Banken" am 7.10.2015 in Bonn, nicht zu erwarten, dass die Prüfungspraxis wesentlich von der der BaFIN abweichen wird.

Auch aus der MaRisk lassen sich Anforderungen an die Dienstgüteparameter einer „... as a Service" Lösung ableiten. Im Gegensatz zu den eindeutigen Forderungen in Bezug auf die TOM, ist eine vergleichbare Eindeutigkeit im Bereich der MaRisk nicht gegeben.

[5] engl. European Central Bank

Mit Blick auf die derzeit bestehenden / angekündigten Regulierungen, die den regulatorischen Rahmen für die Nutzung von externen IT-Dienste (Cloud) vorgeben, ist hier sicher an erster Stelle die Mindestanforderungen an das Risikomanagement - MaRisk zu nennen. In den Anforderungen aus dem Allgemeinen Teil (AT) AT 4, den Organisationsrichtlinien in AT 5, den Anforderungen an die technisch-organisatorische Ausstattung in AT 7 sowie insbesondere in AT 9 Outsourcing sind die jeweils zu beachtenden Vorgaben zu finden. Im Gegensatz zu den doch relativ eindeutigen Kriterien in welchen Fällen es sich um die Verarbeitung personenbezogener Daten handelt, ist die Entscheidung, welche Anforderungen der MaRisk für eine konkrete „... as a Service“ Lösung durch die Bank anzuwenden sind nicht so einfach zu treffen.

Jede Bank prüft auf Basis ihres individuellen Geschäftsmodells und der Schutzbedarfsanalyse die Sicherheitsrelevanz des externen Dienstes für den jeweiligen Geschäftsprozess (vgl. Abbildung 3.3).

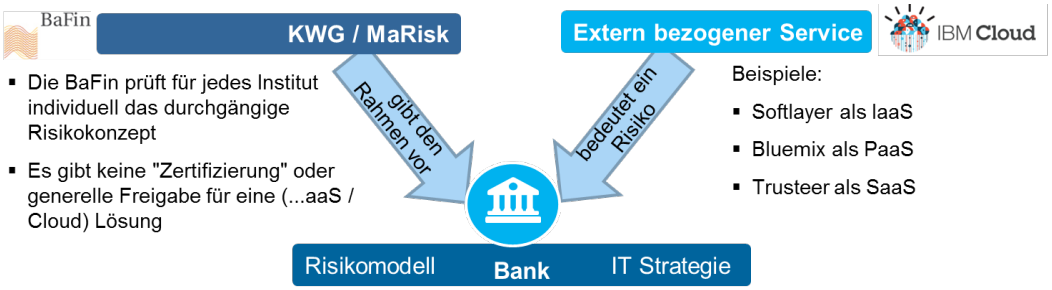

Abbildung 3.3: Individuelle Risikobewertung durch die Bank (Quelle IBM)

Referenz? Woher kommt das Bild?

Diese Einschätzung ist - neben den in der IT-Strategie festgeschriebenen Rahmenbedingungen zur Cloud Nutzung - entscheidend für die entsprechenden individuellen Anforderungen der Bank an den Bezug des externen Dienstes.

Die wesentlichen Fragen die dabei zu stellen und zu beantworten sind:

- Welche Risiken bestehen?
- Welche Risiken werden durch einen externen Dienst erst hervorgerufen (oder auch vermindert)
- Wie werden diese Risiken adressiert / minimiert?
- Wie schützenswert sind die betroffenen Daten?

Für Abschätzungen des Risikos, welches der Bezug eines externen Dienst für ein Finanzinstitut bedeutet, ist es zunächst unerheblich, ob es sich dabei um ein klassisches IT Outsourcing oder einen extern bezogenen Cloud „... as a Service“ handelt. In allen Fällen ist zu prüfen, ob die Definition zur Auslagerung nach AT 9 Satz 1 anzuwenden ist. Eine Hilfestellung kann dabei die Betrachtung des Bereitstellungsmodells und der Dienstebene sein (vgl. Abbildung 2.2).
Unter der Annahme, dass die Bank zu der Einschätzung kommt, dass eine Auslagerung nach AT 9 der MaRisk vorliegt, kommen neben den Kriterien aus dem AT 4 und AT 7 der MaRisk auch die relevanten Punkte aus AT 9 zur Anwendung.

Die Aufstellung in Tabelle 3.5 enthält nur diejenigen Anforderungen der MaRisk, die gegenseitigen Einfluss auf Dienstgüteparameter haben. Analog zu der Zuordnung in Tabelle 3.3 - ist im Folgenden auch die Zuordnung der relevanten Kriterien aus der MaRisk zu den zuvor definierten Dienstgüteparametern erfolgt. In der Zuordnung, dargestellt in Tabelle 3.6, sind die folgenden Dienstgüteparameter nicht aufgenommen, da diese keine oder nur indirekte Relevanz für die Einhaltung der MaRisk haben:

- Wirtschaftlichkeit / Kosten

- Skalierbarkeit / Elastizität

MaRisk allg. Teil		Beschreibung (Zusammenfassung aus [6])
AT 4.3.1	IT-Berechtigungen	Bei IT-Berechtigungen wird eine mindestens jährliche, bei kritischen IT-Berechtigungen eine mindestens halbjährliche Überprüfung erwartet.
AT 7.2	Technisch-organisatorische Ausstattung	7.2.2 Integrität, die Verfügbarkeit, die Authentizität sowie die Vertraulichkeit der Daten incl. Berechtigungsvergabe und Nutzung von Standards. 7.2.3 Qualitätssicherung und Tests sowie die Trennung von Entwicklung, Test und Produktion.
AT 7.3	Notfallkonzept	Fortführung des Betriebs in Notfällen
AT 9.6 e	Datenschutz	Regelungen, die sicherstellen, dass datenschutzrechtliche Bestimmungen beachtet werden. Damit sind auch alle in Tabelle 3.4 aufgeführten Punkte relevant.
AT 9.6 g	Weiterverlagerung	Regelungen über die Möglichkeit und über die Modalitäten einer Weiterverlagerung, die sicherstellen, dass das Institut die bankaufsichtsrechtlichen Anforderungen weiterhin einhält.

Tabelle 3.5: Auszug der Anforderungen nach MaRisk

Dienstgüte-parameter		Verfügbarkeit	Sicherheit	Portabilität	Robustheit
MaRisk	AT 4.3.1 IT-Berechtigungen		Regelm. Überprüfung und Monitoring		
	AT 7.2 Technisch-organ. Ausstattung	Anwendung lokaler od. int. Standards (IT Grundschutz / ISO 27001)		Test und Abn. bei Veränderungen	
	AT 7.3 Notfallkonzept	Wiederanlaufpläne / Ersatzlösungen			Wiederanlaufpläne / Ersatzlösungen
	AT 9.6 e Datenschutz	siehe: Tabelle Technische und organisatorische Maßnahmen nach Anlage zu §9 Satz 1 BDSG			
	AT 9.6 g Weiterverlagerung			Übertragung auf anderen Dienstleister	

Tabelle 3.6: relevante Dienstgüteparameter für MaRisk

Wie in Abbildung 3.3 dargestellt werden die jeweils relevanten Kriterien durch die das Finanzinstitut zunächst auf Basis der eigenen Risikoeinschätzung festgelegt. Tabelle 3.6 verknüpft Aspekte des MaRisk mit den entsprechenden Dienstgüteparamenter.

3.3.3 IT-Sicherheitsgesetz

Das IT-Sicherheitsgesetz wurde am 12. Juni 2015 vom deutschen Gesetzgeber im Kontext der deutschen Cybersicherheitsstrategie verabschiedet und am 24. Juli 2015 im Bundesgesetzblatt verkündet. Banken und Finanzdienstleister fallen dabei unter den Sektor Finanzwesen nach §2 Absatz 10. Inwiefern sich praktische Auswirkungen auf Banken Finanzdienstleister ergeben bleibt abzuwarten, da diese im Rahmen der Vorgaben der Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin) bereits weitreichenden Regulierungen - auch in Bezug auf die IT-Sicherheit - unterliegen. Weitere Details werden sich erst aus den, noch ausstehenden Verordnungen und der Abstimmung zwischen der BaFin und dem Bundesamt für Sicherheit in der Informationstechnik ergeben.

3.4 Fazit

Softwaregüter, und damit auch SaaS-Dienste, gelten als Erfahrungsgüter, das heißt, sie können vom Nutzer erst bewertet werden, nachdem sie verwendet wurden [61]. Hinsichtlich des Preismodels hat es sich gezeigt, dass es vorteilhaft ist, eher ein einfaches Preismodell zu wählen, welches vom Kunden verstanden und evaluiert werden kann [23]. Weiterhin zeigen empirische Studien, dass Anbieter sowie Nutzer eher wenig Interesse bezüglich nutzungsabhängigen Preismodellen haben [61]. Hier sei eine Studie der *Software and Information Industry Association* genannt [61, 93], welche ermittelt hat, dass nur 5% der Geschäftskunden und 17% der Anbieter ein nutzungsabhängiges Preismodell bevorzugen. Die ursprünglich angenommene These der favorisierten Nutzung und das Anbieten von nutzungsabhängigen Preismodellen (*pay-as-you-go*) kann damit nicht für SaaS bestätigt werden. Überwiegend finden nutzungsunabhängige und hybride Modelle Anwendung.

Sowohl aus den Regelungen im Bundesdatenschutzgesetz, als auch die einschlägigen Anforderungen aus der MaRisk beeinflussen die relevanten, zu definierenden und zu überwachenden Dienstgüte Parameter einer „as a Service“-Lösung. Je weiter sich die Bereitstellung einer „as a Service“-Lösung von einer „on Premise“-Umgebung in Richtung „Public“ verschiebt umso wichtiger wird ein eindeutige Definition und Überwachung der aus regulatorischer Sicht notwendigen Dienstgüteparameter. Die regulatorischen Anforderungen spiegeln sich bei Banken unter anderem in den zu erfüllenden Dienstgüteparametern wieder. Wesentliche Einflussfaktoren sind dabei Aspekte wie vertragliche Festlegungen und die Ausgestaltung der Governance für eine cloud-basierte Diensterbringung. Die auch geforderten, technischen Maßnahmen unterscheiden sich nicht maßgeblich von denen, die bei einer traditionellen Diensterbringung einzuhalten sind. In Summe sind die zusätzlichen, im Fall einer cloud-basierten Diensterbringung zu erfüllenden Anforderungen lösbar – und die Anbieter von Cloud-basierten Bereitstellungsmodellen sind entspechend daran interessiert diese auch umzusetzten.

Nach heutigem Stand ein wesentlicher Hinderungsgrund für ein umfassenden Nutzung von Cloud-Diensten bei Banken darin zu sehen, dass seitens dieser Institute noch das notwendige Vertrauen für die Nutzung von IT-Lösungen außerhalb der eigenen Zugriffsmöglichkeiten fehlt. Je mehr sich auch bei Banken eine neutrale Abwägung von Risiken versus Vorteilen einer Cloud-Nutzung durchsetzt, umso mehr werden auch in diesem Bereich cloud-basierte Bereitstellungsmodelle Einzug halten und sich die cloud-basierte Diensterbringung durchsetzen.

4 Überwachung von Service Level Agreements

Im Rahmen dieses Whitepapers stehen die nicht-funktionalen Anforderungen von Cloud-Diensten (vgl. Kapitel 3) im Vordergrund. Diese Anforderungen an die bereitgestellten Ressourcen werden im Rahmen von SLAs zwischen Nutzer und Anbieter festgelegt und umfassen sowohl die Art der Bereitstellung als auch Nutzungsbedingungen sowie messbare nummerische Werte wie z.B. die Verfügbarkeit und die Reaktionszeit. Des Weiteren werden Strafen festgelegt, wenn der Anbieter die SLAs im Verlauf der Vertragslaufzeit verletzt [68]. Das SLA Management umfasst in dieser Hinsicht Maßnahmen zur Überwachung und Einhaltung von SLAs [89]. Im Gegensatz zur Überwachung traditioneller SLAs stellt die Überwachung von SLAs im Rahmen von Cloud Computing eine komplexe Problemstellung dar. Dies resultiert aus den besonderen Eigenschaften des Architekturparadigmas Cloud Computing. Im Unterschied zur traditionellen Überwachung sind komplexe Systeme notwendig, die die besonderen Aspekte dieser Architektur berücksichtigen. Im Sinne der NIST Definition [72] wird Cloud Computing durch mehrere wesentliche Eigenschaften charakterisiert (vgl. Kapitel 2.2). So werden von der physikalischen Hardware abstrahierte Ressourcen dynamisch einer beliebigen Anzahl von Nutzern über ein Netzwerk zur Verfügung gestellt. Die Ressourcen eines Cloud Computing Anbieters werden zwischen den einzelnen Nutzern aufgeteilt (Multi-Tenant Modell) und sind durch eine rasche Skalierung an die Bedürfnisse der einzelnen Nutzer gekennzeichnet. Dies wird durch eine generelle Abstraktion der physikalischen Ressourcen mittels virtueller Maschinen erreicht. Diese Art der Bereitstellung stellt besondere Anforderungen an die Überwachung von Cloud SLAs.

In diesem Kapitel werden Ansätze zur Überwachung von SLAs vorgestellt. Zunächst werden Konzepte für die Überwachung von traditionellen SLAs im Abschnitt 4.1 erläutert. Darauf aufbauend werden im zweiten Schritt die Besonderheiten und die damit verbundenen Herausforderungen der Überwachung von SLAs im Cloud Computing in Abschnitt 4.2 betrachtet.

4.1 Überwachung traditioneller SLAs

Keller et al. [50] haben festgestellt, dass SLAs oft in Dokumenten mit natürlicher Sprache festgehalten werden. Dies führt jedoch zu einem aufwendigen und langsamen Überwachungsprozess. Aus diesem Grund sollten SLAs in einer formalisierten Sprache vorliegen, die von Systemen interpretiert werden können, um einen automatisierten Prozess zu ermöglichen. Zu diesem Zweck müssen SLAs in natürlicher Sprache in Metriken umgewandelt werden, die eine automatische Überwachung erlauben. Es bedarf somit eine Zuordnungsvorschrift von SLAs zu den Hardwareressourcen der genutzten Dienste. Das heißt, es müssen Metriken definiert werden, die SLAs auf Hardwareebene abbilden. Keller et al. haben in ihrer Arbeit einen Ansatz entwickelt, der eine automatisierte Überwachung von SLA ermöglicht [50]. So wird zunächst das SLA Dokument in eine formalisierte Sprache umgewandelt, die auf dem XML-Schema basiert. In diesem Zusammenhang werden sowohl die Charakteristika als auch die möglichen Parameter zur Überwachung der verwendeten Dienste beschrieben. Darüber hinaus werden automatisch ablaufende Handlungsanweisungen definiert, die ausgeführt werden, wenn gegen bestimmte Grenzen und somit gegen ein SLA verstoßen wird. Im Sinne der formalisierten SLAs überwacht eine *SLA Compliance Monitor* Komponente der entsprechenden Dienste und stellt diese Informationen bereit [50].

Ein ähnliches Konzept zur Überwachung von traditionellen SLAs wurde von Ameller et al. [3] entwickelt. Nachdem eine Zuordnung zwischen SLAs und den eingesetzten Hardwareressourcen stattgefunden hat, werden Daten über die einzelnen Ressourcen gesammelt und überwacht. Dabei wird laufend überprüft, ob eine SLA Verletzung vorliegt. Sollte diese vorliegen, werden in einem automatischen Prozess Gegenmaßnahmen eingeleitet.

Die Überwachung von traditionellen SLAs erfolgt vor allem im Bereich der Web-Dienste. Die dortige Überwachung der SLAs kann nach Robinson [85] beispielsweise durch die Nutzung von dedizierten

Überwachungsservern erreicht werden. Insgesamt ist die traditionelle Überwachung durch statische Überwachungslösungen gekennzeichnet.

4.2 Verständnis und Anforderungen von Cloud-Überwachung in der Literatur

Im Allgemeinen erhöht der Einsatz von Cloud Computing die Anforderungen an die Flexibilität von SLAs [1]. Im Gegensatz zu klassischen Ansätzen findet im Cloud Computing eine dynamische Ressourcenallokation statt. Nutzer sind in der Lage, rasch Ressourcen zu allokieren oder freizugeben, und es kann somit eine bedarfsgerechte Bereitstellung der Ressourcen erreicht werden [72]. Jedoch besteht somit auch die Möglichkeit, dass ein Nutzer die zugesicherten Ressourcen aus einem SLA nicht vollständig ausschöpft. Aufgrund des Multi-Tenant Modells ist es im Interesse eines Anbieters, die Ressourcen einer großen Anzahl an Nutzern bereitzustellen. Die generelle Problematik, die sich in diesem Zusammenhang ergibt, besteht darin, die richtige Menge an Ressourcen für einen Nutzer zu reservieren. Sollte ein Cloud Anbieter mehr Ressourcen reservieren als der Nutzer momentan nutzt, führt dies zu einer Erhöhung des Energieverbrauches des Anbieters und diese Ressourcen stehen nicht mehr für andere Nutzer bereit [26, 71]. Das Bereitstellen der exakt verwendeten Menge eines Nutzers kann jedoch das Risiko einer SLA Verletzung erhöhen, da ein Nutzer rasch neue Ressourcen allokieren kann [68]. Ein Anbieter muss zu jeder Zeit sicherstellen, dass genügend Ressourcen für alle Nutzer bereitstehen. Zu diesem Zweck ist eine laufende Überwachung der Ressourcen erforderlich.

Die Überwachung von SLAs wird überwiegend über verschiedene Überwachungsansätze erreicht. Insgesamt wird die gesamte Infrastruktur eines Cloud Computing Anbieters überwacht. Die Informationen aus Überwachungsansätzen eignen sich neben dem Management der Infrastruktur ebenfalls zur Implementierung von Mechanismen, die SLA Verstöße vorzeitig erkennen und somit entsprechende Gegenmaßnahmen einleiten können [1]. Weiterhin könnten geeignete Überwachungsansätze auch dazu genutzt werden, dynamische SLAs anzubieten, die beispielsweise an die aktuelle Auslastung der Infrastruktur angepasst sind. Dadurch wäre eine effizientere Preisgestaltung möglich [1, 51]. Somit werden im Kontext des SLA Managements Überwachungsansätze sowohl zur Kontrolle der vereinbarten SLAs als auch als Ausgangspunkt zur Generierung von geeigneten SLAs eines Anbieters eingesetzt [1]. Darüber hinaus bilden Überwachungsansätze eine wichtige Grundlage für verschiedene Bereiche des Cloud Computings. Aceto et al. identifizierten die folgenden Bereiche: Kapazitäts-/Ressourcenplanung, Kapazitäts-/Ressourcenmanagement, Rechenzentrumsmanagement, Performance Management, Abrechnungsmanagement, Störungsmanagement, Sicherheitsmanagement [1]. Die genannten Bereiche stehen im unmittelbaren Zusammenhang mit dem SLA Management. Beispielsweise kann eine Störung des Systems zu einer Verletzung eines SLAs führen und alleinig durch ein geeignetes Überwachungsanssystems sowie eines Störungsmanagements rasch erkannt und behoben werden. Ferner verlangt die Einhaltung von regulatorischen Anforderungen ein funktionierendes Sicherheitsmanagement, das auf den Daten von Überwachungsansätzen aufbaut [1]. In dieser Hinsicht erschwert die generelle Ausrichtung eines Cloud Computing Systems die Einhaltung von regulatorischen Anforderungen. Durch die ganzheitliche Abstraktion der physikalischen Hardware ist es schwieriger Anforderungen zu erfüllen, die sich beispielsweise auf eine örtliche Beschränkung von Daten und der Datenverarbeitung beziehen. Gleichzeitig entstehen durch deren Nichteinhaltung erhebliche finanzielle Strafen für Anbieter. Somit muss zu jeder Zeit ersichtlich sein, dass entsprechende Anforderungen eingehalten werden [67]. Zu diesem Zweck werden geeignete Überwachungssysteme eingesetzt.
Der Einsatz von Überwachungssystemen im Kontext von Cloud Computing ist mit einigen Herausforderungen verbunden. Gleichermaßen wie Cloud-Applikationen sind Überwachungssysteme Dienste, die bestimmte Anforderungen erfüllen müssen. Die Anforderungspunkte eines Überwachungssystems weisen Ähnlichkeiten hinsichtlich der vorgestellten Merkmale von nicht-funktionalen Anforderungen an Dienste aus Tabelle 3.1 auf. Sie unterscheiden sich jedoch bezüglich ihrer spezifischen Ausprägung. Insgesamt

lassen sie sich zum Großteil auf die Überwachung von traditionellen SLAs übertragen. So verlangt die Überwachung von Web-Diensten ebenso eine vollständige und genaue Überwachung der einzelnen Komponenten. Durch eine Missachtung dieser Anforderungen ist es denkbar, dass ein Überwachungssystem fehlerhafte Werte liefert und auf dessen Basis inkorrekte Entscheidungen getroffen werden. Daneben sind die Anforderungspunkte Skalierbarkeit und Elastizität im Rahmen der Überwachung von Cloud SLAs von besonderer Bedeutung. Durch die dynamischen Ressourcen im Cloud Computing unterliegt die Systemstruktur einer ständigen Veränderung, die beachtet werden muss [1]. Diese Art der Veränderung liegt in der Überwachung traditioneller SLAs nicht vor. Aceto et al. haben die allgemeinen Anforderungen an ein Überwachungssystem im Cloud Computing definiert, um zu gewährleisten das grundlegende Konzepte des Cloud Computing berücksichtigt werden [1]. Die verschiedenen Anforderungspunkte werden in Tabelle 4.1 im Detail erläutert.

Grundsätzlich muss ein Überwachungssystem verschiedenartige Ressourcen überwachen können, um eine geeignete Informationsbasis für ein SLA Management im Cloud Computing Bereich darzustellen. Weiterhin muss es flexibel bezüglich der dynamischen Skalierung eines Cloud Computing Systems sein

Merkmal	Ausprägung
Skalierbarkeit	Ein Überwachungssystem muss für eine große Anzahl an Ressourcen effizient Daten sammeln, transferieren und analysieren können, ohne dass das System beeinträchtigt wird.
Elastizität	Ein Überwachungssystem muss mit der laufenden dynamischen Änderung der Cloud Computing Architektur umgehen können. In der Cloud Infrastruktur können beispielsweise laufend Änderungen bezüglich der Anzahl und der Größe von virtuellen Instanzen auftreten.
Verfügbarkeit, Belastbarkeit	Ein Überwachungssystem muss zu jedem Zeitpunkt funktionieren (Verfügbarkeit), der Ausfall von verschiedenen Komponenten sorgt nicht für einen Ausfall des Überwachungssystems (Belastbarkeit).
Adaptivität	Ein Überwachungssystem sollte die Aktivitäten im Cloud Computing so wenig wie möglich einschränken. Ein System kann sich der Auslastung des Systems anpassen.
Aktualität	Ein Überwachungssystem muss alle Daten zeitnah bereitstellen, um auf dessen Basis Entscheidungen treffen zu können.
Autonomität	Ein Überwachungssystem muss automatisch auf unvorhergesehene Ereignisse reagieren, ohne dass ein manueller Eingriff erforderlich ist. Das bedeutet, dass sich das System automatisch den Veränderungen von verschiedenen Ressourcen anpasst.
Vollständigkeit, Erweiterbarkeit	Ein Überwachungssystem muss eine Vielzahl von Ressourcen (physikalisch / virtuell) in einer Multi-Tenant Umgebung unterstützen. Zusätzlich sollte es bezüglich der Funktionalität erweiterbar sein.
Genauigkeit	Ein Überwachungssystem muss genaue Werte der überwachten Komponenten liefern. Ungenaue Werte können zum fälschlichen Verletzen von SLAs führen.

Tabelle 4.1: Anforderungen an ein Überwachungssystem; basierend auf [1]

[65]. Caron et al. unterscheiden zwischen “High-Level Monitoring” und “Low-Level Monitoring” [20]. “Low-Level Monitoring” bezieht sich auf die Überwachung von physikalischen Ressourcen [20]. Dies kann beispielsweise die Auslastung der CPUs, des Speichers oder die Temperatur der Komponenten betreffen. Dadurch, dass im Cloud Computing nur eine virtualisierte Infrastruktur angeboten wird, werden diese Informationen alleinig vom Anbieter erhoben und verarbeitet [96]. Diese Überwachung ist jedoch wichtig, um beispielsweise die Auslastung oder den Ausfall einer physikalischen Hardware rechtzeitig zu erkennen. Es besteht ein unmittelbarer Einfluss von Problemen der physikalischen Hardware und den angebotenen SLAs für den Nutzer. “High-Level Monitoring” betrifft hingegen die Überwachung der virtualisierten Infrastruktur. In diesem Bereich sollte sichergestellt werden, dass alleinig berechtigte Parteien Zugriff auf die Überwachungsdaten haben [20]. Abhängig von der verwendeten Diensteebene (siehe Abbildung 2.1), werden verschiedene Bereiche vom Cloud Computing Anbieter überwacht, um sicherzustellen, dass die SLAs eines Nutzers eingehalten werden [20]. Die Dienstebene definiert, inwieweit die Kontrolle der virtualisierten Hardware beim Nutzer liegt. Im Layer SaaS liegt die Kontrolle bis zur Anwendungsebene beim Cloud Computing Anbieter [96]. Dies macht eine Überwachung bis zur Anwendungsebene erforderlich. SLA Metriken werden dabei für jede Anwendung separat definiert [68]. Insgesamt müssen zwischen “Low-Level Monitoring”, “High-Level Monitoring” sowie den definierten SLAs eindeutige Beziehungen existieren, um Verstöße von SLAs bestimmen zu können [29].

In der Literatur existieren unterschiedliche Ansätze zur Überwachung von Cloud SLAs, die im Folgenden unter Berücksichtigung der in diesem Abschnitt identifizierten Anforderungen (vgl. Tabelle 4.1) näher beschrieben werden. Dabei liegt der Fokus insbesondere auf den Anforderungen der Skalierbarkeit, Elastizität, Adaptivität, Vollständigkeit und Erweiterbarkeit, da diese Anforderungen die besonderen Merkmale der hohen Dynamik und Ressourcenvielfalt im Cloud Computing erfassen.

4.3 Ausgewählte Überwachungsansätze in der Literatur

In [78] stellen Povedano-Molina et al. die dezentralisierte Überwachungsarchitektur *DARGOS* für Cloud Computung vor, die ein skalierbares publish/subscribe System nutzt, um die Überwachung der verschiedenen Komponenten der Cloud Infrastruktur und die Einhaltung von SLAs zu gewährleisten. Dieses System überwacht sowohl die virtuellen als auch die physikalischen Ressourcen einer Cloud Infrastruktur. Es besteht aus zwei Hauptkomponenten. Der *Node Monitoring Agent* sammelt verschiedene Nutzungsdaten, wie z.B. die CPU Auslastung oder die Speicherplatzbelegung, kann aber auch um neue Metriken erweitert werden. Mehrere dieser “Monitoring Agents” werden innerhalb der Cloud verteilt, die spezifische Daten sammeln. Der *Node Supervisor Agent* aggregiert die Daten von mehreren *Node Monitoring Agents* und stellt diese Informationen bereit. Der *Node Supervisor Agent* kann dabei flexibel unterschiedliche Überwachungsdaten aggregieren und somit adaptiv an die Eigenschaften der Cloud-Infrastruktur angepasst werden. Insgesamt implementiert DARGOS eine agentenbasierte Überwachungsarchitektur.

Das Elastic Monitoring Framework wurde von König et al. [54] entwickelt und basiert auf dem Peer-to-Peer (P2P) Protokoll. Aufgrund der Wahl eines dezentralen P2P-basierten Überwachungssystems, handelt es sich um einen skalierbaren Überwachungsansatz. Des Weiteren adressiert das Framework auch insbesondere das Merkmal der Elastizität im Cloud Computing, wie der Name bereits andeutet. Zu diesem Zweck besteht das Framework aus drei unterschiedlichen Komponenten. Diese arbeiten auf den verschiedenen Dienstebenen des Cloud Computings (IaaS, PaaS, SaaS), um eine ganzheitliche Überwachung zu ermöglichen. Die erste Komponente *Data* beinhaltet eine Reihe von Adaptern, die auf der physikalischen Hardwareebene und auf allen Dienstebenen zum Einsatz kommen. Adapter werden verwendet, um Daten aus den verschiedenen Komponenten der Cloud Architektur zu sammeln und bereitzustellen. Adapter besitzen ähnliche Eigenschaften zu den *Node Monitoring Agents*, die im Ansatz *DARGOS* zum Einsatz kommen. Zur Auswertung der gesammelten Informationen wird die *Processing* Komponente verwendet. Diese stellt eine Abfragesprache bereit mit deren Hilfe Überwachungsanfragen in der gesam-

ten Infrastruktur verteilt und Informationen von den Adaptern aggregiert werden können. SLAs können über Regeln innerhalb der Abfragesprache definiert werden. So führt das Zutreffen einer Regel zu einem automatischen Prozess, der ebenfalls festgelegt werden kann. Die *Distribution* Komponente sorgt für die korrekte Verteilung der Adapter in der Cloud Infrastruktur. Diese muss zum Beispiel bei einer Veränderung in der Anzahl der virtuellen Maschinen im System angepasst werden. Die gesammelten Informationen über alle Ebenen hinweg können schließlich ausgewertet und dazu genutzt werden, um komplexe Management-Aufgaben innerhalb der Cloud-Infrastruktur zu automatisieren.

Der Ansatz M4Cloud von Mastelic et al. [68] basiert auf einer Überwachungskomponente in Zusammenarbeit mit einer sogenannten FoSII Infrastruktur. Insgesamt ermöglicht dieser Ansatz die Überwachung von SLAs auf Anwendungsebene. Die FoSII Infrastruktur wird in diesem Ansatz als Cloud Management System eingesetzt und dient vorrangig zum Management der SLAs. Insbesondere ermöglicht diese Infrastruktur eine Zuordnung von Ressourcenmetriken zu SLA spezifischen Metriken auf Anwendungsebene. Beispielsweise kann die SLA Metrik Performance abhängig vom Datendurchsatz und der Antwortzeit auf Ressourcenebene ausgedrückt werden. Die FoSII Infrastruktur stellt diese Beziehungen her und legt somit fest, welche Ressourcen in einer Cloud Umgebung überwacht werden müssen, um eine ganzheitliche Überwachung zu ermöglichen. Ein Fokus von M4Cloud liegt somit auf der Vollständigkeit. Darüber hinaus wird ein *SLA-aware Scheduler* bereitgestellt. Dieser leitet Informationen über die zu überwachenden Metriken an die Infrastruktur weiter. Die Überwachung der SLAs erfolgt durch eine Überwachungskomponente, die aus drei Bereichen besteht. Der *Application Deployer* erhält die Informationen vom FoSII System und startet gemäß der SLAs die Anwendung. Darüber hinaus erhält jede Anwendung eine individuelle ID, um eine Überwachung auf Anwendungsebene zu ermöglichen. Die *Application Level Monitoring* Komponente nutzt individuellen IDs zur Überwachung spezifischer Ressourcen. Diese Daten werden der FoSII Infrastruktur bereitgestellt. Der *Metric Plugin Container* stellt hingegen verschiedene Plugins bereit, die zur Überwachung von verschiedenen Metriken dienen. Durch das Hinzufügen neuer Plugins können weitere Metriken überwacht werden, so dass es sich bei M4Cloud auch um einen adaptiven und erweiterbaren Ansatz handelt.

Der Ansatz CASViD von Emeakaroha et al. [28] dient zur schnellen Erkennung von Verletzungen der SLAs in Cloud Computing Umgebungen und ermöglicht die Erkennung von Verstößen auf Anwendungsebene. CASViD basiert auf einem Manager/Agenten-Modell. Zunächst werden in einem SLA Management Framework diejenigen Ressourcen definiert, die abhängig von den vereinbarten SLAs überwacht werden sollen. Der Ansatz ist somit anpassbar an die zu überwachenden Ressourcen. Die SLAs werden in dieser Hinsicht in einer Datenbank gespeichert und können jederzeit abgerufen werden. Zusätzlich werden im SLA Management Framework für jeden Teilbereich der SLAs Schwellwerte definiert. Dies dient zur rechtzeitigen Erkennung von möglichen SLA Verstößen. Die Informationen zur Bestimmung eines etwaigen Verstoßes entstammen einem integrierten Überwachungsansatz. Das Framework besteht aus zwei Komponenten. Die *Management Node*-Komponente erhält Informationen vom SLA Management Framework zu den Ressourcen, die überwacht werden sollen und greift im Anschluss daran periodisch auf eine große Anzahl an Monitor-Agenten zurück. Diese implementieren die Methodiken zur Überwachung der spezifizierten Ressourcen in der Cloud Computing Umgebung und werden in der *Processing Node*-Komponente zusammengefasst. Die Kommunikation zwischen diesen beiden Komponenten basiert auf dem Simple Network Management Protokoll (SNMP). Dies ermöglicht eine flexible als auch skalierbare Überwachungsinfrastruktur.

Der Ansatz DeSVi von Emeakaroha et al. [29] weißt große Ähnlichkeiten zum Ansatz *M4Cloud* auf. So wird beispielsweise ebenfalls eine FoSII-Infrastruktur zum SLA Management eingesetzt und es findet eine Zuordnung von Ressourcenmetriken zu SLA spezifischen Metriken statt. Die Überwachung der einzelnen Komponenten wird jedoch durch eine sogenannte DeSVi (Detecting SLA Violation Infrastructure) benannte Infrastruktur durchgeführt. Diese unterteilt sich in drei Komponenten zur Überwachung sowie

zur Erkennung von SLA Verletzungen. Die *VM Deployer*-Komponente allokiert gemäß der festgelegten SLAs und des angefragten Dienstes entsprechende Ressourcen in der Cloud Infrastruktur und unterstützt somit die Eigenschaften der Adaptivität und Elastizität. Die *Application Deployer*-Komponente ist hingegen für die Verwaltung der virtuellen Maschinen auf Anwendungsebene verantwortlich. Sie speichert für jede virtuelle Maschine eine eindeutige ID. Die *LoM2HiS*-Komponente nutzt diese IDs schließlich zur Überwachung der virtuellen Instanzen. Gleichzeitig werden Ressourcenmetriken in SLA Metriken überführt. Damit mögliche SLA Verletzungen rechtzeitig erkannt werden können, werden Schwellwerte eingesetzt, die restriktiver sind als die durch die SLAs festgelegten Werte. Zusätzlich wird eine Wissensdatenbank zur Evaluation der Daten verwendet. Dadurch sollen bei drohenden Verstößen automatische Maßnahmen zur Behebung der Probleme durchgeführt werden.

Tabelle 4.2 fasst die Charakteristika der zuvor vorgestellten Ansätze aus dem Cloud Computing Bereich zusammen.

Ansatz	Dienstebene	Komponenten	Eigenschaften
DARGOS[78]	Ganzheitliche Überwachung der Komponenten	• Node Supervisor Agent • Node Monitoring Agent	• Dezentraler Ansatz • Monitor-Agent Architektur • „Publish/Subscribe“ Protokoll
Elastic Monitoring Framework [54]	Ganzheitliche Überwachung der Komponenten	• Data • Processing • Distribution	• Dezentraler Ansatz • Peer-to-Peer Protokoll • Abfragesprache zur Definition von Regeln
M4Cloud [68]	Anwendungsebene	• Application Deployer • Application Level Monitoring • Metric Plugin Container	• Hybrider Ansatz • FoSII Infrastruktur zum Management der SLAs
CASViD [28]	Anwendungsebene	• Management Node • Processing Node	• Monitor-Agent Architektur • Simple Network Management Protokoll (SNMP)
DeSVi [29]	Ganzheitliche Überwachung der Komponenten	• VM Deployer • Application Deployer • LoM2HiS	• Hybrider Ansatz • FoSII Infrastruktur zum Management der SLAs

Tabelle 4.2: Ausgewählte Überwachungsansätze

4.4 Cloud-Überwachung in der Praxis

Im Folgenden werden ausgewählte Lösungen zur Cloud-Überwachung vorgestellt, welche in der Praxis eingesetzt werden. Die Auswahl erfolgte unter Berücksichtigung der Alexa[1] Rangordnung für die jeweilige Webseite der Lösungen. Dabei sind proprietäre Lösungen ausgeschlossen, welche ausschließlich für einen einzigen Cloud-Anbieter ausgelegt sind. Der Fokus der Betrachtung liegt auch hier insbesondere auf den Anforderungen der Skalierbarkeit, Elastizität, Adaptivität, Vollständigkeit und Erweiterbarkeit, um die besonderen Merkmale des Cloud Computings zu berücksichtigen.

Hyperic [103] war ursprünglich eine Open Source Lösung, entworfen zur Überwachung von Systemen und Applikationen. Seitdem Hyperic von VMWare übernommen wurde, wurde die Lösung noch weiter ausgebaut, damit es diese auch zur Überwachung von Cloud-Lösungen geeignet ist. Hyperic besteht aus zwei Hauptkomponenten, dem Hyperic Agent und dem Hyperic Server. Der Hyperic Agent wird direkt

[1] http://www.alexa.com/siteinfo

auf den zu überwachenden Cloud-Ressourcen betrieben. Mit Hilfe einer Reihe von Plugins kann der Hyperic-Agent QoS-Metriken auf unterschiedlichen Cloud-Ebenen bzw. Plattformen und für verschiedene Applikationen überwachen. Hyperic stellt durch die Flexibilität bzgl. der genutzten Metriken somit einen adaptiven als auch erweiterbaren Ansatz dar. Als Beispiel für eine allgemeine Metrik kann die Verfügbarkeit der Cloud-Ressourcen genannt werden, wohingegen der Zustand einer TCP Verbindung eine applikationsbezogene Metrik repräsentiert. Ein Hyperic-Agent ist auch in der Lage, die Ressourcen auf einer Cloud-Plattform zu erkennen, um diese zu überwachen. Der Hyperic Server hingegen stellt die zentrale Instanz zum Sammeln und Visualisieren von Überwachungsdaten dar. Eine Hyperic Datenbank dient zur Aufzeichnung von Überwachungsdaten und kann auf dedizierten Plattformen aufgesetzt und mit dem Hyperic Server verbunden werden. Aus Gründen der Ausfallsicherung gibt es die zudem Möglichkeit, mehrere Hyperic Server als Cluster aufzusetzen. Da mit Hyperic sowohl eine Überwachung auf Infrastruktur- als auch auf Applikationsebene möglich ist, handelt es sich um einen ganzheitlichen Ansatz. Des Weiteren wird aufgrund des Einsatzes von Agenten-basierter Technologie auch die Skalierbarkeit des Überwachungssystems ermöglicht.

Eine weitere, ganzheitliche Überwachungslösung, die für alle Cloud-Ebenen, angefangen von der Infrastruktur, über die Applikationsebene bis hin zur Überwachung des Netzwerks geeignet ist, stellt Nagios [74] dar. Der Kern der Lösung ist Nagios XI, eine zentrale Überwachungsinstanz. Nagios XI stellt Benutzern Konfigurationsassistenten zur Verfügung, mit denen erforderliche Einstellungen bzw. Berechtigungsinformationen zur Überwachung festlegt werden. Mit Nagios XI ist sowohl eine aktive als auch passive Überwachung möglich. Für eine aktive Überwachung werden Probing oder Skripte, je nach Cloud-Ressourcen und Zielen, eingesetzt. Für eine passive Überwachung werden dedizierte Agenten auf den Cloud-Ressourcen installiert, welche Plugins und Konfigurationsinformationen vom Nagios XI Server herunterladen und diese entsprechend ausführen. Nagios unterstützt somit hervorragend die hohe Dynamik im Cloud Computing durch Bereitstellung eines adaptiven und skalierbaren Überwachungsansatzes. Eine weitere besondere Eigenschaft von Nagios ist die Benutzer-Verwaltung. Diese bietet die Möglichkeit, verschiedene Rechte für unterschiedliche Benutzer zu verwalten und somit Multitenancy zu unterstützen.

Eine vollständige Überwachungslösung auf Anwendungsebene für Entwickler, d.h. für “Software as a Service”, bietet New Relic [82] Benutzern als Dienst an. Die Ressourcen, die dabei zur Überwachung der Cloud-Dienste benötigt werden, werden bei New Relic betrieben. Zur Sammlung von Überwachungsdaten müssen Software-Agenten abhängig von der genutzten Plattform und Entwicklungsumgebung installiert werden. Die Cloud-Dienst-Entwickler können die von den Agenten bereitgestellte Möglichkeit der Code-Instrumentierung nutzen, um eigene Metriken nach ihrem Bedarf zu definieren. Somit ist es möglich, auch den Ablauf (Transaktionen) der Cloud-Dienste mit zu überwachen. Somit strebt New Relic eine vollständige, erweiterbare und skalierbare Überwachung an. Eine weitere Besonderheit von New Relic ist die Möglichkeit, die Performanz der Cloud-Dienste aus Perspektive der End-Benutzer zu messen. Diese Funktionalität wird Benutzern über den Browser bereitgestellt.

Computer Associates Unified Infrastructure Management (CA UIM) [45] bietet eine Lösung zur ganzheitlichen Verwaltung der Komponenten einer Cloud-Infrastruktur. Diese Lösung ist sowohl für Private als auch für Public Cloud-Lösungen verwendbar. Die Architektur von CA UIM besteht aus den vier Komponenten Probes, Robots, Hub und Domain. Unter dem Begriff Domain wird die Zusammenführung der Überwachungsserver, Datenbanken und der Überwachungsinfrastruktur, d.h. Probes, Robots und Hub verstanden. Der Hub ist eine Software-Komponente, die die anderen Komponenten einer Domain mit einem Nachrichten-Bus verbindet. Der Nachrichten-Bus ermöglicht die Kommunikation sowohl per Request/Response als auch per Publish/Subscribe zwischen den Komponenten in der Domain. Das Publish/Subscribe-Kommunikationsschema ermöglicht dabei nicht nur die Verteilung von Überwachungsdaten an dedizierte Server, sondern auch an alle Abonnenten. Somit unterstützt die

Überwachungslösung einen hybriden Überwachungsansatz. Zuständig für die direkte Erhebung von Überwachungsdaten sind Probes und Robots. Probes sind die Komponenten, die direkt zur Erhebung von Überwachungsdaten benötigt werden. Genauer gesagt stellt jede Probe einen Softwareprozess dar, der eine bestimmte Aufgabe ausführen soll. CA UIM stellt verschiedene Probes zur Überwachung von Infrastruktur, Netzwerk und Applikationen bereit. Es besteht zudem auch die Möglichkeit, eigene Probes selbst zu erstellen. Probes werden durch Robots verwaltet, d.h. gestartet bzw. beendet. CA UIM stellt somit sowohl eine erweiterbare als auch adaptive Überwachungslösung dar. Aus Gründen der Ausfallsicherung besitzt jeder Robot eine Datenbank für Probes, in der Abweichungen von den Vorgaben und Trends der Daten gespeichert werden. Prinzipiell haben Robots eine ähnliche Funktion wie ein Software Container. Robots können zur lokalen Überwachung oder zur Fern-Überwachung aufgesetzt werden. CA UIM stellt Vorlagen zur Konfiguration zur Verfügung, so dass Überwachungsfunktionalität sehr schnell in hoch elastischen Cloud Umgebungen verteilt werden kann.

Eine Übersicht der untersuchten Überwachungslösungen ist in Tabelle 4.3 dargestellt.

Ansatz	Dienstebene	Komponenten	Eigenschaften
Hyperic [103]	Ganzheitliche Überwachung der Komponenten	• Hyperic Agent • Hyperic Server	• Automatisches Detektieren von Cloud-Ressourcen • Plugin zur Erweiterung der Monitor-Funktionalität für Agenten • Monitoring-Cluster zur Ausfallsicherung
Nagios [74]	Ganzheitliche Überwachung der Komponenten	• Nagios XI Server • (optional) Agent	• Zentraler Ansatz • Aktives und Passives Monitoring • Unterstützung zum Multitenancy
New Relic [82]	Anwendungsebene	• Software Agenten	• “Monitoring as a Service”-Ansatz • Monitoring von transaktionbasierten Cloud-Diensten • Monitoring aus der Perspektive der End-Benutzer
CA UIM [45]	Ganzheitliche Überwachung der Komponenten	• Probes • Robots • Hub • Domain	• Sowohl Request/Response als auch Publish/Subscribe Kommunikation • Hybrider Ansatz

Tabelle 4.3: Übersicht Cloud-Überwachung in der Praxis

4.5 Fazit

Insgesamt ergeben sich bei der Überwachung von Cloud SLAs verschiedene, neue Herausforderungen für den Einsatz und die Implementierung von Überwachungslösungen. Generell muss festgehalten werden, dass zur Überwachung von Cloud SLAs ein dynamisches, adaptives und elastisches System erforderlich ist. Daraus folgt, dass bestehende Ansätze aus dem Bereich der Überwachung klassischer SLAs nicht einsetzbar sind. Dennoch gelten die allgemeinen Anforderungen an Überwachungssysteme gleichermaßen auch im Cloud Computing. So ist z.B. die Genauigkeit und die Aktualität der Überwachungsdaten zu nennen. Die allgemeinen Anforderungen müssen jedoch im Cloud Computing um spezielle Aspekte erweitert werden, um den spezifischen Charakteristika aus dem Cloud Computing gerecht zu werden.

Viele der im Rahmen dieses Whitepapers untersuchten und in der Praxis eingesetzten Cloud-Überwachungslösungen (vgl. Tabelle 4.3) wurden in der Vergangenheit auf Basis etablierter Produkte zur Überwachung von verteilten Systemen weiter ausgebaut, hin zu einer ganzheitlichen Überwachung

Cloud-basierter Systeme. Aus diesem Grund sind viele der untersuchten Lösungen noch an den Bedürfnissen von Entwicklern und Administratoren ausgerichtet. Es konnten jedoch auch Lösungen identifiziert werden, die den Fokus der Überwachung auf die Anwendungsebene legen, wie beispielsweise CA UIM. Im Vergleich zu den existierenden Ansätzen in der Literatur kann festgehalten werden, dass die Lösungen in der Praxis eher einem zentralen Überwachungsansatz folgen. Deren Fokus liegt mehr auf der ganzheitlichen Überwachung der gesamten Architektur und der Unterstützung möglichst vieler Standard-Technologien, wobei Adaptivität und Skalierbarkeit im Mittelpunkt der existierenden, häufig dezentralen Lösungen in der Literatur stehen.

5 Zusammenfassung

In diesem Whitepaper wurden verschiedene Aspekte zur Anwendbarkeit von Cloud Computing in der Finanzindustrie beleuchtet. Dabei wurden Dienstgüteanforderungen, die sich aus der Literatur ergeben erarbeitet und vorgestellt. Die nicht-funktionalen Anforderungen wurden dabei kategorisiert und bewertet. Es wurde ein Kategorisierungsschema erarbeitet, was die Anforderungen in qualitative und quantitative Aspekte unterteilt. Ferner wurden Preismodelle von IT-Dienstleistungen in der Cloud untersucht und verglichen. Insbesondere regulatorischen Anforderungen haben einen hohe Bedeutung für die Finanzindustrie. Im Rahmen dieses Whitepapers wurden daher die Kernpunkte des *Bundesdatenschutzgesetz*, der *Mindestanforderungen an das Risikomanagement* sowie des *IT Sicherheitsgesetz* erarbeitet.

Sowohl bei traditionalle Diensterbringung als auch bei Cloud-Diensten sind Dienstgütegarantien Gegenstand von Service Level Agreement (SLA)s. Zur Überwachung von Cloud SLA, existieren in der Literatur unterschiedliche Ansätze. Für die Überwachung dieser Verträge sind dynamische, adaptive und elastische Überwachungssysteme erforderlich. Bestehende Ansätze aus dem Bereich der Überwachung klassischer SLAs sind daher für Cloud-Applikationen nur bedingt einsetzbar. Um den spezifischen Charakteristika des Cloud Computing gerecht werden zu können werden bestehende Ansätze um verschiedene Aspekte erweitert.

Die technischen und rechtlichen Anforderungen seitens der Finanzindustrie stellen Cloud-Dienstleister vor großen Herausforderungen. Daher setzt die Finanzindustrie derzeit noch primär auf die Bereitstellungsform der privaten Cloud. Diese vereint die Vorteile des Cloud Computing mit den Vorteilen eines unternehmenseigenen Rechenzentrums. Um zukünftig auch die Diensterbringung aus der Public Cloud in das IT-Spektrum von Banken einbeziehen zu können, müssen Cloud-Anbieter in vollem Umfang in der Lage sein, alle Dienstgüteanforderungen, insbesondere die regulatorischen Vorschriften, zu erfüllen.

Literaturverzeichnis

[1] Aceto, G. ; Botta, A. ; De Donato, W. ; Pescapè, A. : Cloud Monitoring: A Survey. In: *Computer Networks* 57 (2013), Nr. 9, S. 2093–2115

[2] Amazon: *Amazon EC2 - Preise*. Online. http://aws.amazon.com/de/ec2/pricing/. Version: 2013. – Accessed Apr. 01, 2016

[3] Ameller, D. ; Franch, X. : Service Level Agreement Monitor (SALMon). In: *Seventh International Conference on Composition-Based Software Systems (ICCBSS)*, 2008, S. 224–227

[4] Armbrust, M. ; Fox, A. ; Griffith, R. ; Joseph, A. ; Katz, R. ; Konwinski, A. ; Lee, A. ; Patterson, D. ; Rabkin, A. ; Stoica, I. ; Zaharia, M. : *Above the Clouds: A Berkeley View of Cloud Computing*. Online. http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf. Version: 2009

[5] Armbrust, M. ; Fox, A. ; Griffith, R. ; Joseph, A. D. ; Katz, R. ; Konwinski, A. ; Lee, G. ; Patterson, D. ; Rabkin, A. ; Stoica, I. u. a.: A View of Cloud Computing. In: *Communications of the ACM* 53 (2010), Nr. 4, S. 50–58

[6] BaFin: *Rundschreiben*. Online. http://www.bafin.de/SharedDocs/Downloads/DE/Rundschreiben/dl_rs1210_erlaeuterungen_ba.html. Version: 2012

[7] Baun, C. ; Kunze, M. ; Nimis, J. ; Tai, S. : *Cloud Computing*. 2. Auflage. Springer, 2010

[8] Bedner, M. ; Ackermann, T. : Schutzziele der IT-Sicherheit. In: *Datenschutz und Datensicherheit (DuD)* 34 (2010), Nr. 5, S. 323–328

[9] Berbner, R. : *Dienstgüteunterstützung für Service-orientierte Workflows*, Technische Universität Darmstadt, Diss., 2007

[10] Berger, T. G.: *Konzeption und Management von Service-Level-Agreements für IT-Dienstleistungen*, Technische Universität Darmstadt, Diss., 2005

[11] Bernhardt, M. ; Spann, M. ; Skiera, B. : Reverse Pricing. In: *Die Betriebswirtschaft* 65 (2005), Nr. 1, S. 104–107

[12] BFDI: *Bundesdatenschutzgesetz*. Online. https://www.bfdi.bund.de/bfdi_wiki/index.php/BDSG. Version: 2016

[13] Birk, D. ; Wegener, C. : Über den Wolken: Cloud Computing im Überblick. In: *Datenschutz und Datensicherheit (DuD)* 34 (2010), Nr. 9, S. 641–645

[14] BITKOM: *Cloud Computing - Evolution in der Technik, Revolution im Business*. 2009

[15] Bitkom: *Erstmals nutzt die Mehrheit der Unternehmen Cloud Computing*. Online. https://www.bitkom.org/Presse/Presseinformation/Erstmals-nutzt-die-Mehrheit-der-Unternehmen-Cloud-Computing.html. Version: 2016. – Accessed Okt. 29, 2016

[16] BMI: *IT-Sicherheitsgesetz*. Online. http://www.bmi.bund.de/DE/Nachrichten/Dossiers/ITSicherheit/itsicherheit_node.html#a-info-1

[17] Briscoe, G. ; Marinos, A. : Digital Ecosystems in the Clouds: Towards Community Cloud Computing. In: *Proceedings of the 3rd IEE International Conference on Digital Ecosystems and Technologies*, 2009, S. 103–108

[18] Buxmann, P. ; Lehmann, S. ; Draisbach, T. ; Koll, C. ; Diefenbach, H. ; Ackermann, T. : Cloud Computing und Software as a Service: Konzeption und Preisgestaltung. In: *Online-Recht 2.0: Alte Fragen ? Neue Antworten?*, 2010, S. 21–30

[19] Buxmann, P. ; Diefenbach, H. ; Hess, T. : *Die Softwareindustrie: Ökonomische Prinzipien, Strategien, Perspektiven*. Springer, 2015

[20] Caron, E. ; Rodero-Merino, L. ; Desprez, F. ; Muresan, A. : Auto-Scaling, Load Balancing and Monitoring in Commercial and Open-Source Clouds / INRIA. 2012. – Research Report

[21] Choy, S. ; Wong, B. ; Simon, G. ; Rosenberg, C. : The Brewing Storm in Cloud Gaming: A Measurement Study on Cloud to End-User Latency. In: *Proceedings of the 11th Annual Workshop on Network and Systems Support for Games*, 2012, S. 2

[22] Clement, R. ; Schreiber, D. : *Internet-Ökonomie: Grundlagen und Fallbeispiele der vernetzten Wirtschaft*. Springer, 2013

[23] Deeter, B. and Jung, R.: *Software as a Service Pricing Strategies*. Online. https://bvp.box.com/shared/static/05d7zb2zi64q7rbv1opl.pdf. Version: 2013

[24] Dittes, S. ; Urbach, N. ; Ahlemann, F. : *IT-Standardisierung - vom Lippenbekenntnis zu nachhaltigem Nutzen*. Online. https://www.researchgate.net/publication/275494636_IT-Standardisierung_-_vom_Lippenbekenntnis_zu_nachhaltigem_Nutzen. Version: 2014. – Accessed Okt. 26, 2016

[25] Dustdar, S. ; Guo, Y. ; Satzger, B. ; Truong, H.-L. : Principles of Elastic Processes. In: *Internet Computing, IEEE* 15 (2011), Nr. 5, S. 66–71

[26] Duy, T. V. T. ; Sato, Y. ; Inoguchi, Y. : Performance Evaluation Of A Green Scheduling Algorithm For Energy Savings In Cloud Computing. In: *International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*, 2010, S. 1–8

[27] Eckert, C. : *IT-Sicherheit: Konzepte-Verfahren-Protokolle*. Oldenbourg, 2013

[28] Emeakaroha, V. C. ; Ferreto, T. C. ; Netto, M. A. ; Brandic, I. ; De Rose, C. A.: Casvid: Application Level Monitoring For SLA Violation Detection In Clouds. In: *Computer Software and Applications Conference (COMPSAC)*, 2012, S. 499–508

[29] Emeakaroha, V. C. ; Netto, M. A. ; Calheiros, R. N. ; Brandic, I. ; Buyya, R. ; De Rose, C. A.: Towards Autonomic Detection Of SLA Violations In Cloud Infrastructures. In: *Future Generation Computer Systems* 28 (2012), Nr. 7, S. 1017–1029

[30] Foster, I. ; Zhao, Y. ; Raicu, I. ; Lu, S. : Cloud Computing and Grid Computing 360-Degree Compared. In: *Grid Computing Environments Workshop (GCE)*, 2008, S. 1–10

[31] Fouquet, M. ; Niedermayer, H. ; Carle, G. : Cloud Computing for the Masses. In: *Proceedings of the 1st ACM Workshop on User-Provided Networking (U-NET)*, 2009, S. 31–36

[32] Gabler: *Gabler Wirtschaftslexikon*. Online. http://wirtschaftslexikon.gabler.de/Definition/bankgeheimnis.html. – Accessed Sep. 09, 2016

[33] Garg, S. K. ; Versteeg, S. ; Buyya, R. : SMICloud: A Framework for Comparing and Ranking Cloud Services. In: *Proceedings of the 4th IEEE International Conference on Utility and Cloud Computing (UCC)*, 2011

[34] Gentry, C. : *A Fully Homomorphic Encryption Scheme*, Stanford University, Diss., 2009

[35] Glinz, M. : On Non-Functional Requirements. In: *Requirements Engineering Conference*, 2007, S. 21–26

[36] Google: *App Engine*. Online. https://cloud.google.com/products/app-engine/. Version: 2013. – Accessed Apr. 01, 2016

[37] Google: *Google Apps for Work*. Online. https://apps.google.com/intx/de/. Version: 2016. – Accessed Jan. 27, 2016

[38] Gouscos, D. ; Kalikakis, M. ; Georgiadis, P. : An Approach to Modeling Web service QoS and Provision Price. In: *Fourth International Conference on Web Information Systems Engineering Workshops*, 2003, S. 121–130

[39] Guzmán, F. ; Abimbola, T. ; Linde, F. : Pricing Information Goods. In: *Journal of Product & Brand Management* 18 (2009), Nr. 5, S. 379–384

[40] IBM: *IBM - flexible Cloud-Services für große Unternehmen in Deutschland*. Online. http://www-05.ibm.com/de/selectcloud/softlayer/. Version: 2016. – Accessed Jan. 27, 2016

[41] IBM: *IBM Bluemix Hybrid - Cloud Platform Everywhere*. Online. http://www.ibm.com/cloud-computing/bluemix/hybrid/index-b.html. Version: 2016. – Accessed Jan. 27, 2016

[42] IBM: *IBM Smarter Workforce*. Online. http://www-01.ibm.com/software/smarterworkforce/. Version: 2016. – Accessed Jan. 27, 2016

[43] IEEE: Standard Glossary of Software Engineering Terminology. In: *IEEE Standard 610*. 1990

[44] IEEE: IEEE Recommended Practice for Software Requirements Specifications. In: *IEEE Standard 830*. 1998

[45] International, C. A.: *CA Unified Infrastructure Management*. Online. http://www.ca.com/us/products/ca-unified-infrastructure-management.html. – Accessed Apr. 01, 2016

[46] Jacobson, I. ; Booch, G. ; Rumbaugh, J. ; Rumbaugh, J. ; Booch, G. : *The Unified Software Development Process*. Bd. 1. Addison-Wesley Reading, 1999

[47] Jung, H.-W. ; Kim, S.-G. ; Chung, C.-S. : Measuring Software Product Quality: A Survey of ISO/IEC 9126. In: *Software, IEEE* 21 (2004), Nr. 5, S. 88–92

[48] Kalepu, S. ; Krishnaswamy, S. ; Loke, S. W.: Verity: a QoS Metric for Selecting Web Services and Providers. In: *Fourth International Conference on Web Information Systems Engineering Workshops*, 2003, S. 131–139

[49] Karlinger, M. ; Ettmayer, K. ; Schrefl, M. : Verschlüsselung bei ausgelagerter Datenhaltung. In: *Praxis der Wirtschaftsinformatik (HMD)* 5 (2011), Nr. 48, S. 35–43

[50] Keller, A. ; Ludwig, H. : The WSLA framework: Specifying and Monitoring Service Level Agreements for Web Services. In: *Journal of Network and Systems Management* 11 (2003), Nr. 1, S. 57–81

[51] Khurshid, A. ; Al-Nayeem, A. ; Gupta, I. : *Performance Evaluation of the Illinois Cloud Computing Testbed*. 2009

[52] Klipper, S. : *Information Security Risk Management*. Bd. 1. Springer, 2011

[53] Koch, P. ; Ahlemann, F. ; Urbach, N. : *Managementorientiertes IT-Controlling und IT-Governance*. 2. Auflage. Springer, 2012

[54] König, B. ; Calero, J. A. ; Kirschnick, J. : Elastic Monitoring Framework for Cloud Infrastructures. In: *IET Communications* 6 (2012), Nr. 10, S. 1306–1315

[55] Lambrecht, A. ; Skiera, B. : Paying Too Much and Being Happy About It: Existence, Causes, and Consequences of Tariff-Choice Biases. In: *Journal of Marketing Research* 43 (2006), Nr. 2, S. 212–223

[56] Lampe, U. ; Wenge, O. ; Müller, A. ; Schaarschmidt, R. : On the Relevance of Security Risks for Cloud Adoption in the Financial Industry. In: *Proceedings of the 18th Americas Conference on Information Systems*, 2013, S. 1–8

[57] Lampe, U. : *Monetary Efficiency in Infrastructure Clouds - Solution Strategies for Workload Distribution and Auction-based Capacity Allocation*, Technische Universität Darmstadt, Diss., 2013

[58] Lampe, U. ; Hans, R. ; Seliger, M. ; Pauly, M. : Pricing in Infrastructure Clouds - An Analytical and Empirical Examination. In: *Proceedings of the 20th Americas Conference on Information Systems (AMCIS)* 19 (2014), Nr. 1, S. 1–10

[59] Lampe, U. ; Wu, Q. ; Hans, R. ; Miede, A. ; Steinmetz, R. : To Frag Or To Be Fragged - An Empirical Assessment of Latency in Cloud Gaming. In: *Proceedings of the 3rd International Conference on Cloud Computing and Services Science*, 2013, S. 5–12

[60] Lehmann, S. ; Buxmann, P. : Pricing Strategies of Software Vendors. In: *Business & Information Systems Engineering* 1 (2009), Nr. 6, S. 452–462

[61] Lehmann, S. ; Draisbach, T. ; Buxmann, P. ; Dörsam, P. : Pricing of Software as a Service- An Empirical Study in View of the Economics of Information Theory. In: *Software Business*. Springer, 2012

[62] Lehmann, S. ; Draisbach, T. ; Buxmann, P. ; Koll, C. u. a.: Pricing Models of Software as a Service Providers: Usage-Dependent Versus Usage-Independent Pricing Models / Technische Universität Darmstadt. 2010. – Forschungsbericht

[63] Lenk, A. ; Klems, M. ; Nimis, J. ; Tai, S. ; Sandholm, T. : What's Inside the Cloud? An Architectural Map of the Cloud Landscape. In: *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing ((ICSE)*, 2009, S. 23–31

[64] Limberger, C. ; Nguyen, T. A. B. ; Post, J. ; Siebenhaar, M. : QoS Model for Cloud Computing / Technische Universität Darmstadt. 2013. – Forschungsbericht

[65] Ma, K. ; Sun, R. ; Abraham, A. : Toward a Lightweight Framework for Monitoring Public Clouds. In: *Fourth International Conference on Computational Aspects of Social Networks (CASoN)*, 2012, S. 361–365

[66] Marston, S. ; Li, Z. ; Bandyopadhyay, S. ; Zhang, J. ; Ghalsasi, A. : Cloud Computing - The Business Perspective. In: *Decision Support Systems* 51 (2011), Nr. 1, S. 176–189

[67] Massonet, P. ; Naqvi, S. ; Ponsard, C. ; Latanicki, J. ; Rochwerger, B. ; Villari, M. : A Monitoring and Audit Logging Architecture for Data Location Compliance in Federated Cloud Infrastructures. In: *IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW)*, 2011, S. 1510–1517

[68] Mastelic, T. ; Emeakaroha, V. C. ; Maurer, M. ; Brandic, I. : M4Cloud-Generic Application Level Monitoring for Resource-shared Cloud Environments. In: *Proceedings of the 2nd International Conference on Cloud Computing and Services Science*, 2012, S. 522–532

[69] Mathew, G. E. ; Shields, J. ; Verma, V. : Qos Based Pricing for Web Services. In: *Web Information Systems-WISE Workshops*, 2004, S. 264–275

[70] McKinsey: *Using a Plan-Build-Run Organizational Model to Drive IT Infrastructure Objectives*. Online. http://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/using-a-plan-build-run-organizational-model-to-drive-it-infrastructure-objectives. Version: 2013. – Accessed Okt. 26, 2016

[71] Mehta, A. ; Menaria, M. ; Dangi, S. ; Rao, S. : Energy Conservation in Cloud Infrastructures. In: *IEEE International Systems Conference (SysCon)*, 2011, S. 456–460

[72] Mell, P. ; Grance, T. : The NIST Definition of Cloud Computing. Version: 2011. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. – Forschungsbericht

[73] Molina-Jimenez, C. ; Shrivastava, S. ; Crowcroft, J. ; Gevros, P. : On the Monitoring of Contractual Service Level Agreements. In: *Proceedings of the 1th IEEE International Workshop on Electronic Contracting*, 2004, S. 1–8

[74] Nagios: *Nagios*. Online. https://www.nagios.com/. Version: 2016. – Accessed Apr. 01, 2016

[75] Patel, C. ; Supekar, K. ; Lee, Y. : A QoS Oriented Framework For Adaptive Management of Web Service Based Workflows. In: *Proceedings of the 14th International Conference on Database and Expert Systems Applications (DEXA)*, 2003, S. 826–835

[76] Patel, P. ; Ranabahu, A. H. ; Sheth, A. P.: Service Level Agreement in Cloud Computing / Kno.e.sis Publications. 2009. – Forschungsbericht

[77] Petri, T. : Cloud Computing - Rechtspolitische Fragen zum Datenschutz. In: *Praxis der Informationsverarbeitung und Kommunikation (PIK)* 36 (2013), Nr. 3, S. 161–164

[78] Povedano-Molina, J. ; Lopez-Vega, J. M. ; Lopez-Soler, J. M. ; Corradi, A. ; Foschini, L. : DARGOS: A Highly Adaptable and Scalable Monitoring Architecture for Multi-Tenant Clouds. In: *Future Generation Computer Systems* 29 (2013), Nr. 8, S. 2041–2056

[79] Prodan, R. ; Ostermann, S. : A Survey and Taxonomy of Infrastructure as a Service and Web Hosting Cloud Providers. In: *Proceedings of the 10th IEEE/ACM International Conference on Grid Computing (GRID 2009)*, 2009, S. 17–25

[80] pwc: *Cloud Computing Evolution in der Wolke*. Online. http://www.pwc.de/de_DE/de/prozessoptimierung/assets/evolution-in-der-wolke-reifegrad-der-cloud-services-steigt2.pdf. Version: 2013

[81] Ran, S. : A Model for Web Services Discovery with QoS. In: *ACM SIGecom Exchanges* 4 (2003), Nr. 1, S. 1–10

[82] Relic, N. : *New Relic*. Online. http://newrelic.com/. Version: 2016. – Accessed Apr. 01, 2016

[83] Robertson, S. ; Robertson, J. : *Mastering the Requirements Process*. ACM Press/Addison-Wesley Publishing Co., 1999

[84] Robertson, S. ; Robertson, J. : *Mastering the Requirements Process: Getting Requirements Right*. Addison-Wesley Reading, 2012

[85] Robinson, W. N.: Monitoring web service requirements. In: *11th IEEE International Requirements Engineering Conference*, 2003, S. 65–74

[86] Roll, O. : *Das Aus für klassische Preismodelle*. Online. http://www.ihk-koeln.de/upload/Aus_fuer_klassischePreismodelle_Roll_Pastuch_Feb2015_38471.pdf. Version: 2015

[87] Ruppel, A. ; Stephanow, P. : *Studie zu den Mindestanforderungen an Cloud-Computing-Anbieter*. Online. http://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Veranstaltungen/Grundschutz/4GS_Tag2010/Studie_Mindestsicherheitsanforderungen.pdf. Version: 2011

[88] Sabata, B. ; Chatterjee, S. ; Davis, M. ; Sydir, J. J. ; Lawrence, T. F.: Taxonomy for QoS specifications. In: *Third International Workshop on Object-Oriented Real-Time Dependable Systems*, 1997, S. 100–107

[89] Sahai, A. ; Machiraju, V. ; Sayal, M. ; Van Moorsel, A. ; Casati, F. : Automated SLA Monitoring for Web Services. In: *Management Technologies for E-Commerce and E-Business Applications*. Springer, 2002

[90] Salesforce: *Salesforce*. Online. https://www.salesforce.com/de/form/sem/landing/sales-cloud.jsp. Version: 2016. – Accessed Jan. 30, 2016

[91] Schubert, L. ; Jeffery, K. ; Neidecker-Lutz, B. : *The Future of Cloud Computing - Opportunities For European Cloud Computing Beyond 2010*. Online. http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf. Version: 2010

[92] Shapiro, C. ; Varian, H. R.: *Information Rules: A Strategic Guide to the Network Economy*. Harvard Business Press, 2013

[93] SIIA, Macrovision, SoftSummit: *Key Trends in Software Pricing and Licensing*. Online. http://www.softsummit.com/library/reports/2008KeyTrendsSurvey.pdf. Version: 2008

[94] Spann, M. ; Zeithammer, R. ; Häubl, G. : Optimal Reverse-Pricing Mechanisms. In: *Marketing Science* 29 (2010), Nr. 6, S. 1058–1070

[95] Sperl, A. : Der Umbruch aus der Wolke - Intelligente Netzwerke & Cloud Computing. In: *Wirtschaftsblatt* 3957 (2011), S. 28

[96] Spring, J. : Monitoring Cloud Computing by Layer, Part 1. In: *Security & Privacy, IEEE* 9 (2011), Nr. 2, S. 66–68

[97] Sriram, I. ; Khajeh-Hosseini, A. : *Research Agenda in Cloud Technologies*. Online, 2010

[98] Standardization, I. O.: *Quality Management Systems - Fundamentals and Vocabulary*. International Organization for Standardization, 2000

[99] Steinmetz, R. : *Multimedia-Technologie; Grundlagen, Komponenten und Systeme*. Springer, 2000

[100] Stewart, J. M. ; Chapple, M. ; Gibson, D. : *CISSP: Certified Information Systems Security Professional Study Guide*. John Wiley & Sons, 2012

[101] Streitberger, W. ; Ruppel, A. : *Cloud Computing Sicherheit: Schutzziele, Taxonomie, Marktübersicht*. Fraunhofer-Institut für Sichere Informationstechnologie (SIT), 2009

[102] Sundararajan, A. : Nonlinear Pricing of Information Goods. In: *Management Science* 50 (2004), Nr. 12, S. 1660–1673

[103] VMWare: *Hyperic*. Online. http://www.vmware.com/products/vrealize-hyperic/. Version: 2016. – Accessed Apr. 01, 2016

[104] Wang, X. ; Vitvar, T. ; Kerrigan, M. ; Toma, I. : A QoS-Aware Selection Model for Semantic Web Services. In: *Service-Oriented Computing-ICSOC*. Springer, 2006

[105] Wenge, O. ; Lampe, U. ; Müller, A. ; Schaarschmidt, R. : Data Privacy in Cloud Computing-An Empirical Study in the Financial Industry. In: *Twentieth Americas Conference on Information Systems (AMCIS)* 1 (2014), Nr. 1, S. 1–10

[106] Zeng, L. ; Benatallah, B. ; Dumas, M. ; Kalagnanam, J. ; Sheng, Q. Z.: Quality Driven Web Services Composition. In: *Proceedings of the 12th international conference on World Wide Web*, 2003, S. 411–421

[107] Zhang, L.-J. ; Zhou, Q. : CCOA: Cloud Computing Open Architecture. In: *IEEE International Conference on Web Services (ICWS)*, 2009, S. 607–616