



BACHELORARBEIT

Fallstudie: Evaluation der Cloud-Migration eines Vereinsverwaltungssystems

Pia Sophie Lamprecht

Betreuer:
Prof. Dr. rer. pol. habil. Hans-Knud Arndt

27. Mai 2019

Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Magdeburg, am 27. Mai 2019

Pia Sophie Lamprecht

Kurzfassung

Cloud Computing ist einer der größten Trends in der IT-Branche der vergangenen Jahre. Der Umzug von On-Premise Lösungen in virtuelle Cloudinstanzen ist in den letzten Jahren zu einer vielfach durchgeführten Projektart geworden. Während Cloudanbieter mit Skalierbarkeit und Kosteneffizienz werben und viele IT-Dienstleister ganze Abteilungen mit diesen Themenstellungen beauftragen, werden insbesondere kleinere Projekte oft außen vor gelassen.

Die vorliegende Arbeit beschäftigt sich mit der Transition eines kleinen gemeinnützigen Projekts und wirft dabei die Frage auf, ob sich der zusätzliche Aufwand einer solchen Umstellung auch in kleinen Projekten lohnt. Exemplarisch wird hier der *Meilenshop*, eine Eigenentwicklung für einen örtlichen Verein, von einem virtuellen Server in die Cloudumgebung von Amazon AWS übertragen.

Dabei wird der Projekterfolg an den vier Dimensionen des Teufelsquadrats für Projektmanagement – Qualität, Leistungsumfang, Kosten und Zeit – gemessen. Hierbei umfasst der Zeitaspekt vor allem den Entwicklungsaufwand und die Zeit der Transition, während der Leistungsumfang die Möglichkeiten für kommende Weiterentwicklungen beschreibt. Ergänzt wird dies durch den *Service Measurement Index* für Cloudentwicklungen im Bereich Qualität und eine Gesamtkostenrechnung nach Lamberth und Weisbecker.

Dabei lassen die Bewertungen in dieser Arbeit den Schluss zu, dass eine Migration in die Cloud auch für kleine Projekte sinnvoll sein kann, wobei ein solcher Aufwand speziell in den Kategorien *Zeit* und *Kosten* einige Belastungen mit sich bringt. Daher kann eine solche Transition nur als Investition in die Zukunft erfolgsversprechend umgesetzt werden.

Abstract

Cloud computing is one of the most successful trends in IT for the last few years. The transition of on-premise solutions running on very own hardware to virtual cloud instances has become one of the most common project types of the last decade. Cloud providers promote their services with buzzwords like scalability and cost efficiency and many IT service companies hire loads of experts with this kind of topics. While this business dynamically grows, most small projects are being left out of this trend due to many risks in context with the transition.

This work is about the transition of a small nonprofit project and tries to give an answer to the open question if such a transition is also suitable for a small project without profit orientation. As an example the *Meilenshop* – an individual project developed for a small local organization – will be transferred from a virtual server to an Amazon AWS Cloud instance.

This project is scoped by the *Teufelsquadrat* – an adaption of the magic triangle for project management. It consists of the four dimensions *quality*, *quantity*, *costs* and *time*. While the time aspect primarily measures the effort needed for transitioning, the quantity shows the new possibilities for next steps with the new technical solution. This will be supplemented by the *Service Measurement Index* for cloud development in the quality dimension and a calculation of the *total cost of ownership* as proposed by Lamberth and Weisbecker.

The used evaluation leads to the knowledge that a migration to the cloud environment can be useful also for very small projects even though the extra effort brings additional stress especially aligned with the dimensions *time* and *costs*. Hence a transition like in the present case study is only promising if seen as investment in the future.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Ziel der Arbeit	2
1.3	Abgrenzungen	2
1.4	Aufbau der Arbeit	3
2	Grundlagen	4
2.1	Meilenwölfe	4
2.2	Meilenshop	5
2.3	IT-Infrastruktur	5
2.3.1	Server	6
2.3.2	Cloud Computing	8
2.4	Teufelsquadrat	11
2.4.1	Kosten	11
2.4.2	Zeit	13
2.4.3	Qualität	14
2.4.4	Leistungsumfang	22
2.5	Verwandte Arbeiten	23
3	Problemanalyse	24
3.1	Technische Ausgangssituation	24
3.2	Anforderungen an die neue Infrastruktur	26
3.2.1	Kernanforderungen	26
3.2.2	Erhöhung der Sicherheit	27
3.2.3	Entlastung der ehrenamtlichen Mitarbeiter	29
3.3	Wahl des Cloud-Anbieters	30
3.4	Auswahl der Qualitätsmerkmale	31
4	Umsetzung	39
4.1	Entwicklung	39
4.1.1	Containervirtualisierung	39
4.1.2	Development Pipeline	43
4.2	Betrieb	46

INHALTSVERZEICHNIS

4.2.1	Monitoring	46
4.2.2	Skalierbarkeit	47
4.2.3	Sicherheit	47
5	Evaluation	49
5.1	Kosten	49
5.2	Zeit	53
5.3	Qualität	54
5.4	Leistungsumfang	60
5.5	Fazit	62
6	Zusammenfassung & Ausblick	63

Abbildungsverzeichnis

2.1	Cloud Service-Modelle	6
2.2	Virtueller Server	8
2.3	Cloud-Begrifflichkeiten	9
2.4	Teufelsquadrat	11
2.5	TCO Eisberg	12
2.6	Service Measurement Index	16
3.1	Aufbau des Meilenshops	25
3.2	Cloud Services – Gartner Quadrat 2017	30
3.3	SMI – Essenzielle Merkmale	38
4.1	VM-Architektur und Container-Architektur	40
4.2	Container Tools	41
4.3	Beispiel einer Development Pipeline	43
4.4	Testpyramide	44
4.5	AWS Auto Scaling Ablauf	47
5.1	Kostentabelle	50
5.2	AWS – EC2 Spezifikationen	51
5.3	Beispielpreise im AWS Marketplace	60
5.4	Evaluation im Netzdiagramm	62

Tabellenverzeichnis

5.1	Evaluation der Zeit mit Story Points	53
5.2	Evaluation der Verantwortlichkeit	54
5.3	Evaluation der Agilität	56
5.4	Evaluation der Verlässlichkeit	57
5.5	Evaluation der Sicherheit & Privatsphäre	58
5.6	Evaluation der Qualität	59

Abkürzungsverzeichnis

Abkürzung	Bedeutung
ACM	AWS Certificate Manager
AGB	Allgemeine Geschäftsbedingungen
AWS	Amazon Web Services
BSI	Bundesamt für Sicherheit in der Informationstechnik
C5	Cloud Computing Compliance Controls Catalogue
CSMIC	Cloud Service Measurement Initiative Consortium
DevOps	Development & Operations
DGQ	Deutschen Gesellschaft für Qualität
DIN	Deutsches Institut für Normung
DSGVO	Datenschutzgrundverordnung
EC2	Amazon Elastic Compute Cloud
FAQ	Frequently Asked Questions
GIS	Geographic Information Systems
IaaS	Infrastructure-as-a-Service
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
IT	Informationstechnik
PaaS	Platform-as-a-Service
RDS	Amazon Relational Database Service
SaaS	Software-as-a-Service
SLA	Service Level Agreement
SMI	Service Measurement Index
SSL	Secure Sockets Layer
ST	Story Points
TCO	Total Cost of Ownership

Kapitel 1

Einleitung

*„Der PC war einmal das Zentrum.
Dieses Modell ist Vergangenheit.“*

Steve Jobs 2011, bei der Vorstellung von iCloud [IDG18]

Die Umwälzung der gesamten IT-Branche durch Cloudlösungen hat das vergangene Jahrzehnt geprägt. Lokale Daten im privaten Bereich gehören beinahe der Vergangenheit an, dennoch stellt die Transition von Unternehmenssystemen auf Cloudmodelle weiterhin eine große Herausforderung dar. Gerade kleinere Systeme besitzen oft nicht die Möglichkeit, eine solch umfassende Umwälzung anzugehen. In der vorliegenden Arbeit wird exemplarisch eine solche Umstellung vorgestellt und die Möglichkeiten und Risiken evaluiert.

1.1 Motivation

Die Administration einer Server-Infrastruktur ist eine typische Tätigkeit in beinahe jedem Unternehmen und stellt eine aufwändige und zeitraubende Aufgabe dar. Im Durchschnitt machen diese Kosten etwa 60% der Gesamtkosten eines Serverbetriebs aus [Ham10].

Neue Service-Modelle in der Cloud versprechen einen störungsfreien Betrieb ohne eigene Wartung. Dabei lässt sich die Dienstleistung der Infrastrukturbetreuung zu geringen Kosten ausgliedern und durch Virtualisierung günstig bereitstellen. Der Umsatz von Cloud Computing soll 2019 auf 214,3 Milliarden Dollar – also im Vergleich zum Jahr 2018 um 17,5% – steigen und lässt somit auf einen starken Wachstumsmarkt schließen [Sch19]. Cloud Provider versprechen viele Vorteile durch die neue Technologie. Besonders Skalierbarkeit, Replikation und eine erheblich kürzere Produkteinführungszeit gehören zu den oft angepriesenen Vorteilen der neuen Lösungen [Krc16].

Mittlerweile nutzen fast alle Großprojekte in der Softwarebranche die Cloud. Beispiele hier sind Netflix, General Electric oder Volkswagen [Amao; Amak; Tag19]. Während die Cloud ungeahnte Potenziale für Projekte dieser

Größenordnung zu bieten scheint, verbleiben besonders kleinere Projekte oft auf den Bestandssystemen. Die Entwickler scheuen hier oft den Aufwand für die Transition. Meist fehlt in kleinen Teams auch das Know-how [Con]. Oft wird aus Angst vor fehlender Kostenkontrolle und einem explodierenden Aufwand der Umstieg immer weiter verschoben. Ein gutes Beispiel für ein solches Vorgehen stellt der *Meilenshop* dar. Nun soll der Umstieg des Shops auf die Cloud gelingen. Die vorliegende Arbeit wird diese Transformation begleiten und dessen Aufwände und Kosten evaluieren.

1.2 Ziel der Arbeit

Die vorliegende Arbeit hat das Ziel, eine Entscheidungsvorlage für Projekte von ähnlicher Größe zu bieten. Dabei wird die bestehende Server-Lösung mit dem neuen cloudbasierten Ansatz verglichen. Dafür sollen die Kriterien aus dem Teufelsquadrat evaluiert werden. Diese sind:

- Kosten
- Zeit
- Qualität
- Leistungsumfang

Die Ziele können untereinander im Konflikt stehen. Dennoch wird eine Lösung unter Erfüllung der Mehrheit der Kriterien gesucht.

1.3 Abgrenzungen

Die vorliegende Arbeit soll ein Beispiel für eine Cloudtransition beschreiben. Hierbei werden der bestehende Linux V-Server von Strato sowie Amazon AWS als Cloud-Alternative betrachtet. Ein weitergehender Vergleich mit anderen Diensten und Service-Modellen erfolgt nicht.

Im Rahmen dieser Arbeit werden keine weiteren Services von den Meilenwölfen übertragen. Dazu gehören unter anderem die Website und der Mailserver. Diese werden im Laufe des kommenden Umzugs zwar ebenfalls umgestellt, sind aber kein Teil der vorliegenden Fallstudie. Außerdem wird die Weiterentwicklung des eigentlichen Shops nicht Teil der Arbeit sein. Hierbei wird nur auf die Infrastruktur eingegangen.

Die Bewertung der Transition erfolgt ausschließlich mithilfe des Teufelsquadrats. Dabei beschränkt sich beispielsweise die Evaluation der Qualität auf das SMI Modell. Andere Modelle für Softwarequalität werden zwar kurz erwähnt, finden in der nachstehenden Betrachtung aber keine Anwendung.

1.4 Aufbau der Arbeit

Nach einer kurzen Erklärung der fachlichen Grundlagen und der Ausgangssituation werden die IT-Infrastruktur und die unterschiedlichen Service-Modelle und -lösungen erklärt. Dabei wird eine genaue Definition von Cloudlösungen gegeben und der Unterschied zu herkömmlichen Modellen dargelegt. Nach der Beschreibung des Teufelsquadrats und der verwendeten Modelle für die vier Dimensionen werden auch andere verwandte Arbeiten angesprochen.

Im folgenden Kapitel werden zunächst die Ausgangssituation und die bestehenden Probleme auf technischer Ebene beleuchtet und anschließend die Anforderungen an eine Cloudlösung aufgestellt. Basierend auf den technischen Anforderungen, wird die Wahl des Cloudanbieters besprochen. Eine genauere Auswahl der notwendigen Merkmale der Qualität erfolgt mit dem SMI Katalog.

Das anschließende Kapitel Umsetzung handelt von den notwendigen Umstellungen im Bereich der Entwicklung und des Betriebs. Dabei wird besonders auf die Virtualisierung mithilfe von Containern, auf *Continuous Integration* und die verwendeten AWS-Services eingegangen.

Das Kapitel 5 beinhaltet die Evaluation der beiden Lösungen. Hierbei wird wieder auf die vier Dimensionen des Teufelsquadrats eingegangen, bevor das letzte Kapitel eine Zusammenfassung über die Arbeit und einen Ausblick auf weitere Themen gibt.

Kapitel 2

Grundlagen

Das folgende Kapitel umfasst einige wichtige Grundlagen, die für das kommende Thema unerlässlich sind, darunter fallen die Beschreibung der fachlichen Ausgangslage, die Einordnung von Server- und Cloud-Technologie und die Vorstellung der Bewertungskriterien. Außerdem werden verwandte Arbeiten beschrieben.

2.1 Meilenwölfe

Der *Meilenwölfe e.V.* ist ein seit 2015 bestehender eingetragener Verein, der sich als Fanclub des *VfL Wolfsburg* aus der Fußball-Bundesliga gegründet hat [Mei18c; VfL18]. Die Gruppe begleitet jedes Auswärtsspiel des VfL und versucht so, jede Meile auf dem Weg zu sammeln, woher auch der Name „Meilenwölfe“ stammt. Aus 19 Mitgliedern zur Gründungsversammlung wurden mittlerweile weit über 150 Mitglieder, die zusammen mit Bullis, Bussen oder dem Flugzeug zu allen Spielen der Wolfsburger fahren. Aus gemeinsamen Fahrten wurde gemeinsames Handeln. Die Meilenwölfe setzen sich auch für gemeinnützige Veranstaltungen und Aktionen – wie das jährliche Tonnenfeuer für einen guten Zweck und den Besuch eines Fußballspiels mit Flüchtlingskindern – ein. Seit 2018 sind die Meilenwölfe Teil des Stadtjugendrings in Wolfsburg [Sta18a], einer Gemeinschaft aus über 70 Vereinen und Verbänden, die sich mit Jugendarbeit und -politik beschäftigen. Außerdem haben die Meilenwölfe eine eigene Merchandise-Kollektion.

Die Meilenwölfe veröffentlichen zu jedem besuchten Spiel einen Bericht auf Facebook und der eigenen Website mit einem Gruppenfoto und gesammelten Eindrücken. Neben der Website haben die Meilenwölfe auch einen Blog, auf dem die Mitglieder Stellung zu aktuellen Themen beziehen können [Mei18a]. Seit der Saison 2018/19 stellen die Meilenwölfe einen eignen Online Shop zur Verfügung. Dieser wird im nächsten Abschnitt weiter beschrieben [Mei18b].

2.2 Meilenshop

Der *Meilenshop* wurde von drei Studierenden der Otto-von-Guericke Universität zwei Jahre lang entwickelt und im Sommer 2018 offiziell eingeführt. Im Shop können Tickets für Touren, Veranstaltungen und Merchandise der Meilenwölfe gekauft werden [Mei18b]. Der Meilenshop hat drei unterschiedliche Bereiche – den öffentlichen, den privaten und den Verwaltungsbereich. Um die Bereiche einzuteilen, wurde ein mehrschichtiges Rechte- und Rollenkonzept eingeführt. Personen, die keine Rechte besitzen, sehen lediglich den öffentlichen Teil und nur die speziell für Nicht-Mitglieder freigegebene Touren. Die Mitglieder der Meilenwölfe sehen alle Inhalte im privaten Bereich. Außerdem besitzen sie ihr eigenes Profil, welches alle Touren zeigt, an denen das Mitglied teilgenommen hat. Im Verwaltungsbereich können Touren, Veranstaltungen und Merchandise angelegt und alle Kosten verwaltet werden. Der Vorstand kann in diesem Bereich alle Mitgliederdaten einsehen und ändern. Für jede Tour bzw. Veranstaltung lässt sich eine Anmelde- und Kostenliste erstellen. So können Kapazitäten vom Vorstand besser kalkuliert werden. In der ersten Saison wurde der Shop bei den Mitgliedern gut angenommen und löste alle vorherigen Listen ab. Für die kommende Saison sollen einige neue Funktionen – wie eine Warteliste für Touren und neue Zahlungsmethoden wie PayPal – hinzugefügt werden.

Auf welchen Technologien der Meilenshop basiert und wie der Shop aufgebaut ist, beinhaltet Kapitel 3.1. Außerdem wird dort näher auf die Transformation und die neuen technischen Gegebenheiten eingegangen.

2.3 IT-Infrastruktur

IT-Infrastruktur beschreibt die Umgebung in der verwendete Anwendungen ausgeführt werden. Dabei lässt sich diese in die drei Schichten *Hardware*, *Systemsoftware* und *Anwendungssoftware* teilen [Clo17].

Die Hardware besteht aus allen physischen Teilen des Rechners, wie dem Speicher, dem Prozessor und allen anderen Systemkomponenten. Auch alle möglichen Virtualisierungslösungen werden hier eingeordnet. Eine nähere Beschreibung erfolgt in späteren Abschnitten. Die Systemschicht besteht aus allen notwendigen Softwaresystemen, die für einen reibungslosen Betrieb unerlässlich sind. Dazu gehören neben dem Betriebssystem vor allem Treiber und Laufzeitumgebungen wie Java. Die letzte Ebene beschreibt die Anwendungssoftware und wird meist von den Entwicklern gewartet. Hier finden der eigentliche Programmcode und die verwendeten Daten(-banken) ihren Platz.

Im traditionellen Modell der IT-Infrastruktur kann jede dieser Schichten von den gleichen Personen administriert werden. Meistens sind die höheren Schichten abhängig von tieferen, was gerade bei dem – manchmal notwendigen – Austausch von Hardware oft zu größeren Problemen führt [Kim16]. Da in

der traditionellen IT der Serverbetreiber meist auch der Nutzer der Systeme ist, können alle Ebenen beliebig konfiguriert werden, was jedoch trotzdem zu – internen – Abhängigkeiten führt.

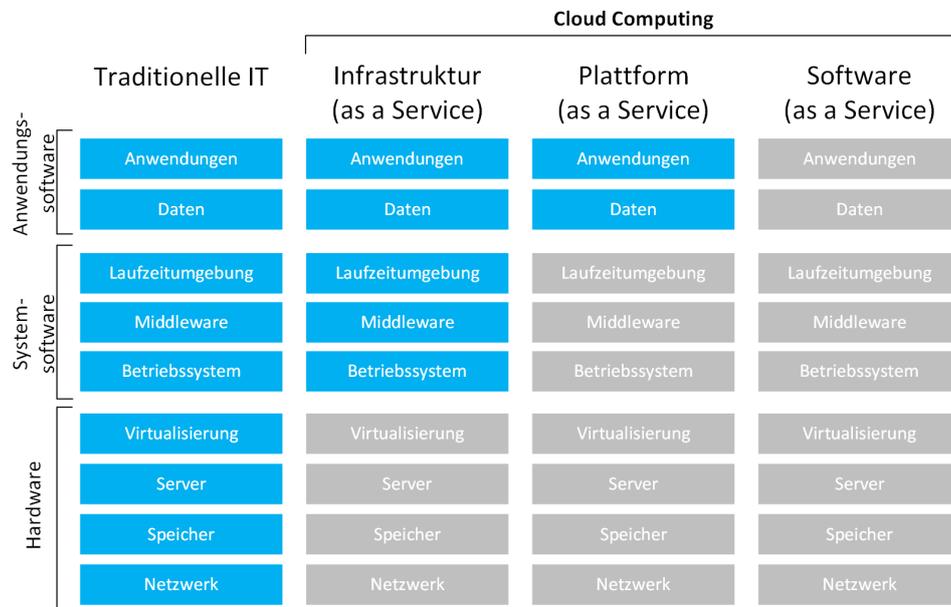


Abbildung 2.1:
Cloud Service-Modelle – ähnlich zu [Clo17]

Mithilfe der Virtualisierung ist eine Abstraktion einzelner Ebenen beim Cloud Computing möglich. So können je nach Service-Modell tiefere Ebenen von Dienstleistern zentral konfiguriert und gewartet werden, während der Nutzer nur die Ebenen verwalten muss, die er braucht [Clo17].

Im kommenden Abschnitt werden zunächst die herkömmlichen und dann die neuen Service-Modelle dargestellt.

2.3.1 Server

Ein zentraler Netzwerkrechner, der seine Ressourcen für andere Computer bereitstellt, wird als Server bezeichnet. Mit diesem Server können funktionale und infrastrukturelle Netzdienste realisiert werden [Exo]. Es gibt unterschiedliche Formen von Servern:

- i) **Inhouse-Server:** Bei einem Inhouse-Server steht das physische Gerät beim Kunden selbst. Der Kunde kümmert sich selbst um Hardware, Wartung und Aktualisierung des Servers [Min]. Somit sind alle Teile der Infrastruktur in der Hand des Kunden.

Dadurch fallen für Hosting und Verwaltung deutlich weniger Kosten an. Jedoch muss die Hardware beschafft und kontinuierlich aktualisiert

werden. Die Sicherheit des Servers und damit der vorhandenen Daten muss garantiert werden. Ein Inhouse-Server hat einen großen Vorteil: Hypersensible Daten können vor Außentstehenden geschützt werden [Min].

- ii) **Dedicated Server:** Der Server steht bei dem Betreiber eines Rechenzentrums. Beim Anmieten wird dem Kunden ein physisches Gerät zugewiesen. Dieses wird nicht mit anderen Kunden des Rechenzentrums geteilt und kann so von dem Mieter nach seinen Erwartungen eingerichtet werden. Je nach Vertrag kann der Kunde unter anderem die Hardwarekomponenten und das Betriebssystem selbst wählen [Gei18]. Das Einrichten des Servers übernimmt der Kunde.

Ein Dedicated Server bietet deutlich mehr Individualisierbarkeit als beispielsweise ein virtueller Server – bei deutlich höheren Kosten. Mithilfe der „eigenen“ Hardware kann die Server-Leistung jederzeit angepasst werden. Im Vergleich zum Virtuellen Server können so Performancevorteile entstehen. Ein Dedicated Server ist vor allem sinnvoll, wenn die Ressourcen der laufenden Anwendungen auf lange Sicht deutlich erhöht werden müssen [Gei18]. Im Vergleich zu Inhouse-Lösungen bietet diese Art den Vorteil, dass physische Sicherheit und Hardwareprobleme nicht mehr selbst administriert werden müssen. Natürlich stellt dies eine Dienstleistung dar, die auch entsprechend bezahlt werden muss.

- iii) **Virtual Server:** Bei einem Virtuellen Server steht das physische Gerät ebenfalls meist bei dem Betreiber eines Rechenzentrums. Die Ressourcen des Servers werden jedoch auf mehrere Klienten verteilt. Jeder hat eine isolierte Umgebung auf dem ein eigenes Betriebssystem zur Verfügung steht. Es gibt dabei auf den ersten Blick wenig Unterschiede für Anwendungen, da ein virtueller Server sich genau wie ein „normaler“ Rechner verhält. Dennoch führt die Tatsache, dass sich unterschiedliche Instanzen die gleichen Ressourcen teilen, zu Problemen. Während die meisten Anwendungen nur selten Lastspitzen erreichen, kann bei ungünstiger Auslastung der unterschiedlichen Instanzen ein beträchtlicher Leistungsabfall folgen. Deshalb werden V-Server bei enorm sicherheitskritischen Anwendungen oft außen vor gelassen [Bau09].

Wie in Abbildung 2.2 hat ein virtueller Server dabei reale Hardware. Auf dieser Hardware läuft ein Hypervisor. Dieser teilt den Klienten die jeweils benötigten Ressourcen zu. Auf jedem Virtuellen Server läuft ein eigenes Betriebssystem, auf dem Anwendungen installiert und ausgeführt werden können [Bau09; Kir17a].

Die Rechnerleistung kann durch mehrere Klienten besser ausgelastet werden. Das schont Ressourcen, die so flexibler und effektiver genutzt werden können. Es gibt weniger administrativen Aufwand, da die

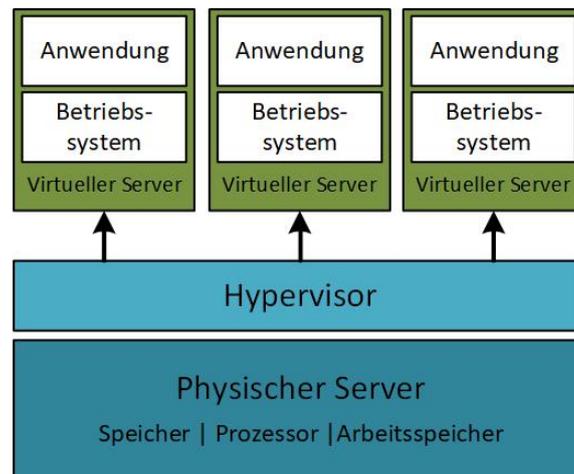


Abbildung 2.2:
Virtueller Server – ähnlich zu [Kir17b]

Wartung und Aktualisierung des Betriebssystems meist der Betreiber des Rechenzentrums übernimmt [Bau09].

Viele Betreiber von Servern bieten ebenfalls Mischformen der hier aufgeführten Serverformen. Ein Beispiel ist der *Managed Server* von Strato. Dieser ist ein Dedicated Server, der durch Strato eingerichtet und gewartet wird [Strb].

2.3.2 Cloud Computing

Nach der Definition des *National Institute of Standards and Technology* (NIST) wird *Cloud Computing* als ein Modell bezeichnet, welches einen flexiblen und bedarfsorientierten Zugriff auf gemeinsam genutzte und konfigurierbare IT-Ressourcen ermöglicht, die über das Internet bzw. ein Netzwerk abgerufen werden [Lan13; Sch15c].

Charakteristiken

Das NIST unterscheidet zwischen zwei notwendigen und drei essenziellen Eigenschaften, die ein Service haben sollte, damit dieser als Cloud Service bezeichnet werden kann [Lan13]. Die notwendigen Charakteristika sind die Netzwerkanbindung und die Ressourcenbündelung. Der bedarfsgerechte Zugriff, die Skalierbarkeit und die verbrauchsabhängige Bezahlung sind die essenziellen Eigenschaften.

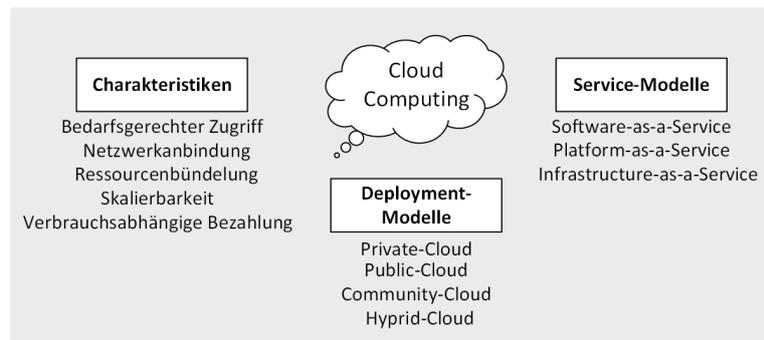


Abbildung 2.3:
Cloud-Begrifflichkeiten nach Schouten [Sch14a]

- i) **Netzwerkanbindung** (Broad Network Access): Ein Cloud-Service muss über ein Breitbandnetzwerk erreichbar sein. Es sollte eine standardisierte Kommunikationsschnittstelle nutzen, damit unterschiedliche Endgeräte wie beispielsweise Smartphones oder Laptops mit der Cloud kommunizieren können [Bie12; Krc16; Lan13].
- ii) **Ressourcenbündelung** (Resource Pooling): Durch eine Multi-Mandanten Architektur werden die Ressourcen von mehreren Kunden zur gleichen Zeit genutzt. Dabei sind die Daten und Anwendungen der Kunden komplett voneinander getrennt. Die Ressourcen werden den Kunden dynamisch je nach Bedarf zugeteilt. Durch die Dynamik der Ressourcen ist es für den Kunden nicht feststellbar wo genau die Daten gerade liegen. Es ist jedoch möglich, ein Rechenzentrum nach Land oder Region auszuwählen [Krc16; Lan13].
- iii) **Bedarfsgerechter Zugriff** (On-Demand Self-Service): Kunden können die Leistungen (beispielsweise die Speicher- oder Rechenkapazität) des Services je nach Bedarf selbstständig verändern. Die Durchführung kann automatisch ohne menschliche Interaktion geschehen [Krc16].
- iv) **Skalierbarkeit** (Elasticity): Die Veränderung der Ressourcen kann flexibel und schnell umgestellt werden. Je nach Bedarf werden die Kapazitäten teils vollautomatisch erhöht oder reduziert. Dies vermittelt dem Kunden den Eindruck, dass die Ressourcen zu jeder Zeit unbegrenzt vorhanden sind [Bie12; Krc16].
- v) **Verbrauchsabhängige Bezahlung** (Pay-per-Use): Damit Kunden eine bedarfsgerechte Abrechnung bekommen können, müssen die „kaufbaren“ Ressourcen messbar oder berechenbar sein [Bie12]. Die Kunden sollten jederzeit die verbrauchten Ressourcen und die angefallenen Kosten sichten können, um so einen möglichst transparenten Service zu erhalten [Krc16].

Service-Modelle

Es gibt vor allem drei Service-Modelle für Cloud Computing: *Infrastructure-as-a-Service*, *Platform-as-a-Service* und *Software-as-a-Service*. In der Praxis gibt es weitere Modelle, die sich aus den grundlegenden Modellen zusammensetzen [Krc16]. Die Modelle unterscheiden sich vor allem darin, welche Schichten vom Kunden verändert werden können, wie in Abbildung 2.1 zu sehen.

- i) **Intrastructure-as-a-Service:** Während Kunden bei einem angemieteten Server alle Komponenten bestimmen können, ist die Hardware bei dem Infrastrukturservice nicht veränderbar. Jedoch kann die Systemsoftware wie das Betriebssystem, die Middleware und die Laufzeitumgebung ausgewählt und verändert werden. Die Anwendungen und Daten sind vom Kunden individuell bestimmbar [Bie12].
- ii) **Platform-as-a-Service:** Bei *Platform-as-a-Service* ist lediglich die Anwendungssoftware veränderbar. Die Kunden können die Anwendungen und Daten selbst bestimmen. Hierbei gibt es eine Reihe von vorher festgelegten Laufzeitumgebungen, die meist als Buildpack zur Verfügung gestellt werden. Dies bewirkt etwas mehr Plattformunabhängigkeit, was gerade im Java-Umfeld bei der Vielzahl von möglichen Versionen das Ausführen der Software entscheidend vereinfacht [Rao16].
- iii) **Software-as-a-Service:** Kunden können bei *Software-as-a-Service* keine Komponenten selbst bestimmen. Der Anbieter stellt eine Software zur Verfügung, die auf einer Cloudinstanz ausgeführt wird. Die Cloud selbst gehört dem Anbieter und kann auch nur von ihm verändert werden. Dieses Service-Modell wird vor allem für Standardsoftware verwendet. Beispiele dafür sind Atlassian Jira oder Salesforce [Rep10; Krc16].

Zusammenfassend stellen die Cloud Service-Modelle – insbesondere der *Platform-as-a-Service*-Ansatz – somit eine interessante Möglichkeit für kleine Projekte mit fest definiertem Technologiestack dar.

2.4 Teufelsquadrat

Im IT-Projektmanagement kann der Erfolg von Projekten in drei Eckpunkten gemessen werden [Aic14].

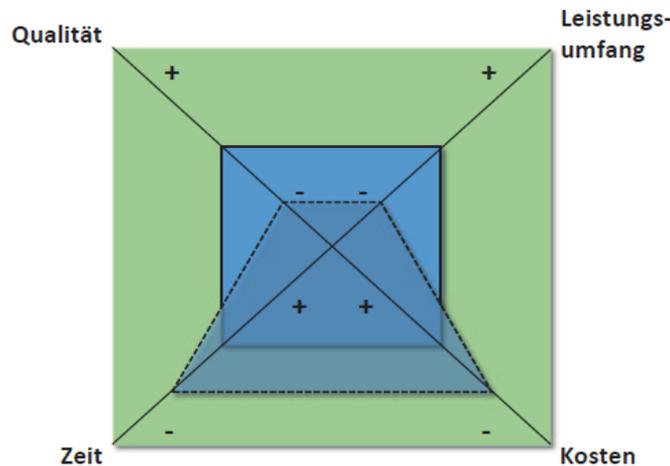


Abbildung 2.4:
Sneeds Teufelsquadrat [Aic14]

Es handelt sich dabei um *Funktionalität & Qualität*, *Zeit* und *Kosten*. Diese Punkte wurden vom Softwaretester Harry Sneed weiter zum *Teufelsquadrat* entwickelt. Er trennt den *Leistungsumfang* von der *Qualität* und erhält so vier Komponenten. Diese Trennung ermöglicht die Entwicklung eines Qualitätsbegriffs unabhängig von der Entwicklung neuer Inhalte. Gerade für Transformationsprojekte lässt dies eine bessere Messung des Projekterfolgs zu [Hum11; Fra15]. Deshalb wird dieses Modell in der vorliegenden Arbeit verwendet.

2.4.1 Kosten

Kosten sind nach Wöhe der bewertete Verzehr von Gütern und Dienstleistungen, der durch betriebliche Leistungserstellung verursacht wird [Wöh08]. Das Teufelsquadrat wird jedoch vor allem zur Bewertung von Projekten genutzt, was eine gewisse Umdeutung des Begriffs erfordert. Bei Projekten wird ein bestimmtes Budget zur Verfügung gestellt. Die Kosten eines Projekts dürfen das Budget nicht übersteigen. Unter Kosten fallen unter anderem Gehaltskosten, die Aufwendungen für Räumlichkeiten, sowie Hard- oder Software. Die Kosten sollten kontinuierlich kontrolliert werden [Aic14].

Um Projekte oder Investitionsobjekte zu vergleichen, wird vor allem die *Kostenvergleichsrechnung* genutzt. Es gibt zwei Arten von Kostenvergleichsrechnungen, der *Gesamtkostenvergleich* oder der *Stückkostenvergleich* [Obe09]. Bei der Gesamtkostenrechnung werden pro Objekt alle Kosten der gesamten

Periode zusammengetragen. Dazu zählen die Kapitalkosten und die fixen und variablen Betriebskosten. Danach findet ein Kostenvergleich der Objekte statt. Nur wenn die Ausbringungsmenge der beiden Objekte *qualitativ* und *quantitativ* gleich ist, kann die Gesamtkostenrechnung angewendet werden [Obe09; Gri04]. Der Stückkostenvergleich wird dagegen eingesetzt, wenn die Ausbringungsmengen unterschiedlich sind, da die Gesamtkosten durch die Menge der Objekte dividiert werden [Obe09].

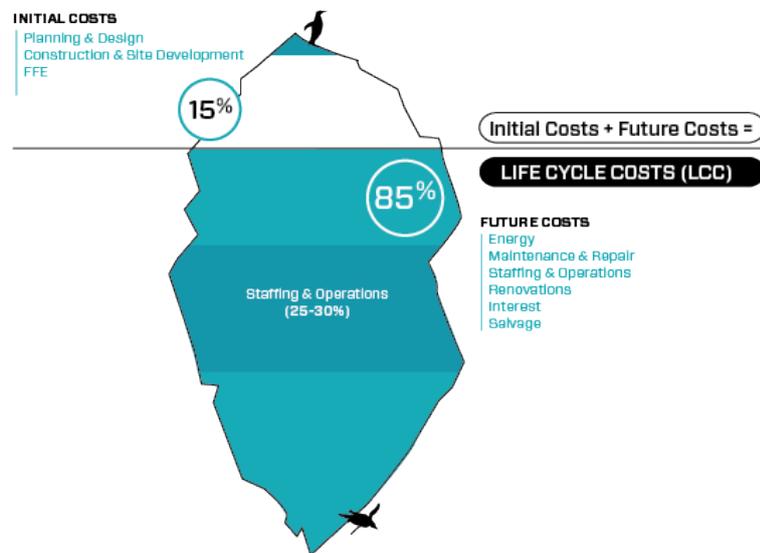


Abbildung 2.5:
TCO Eisberg [ODo15]

Um Kosten von bestehender IT-Infrastruktur mit einer Cloudlösung zu vergleichen, haben Sabrina Lamberth und Anette Weisbecker einen Gesamtkostenvergleich im Bereich der IT-Infrastruktur entwickelt [Pie10]. In der *Wirtschaftlichkeitsbetrachtungen beim Einsatz von Cloud Computing* wird eine erweiterte Nutzenwertanalyse durchgeführt, um festzustellen, ob die hausinterne IT oder die Cloudlösung genutzt werden sollte. Die Analyse besteht aus einem *Gesamtkostenvergleich* und einer *Nutzenwertanalyse* [Pie10]. Da in diesem Kapitel lediglich die Kostenseite betrachtet wird, ist nur eine Erklärung des *Gesamtkostenvergleichs* des Kosten-Nutzen-Bewertungsansatzes nötig.

Lamberth und Weisbecker schlüsseln die Kosten der IT-Infrastruktur nach der *Total Cost of Ownership*-Methode (TCO) auf [Pie10]. TCO berücksichtigt auch alle Kosten, die während des Betriebs einer IT-Lösung entstehen und nicht nur die Entwicklungskosten [Bru09]. Als Erstes werden die *direkten* und *indirekten* Kosten ermittelt. Direkte Kosten können der

IT-Infrastruktur bzw. der IT-Abteilung zugeordnet werden, beispielsweise die Hardware eines Servers. Alle Kosten, die nicht direkt einer IT-Lösung oder Abteilung zugeordnet werden können, gehören zu den indirekten Kosten [Wil04]. Lamberth und Weisbecker unterteilen die direkten in *externe Kosten aus On-Demand-Nutzung*, *IT-Investitionskosten* und *laufende Kosten*. Die letzte Unterkategorie listet alle indirekten Kosten auf. Da das Hosting und die Wartung vom Cloud-Anbieter übernommen werden, sind diese Kosten lediglich bei der traditionellen IT aufzulisten. Energiekosten, Administration & Support, Kosten für Dienstleistungen und IT-Personalkosten müssen bei beiden Alternativen berücksichtigt werden. Besonders schwierig ist es, Kosten zu beziffern, die durch mangelndes Know-how oder Unerfahrenheit mit Cloud-Technologie entstehen. Diese werden als *Opportunitätskosten durch End-User-Operations* in den indirekten Kosten dargestellt. Außerdem werden die Beiträge für Ausfälle der Cloud bzw. der traditionellen IT in den indirekten Kosten betrachtet. Durch Addition werden die Gesamtkosten der jeweiligen Alternative ermittelt und anschließend verglichen [Pie10].

2.4.2 Zeit

Im *Teufelsquadrat* ist die Zeit ein relevanter Faktor, da viele Projekte an schlechter Zeitplanung scheitern [Coo18]. Im Projektmanagement wird die Zeit oft in Meilensteine eingeteilt. Ein Meilenstein ist ein Zeitpunkt, zu dem bestimmte Aufgabenpakete erledigt werden sollten. So erhält das Team einen Überblick, in welcher Projektphase es sich befindet und wie viel Zeit für die Aufgaben bleibt [Aic14]. Um die Cloud-Angebote mit der traditionellen IT zu vergleichen, sind Meilensteine kein passendes Hilfsmittel. Die Entwicklungszeit, die benötigt wird, sowie die Zeit um die Applikation auf der Cloud oder dem Server zu installieren und zum Laufen zu bringen, wird zunächst gemessen und verglichen. Außerdem sollte hier der Zeitaufwand für kommende Weiterentwicklungen berücksichtigt werden. Oft sind diese Messungen gegenläufig, doch bei großen Zeitinvestitionen ist der Amortisationspunkt oft entsprechend später gesetzt.

Für die Schätzung von Zeit wird im Folgenden die agile *Planning Poker* Methode eingesetzt. Die zugehörigen *Story Points* bewerten den Zeitaufwand und die Komplexität einer Aufgabe relativ zu anderen, wodurch die Subjektivität verringert wird. Die Bewertungen werden meist im Konsens unter den Entwicklern getroffen. Dabei werden oft die Fibonacci-Zahlen als Wertebereich verwendet, um eine umfassende Diskussion über einen Story Point mehr oder weniger zu verhindern. Dabei werden Story Points aufsteigend nach Einschätzung der Komplexität vergeben – eine mit 5 bewertete Aufgabe ist somit größer und schwerer als eine mit der Bewertung 3. Story Points werden relativ zu anderen Aufgaben vergeben. Zwei Aufgaben innerhalb eines Teams lassen sich so vergleichen, ein Rückschluss auf absolute Zeiten ist allerdings nicht möglich. Gleichwohl sind Story Points nicht auf andere Teams übertrag-

bar. Während manche Teams in einem Sprint 15 Story Points abarbeiten, verwirklichen andere über 100. Dies ist in erster Linie der relativen Bewertung geschuldet, wodurch auch gleiche Aufgaben manchmal stark abweichende Bewertungen erhalten können. Wichtig ist, dass innerhalb eines Teams ein Konsens gefunden wird [Han16; Prö18].

2.4.3 Qualität

Qualität ist ein sehr schwer zu definierender Begriff, da diese oft eher empfunden als gemessen wird. Die *Deutsche Gesellschaft für Qualität* (DGQ) hat 2017 1010 Personen zwischen 18 und 69 Jahren befragt, was wichtig für die Qualität eines Produktes ist. 60% der Befragten gaben Langlebigkeit, 47% Sicherheit und 40% das PreisLeistungsverhältnis als wichtig für die Produktqualität an [Deu17a]. So scheint es schwierig, genau zu messen, welche Eigenschaft ein qualitativ hochwertiges Produkt haben sollte. *Statistika* fand zwischen 2015 und 2018 wiederholt heraus, dass es den meisten Kunden bei einem Produkt eher auf die Qualität als auf den Preis ankommt [Sta18b]. Damit scheint Qualität ein kontinuierlich wichtiges Merkmal aus Kundensicht zu sein.

Welche Eigenschaften als qualitativ hochwertig gesehen werden, ist jedoch stark produktabhängig. So gelten beispielsweise für Möbel andere Kriterien als für Dienstleistungen. Außerdem wird in der industriellen Fertigung oft nicht das Maximum, sondern ein wiederholbares und garantierbares Maß an Qualität angestrebt. So ist beispielsweise ein Möbelstück von IKEA nicht auf das Höchstmaß an Qualität ausgerichtet. Die Gewinnmarge für eine kontrollierte Qualität ab einem gewissen Mindeststandard liegt viel höher. Dagegen gibt es immer noch Handwerkstätten, die danach streben, ein Höchstmaß an Qualität für jedes Produkt zu erreichen. Daher würde ein Handwerker die Möbel bei IKEA als qualitativ minderwertig empfinden, obwohl die Qualität zum Preis passend erscheint. Deswegen gibt es auch in der Softwarebranche eine große Anzahl an Definitionen der Qualität.

Eine dieser Definitionen ist die *DIN-ISO-Norm 9126*. Sie definiert Softwarequalität als „die Gesamtheit der Merkmale und Merkmalswerte eines Software-Produktes, die sich auf dessen Eignung beziehen, festlegte Erfordernisse zu erfüllen“ [Hof13]. Die genaue Beschaffenheit eines Softwareproduktes wird nicht genau definiert. So müssen eher die Anforderungen, Ziele und die Umgebung eines Produktes festgelegt werden. Um konkret die Qualität eines Produktes der Softwarebranche zu messen, sollte ein *Softwarequalitätsmodell* genutzt werden, weshalb der nächste Abschnitt einen chronologischen Überblick über einige Modelle gibt.

1977 entwickelten McCall et al. Qualitätsmerkmale für General Electrics. Speziell berücksichtigt wurden die Prioritäten der Entwickler und die Anwendersicht [Sch15b]. Im Laufe der Jahre wurden mehrere Qualitätsmodelle vorgestellt, unter anderem *FURPS* 1987 von Grady und Caswell. *FURPS*

stand für die Merkmale *Functionality*, *Usability*, *Reliability*, *Performance* und *Supportability*, die die Hauptmerkmale der Qualität darstellen sollten. Mittlerweile wird oft das FURPS+ Modell verwendet, in dem alle zusätzlichen Anforderungen als *Plus* dargestellt werden [Sch15b].

1991 wurde der erste internationale Standard der Softwarequalität veröffentlicht, die *DIN-ISO-Norm 9126*. Die Softwarequalität wird in dieser Norm als der Grad der Erfüllung aller gestellten Anforderungen an die Software beschrieben [Lig09]. Diese Definition bedeutet, dass je mehr Anforderungen erfüllt werden, desto besser ist die Qualität der Software. Die Norm unterscheidet zwischen interner und externer Softwarequalität. Die interne Softwarequalität ist die Qualität aus Sicht des Entwicklers. So muss auch das Zwischenprodukt eine gute Qualität aufweisen. Die Sichtweise des Endanwenders und seine daraus abgeleiteten Anforderungen stellen die externe Sicht dar. Der Manager hat eine übergeordnete Sichtweise und kann so die Gesamtqualität beobachten [Lig09].

Die *DIN-ISO-Norm 9126* wurde 2005 durch die *ISO/IEC 25000* ersetzt. Diese Norm ist seitdem der internationale Standard der Softwarequalität. Im Vergleich zu der *DIN-ISO-Norm 9126* wurden die Sicherheit und Kompatibilität zu eigenen Merkmalen. Außerdem wurden weitere neue Teilmerkmale hinzugefügt [Int14]. Die Produktqualität, wird hier durch die folgenden Merkmale repräsentiert:

- Funktionale Tauglichkeit
- Performanz
- Kompatibilität
- Gebrauchstauglichkeit
- Sicherheit
- Wartbarkeit
- Übertragbarkeit

Ein großer Vorteil der ISO/IEC 25000 ist die internationale Relevanz der Norm. So ist die Norm eine einheitliche Basis, die es möglich macht, die Softwarequalität zu vergleichen. Es gibt jedoch keine konkrete Anleitung, Qualitätsmetrik oder einen Umsetzungsvorschlag. Dadurch kann die Evaluation stark abweichen, obwohl die gleichen Merkmale genutzt wurden. Außerdem ist die Bewertung stark vom Projekt und Thema der Überprüfung abhängig. Die ISO 25000 eignet sich eher als Checkliste, um die Vollständigkeit von Systemanforderungen und Softwaresystemtests zu überprüfen [Joh15].

Da die ISO-Norm sich nur mit der Qualität der Software befasst, stellt sie kein ausreichendes Modell für den Betrieb dar. Um nicht nur Softwarequalität

zu überprüfen, sondern auch die Infrastruktur, sollte deshalb ein anderes Qualitätsmodell genutzt werden.

Der *Service Measurement Index* (SMI) ist ein Modell zur Überprüfung der Qualität bei Cloudlösungen. Das *Cloud Service Measurement Initiative Consortium* (CSMIC) hat den SMI für Organisationen entwickelt, um ihnen zu helfen, passende Cloudlösungen für ihr spezielles Geschäftsfeld zu finden [Sie12].



Abbildung 2.6:
Service Measurement Index [Sie14]

Der SMI ist hierarchisch aufgebaut und hat zwei Ebenen. Die erste Ebene besteht aus sieben Kategorien *Accountability*, *Agility*, *Assurance*, *Financial*, *Performance*, *Security & Privacy* und *Usability*. Diese haben jeweils mindestens vier Unterkategorien [Sie12].

Die *Accountability* (Verantwortlichkeit) soll aufzeigen, wie vertrauenswürdig der Anbieter ist [Clo14]. Die Unterpunkte der *Accountability* sind:

- **Auditability (Auditierbarkeit):** Nutzer sollten Standards, Prozesse und Grundsätze des Providers verifizieren können [Clo14].
- **Compliance (Rechtmäßigkeit):** Einhaltung von notwendigen Standards, Prozessen und Grundsätzen [Clo14].

- **Contracting Experience (Erfahrungen während des Vertragschlusses):** Indikatoren für Kundenzufriedenheit und -aufwand im Zusammenhang mit dem Eingehen der Vereinbarung, die notwendig sind um den Service zu nutzen [Clo14].
- **Ease Of Doing Business (Klarheit bei der Zusammenarbeit):** Die Zufriedenheit des Kunden bei der Zusammenarbeit mit dem Provider [Clo14].
- **Governance (Kundenverwaltung):** Die Prozesse, die der Cloud-Service-Provider verwendet, um die Erwartungen, Probleme und Serviceperformanz der Kunden zu verwalten [Clo14].
- **Ownership (Inhaberverhältnis):** Die Rechte, die dem Kunden in Form von Daten, Softwarelizenzen und geistigem Eigentum in Verbindung mit dem Service eingeräumt werden [Clo14].
- **Provider Business Dtability (Wirtschaftliche Sicherheit):** Die Wahrscheinlichkeit, dass der Provider während der gesamten Vertragsdauer bestehen bleibt [Clo14].
- **Provider Certifications (Zertifizierung des Providers):** Der Anbieter führt aktuelle Zertifizierungen für Standards durch, die für die Anforderungen seiner Kunden relevant sind [Clo14].
- **Provider Contracts/SLA Verification (Service Level Agreements):** Der Anbieter stellt seinen Kunden Service Level Agreements zur Verfügung, die im Zusammenhang mit der Verwaltung des Dienstes und der Verringerung der Ausfallwahrscheinlichkeit stehen [Clo14].
- **Provider Ethicality (Ethik des Providers):** *Ethik* bezieht sich auf die Art und Weise, wie der Anbieter Geschäfte tätigt. Sie umfasst Geschäftspraktiken und ethische Grundsätze, die außerhalb der Einhaltung gesetzlicher Vorschriften liegen. Außerdem beinhaltet die Ethik faire Praktiken mit Lieferanten, Kunden und Mitarbeitern [Clo14].
- **Provider Personnel Requirements (Anforderung an Mitarbeiter des Providers):** Inwieweit Mitarbeiter von Providern über die Fähigkeiten, Erfahrung, Schulung und Zertifizierungen verfügen, die zur effektiven Bereitstellung eines Services erforderlich sind [Clo14].
- **Provider Supply Chain (Lieferkette):** Die festgelegten Service Level Agreements sollten nicht nur vom Provider, sondern auch vom Lieferanten eingehalten werden [Clo14].
- **Provider Support (Support):** Inwieweit der Provider den Kunden bei der Nutzung des Dienstes unterstützt und Fragen beantwortet, um Probleme zu beheben oder zu korrigieren [Clo14].

- **Sustainability (Nachhaltigkeit):** Die Auswirkungen des Providers auf Wirtschaft, Umwelt und Gesellschaft [Clo14].

Wie schnell und störungsfrei Änderungen, Strategien oder Taktiken neu ausgerichtet werden können, soll die *Agility* (Agilität) zeigen. Dabei sind manche Punkte vom Anbieter, andere aber direkt von der Software beeinflusst. Die Agilität wird in folgende Unterpunkte eingeteilt:

- **Adaptability (Anpassbarkeit):** Die Fähigkeit des Providers, sich an geänderte Kundenanforderungen anzupassen [Clo14].
- **Elasticity (Elastizität):** Die Fähigkeit des Providers, seinen Ressourcenverbrauch für einen Service so schnell anzupassen, dass die Kundenanforderungen erfüllt werden [Clo14].
- **Extensibility (Erweiterbarkeit):** Die Fähigkeit der bestehenden Anwendung neue Funktionen oder Dienste zu dieser hinzuzufügen. Jede Erweiterung an der bestehenden Software fällt in diese Kategorie [Clo14].
- **Flexibility (Flexibilität):** Die Möglichkeit, vordefinierte Funktionen zu einem Dienst hinzuzufügen oder zu löschen. Oft werden bestehende Features und Funktionen per Knopfdruck ein- oder ausgeschaltet [Clo14].
- **Portability (Portabilität):** Die Fähigkeit eines Kunden, eine Anwendung mit minimaler Unterbrechung leicht von einem Anbieter zu einem anderen zu verschieben [Clo14].
- **Scalability (Skalierbarkeit):** Die Fähigkeit eines Providers, den zur Verfügung stehenden Umfang zu erhöhen oder zu verringern, um die Kundenanforderungen und die vereinbarten Service Level Agreements zu erfüllen [Clo14].

Die *Assurance* (Verlässlichkeit) soll zeigen, wie wahrscheinlich es ist, dass der Service genau so läuft wie erwartet. Die Unterkategorien der Verlässlichkeit sind:

- **Availability (Verfügbarkeit):** Die Angemessenheit des Verfügbarkeitsfensters zum gegebenen Anwendungsfall sowie die Wahrscheinlichkeit, dass das Verfügbarkeitsfenster dem Kunden tatsächlich zur Verfügung gestellt wird [Clo14].
- **Maintainability (Wartbarkeit):** Die *Wartbarkeit* bezieht sich auf die Fähigkeit des Providers, Änderungen am Service vorzunehmen, um den Service in einem guten Zustand zu halten [Clo14].

- **Recoverability (Wiederherstellbarkeit):** *Wiederherstellbarkeit* bedeutet, inwieweit ein Dienst nach einer ungeplanten Unterbrechung den normalen Betriebszustand schnell wieder aufnehmen kann [Clo14].
- **Reliability (Zuverlässigkeit):** *Zuverlässigkeit* zeigt, wie eine Anwendung unter bestimmten Bedingungen, während eines bestimmten Zeitraums fehlerfrei arbeitet [Clo14].
- **Resiliency/Fault Tolerance (Widerstandsfähigkeit / Fehlertoleranz):** Die Fähigkeit der Anwendung, bei einem Ausfall einer oder mehrerer seiner Komponenten weiterhin ordnungsgemäß zu funktionieren [Clo14].
- **Service stability (Servicestabilität):** Der Grad, bis zu dem die Anwendung widerstandsfähig gegen Veränderung ist [Clo14].
- **Serviceability (Betriebstauglichkeit):** Die einfache und effiziente Wartung und Korrektur von Problemen mit der Anwendung [Clo14].

In der Kategorie *Financial* (Ökonomie) soll der Preis des Service herausgestellt werden. Es gibt vier Unterkategorien:

- **Billing Process (Rechnungsstellungsprozess):** Der Grad der Integration, der zwischen den Abrechnungssystemen des Kunden und des Anbieters verfügbar ist und die Vorhersagbarkeit periodischer Rechnungen [Clo14].
- **Cost (Kosten):** Die Gesamtkosten des Kunden für die Nutzung des Dienstes in der Gesamtzeit. Dazu gehören die Kosten für die Umstellung des Dienstes sowie wiederkehrende Kosten (z. B. monatliche Zugriffsgebühren) und nutzungsbasierte Kosten [Clo14].
- **Financial Agility (Finanzielle Agilität):** Die Flexibilität und Elastizität der finanziellen Aspekte der Dienstleistungen des Providers [Clo14].
- **Financial Structure (Finanzstruktur):** *Finanzstruktur* bedeutet, inwieweit die Preis- und Abrechnungskomponenten des Providers an die Bedürfnisse des Kunden ausgerichtet werden können [Clo14].

Performance (Performanz) gibt an, wie gut die Anwendung die nicht-funktionalen Anforderungen erfüllt. Dabei ist die pure Geschwindigkeit als Maß für Performanz genauso wichtig wie die Erfüllung anderer Anforderungen an die Software. Die Infrastruktur spielt dabei meist eine untergeordnete Rolle.

- **Accuracy (Exaktheit):** Der Umfang des Anforderungen, die durch die Anwendung erfüllt werden [Clo14].

- **Functionality (Funktionalität):** Die spezifischen Funktionen der Anwendung [Clo14].
- **Suitability (Eignung):** Die *Eignung* beschreibt, wie genau die Funktionen der Anwendung den vom Kunden benötigten Funktionen entsprechen [Clo14].
- **Interoperability (Interoperabilität):** Die Fähigkeit der Anwendung problemlos mit anderen Diensten zu interagieren [Clo14].
- **Service Response Time (Antwortzeit):** Die *Antwortzeit* ist ein Indikator für die Zeit zwischen der Anforderung eines Dienstes und der Verfügbarkeit der Antwort [Clo14].

Die *Sicherheit und Privatsphäre* (Security & Privacy) sind eine Kategorie, die sowohl die Qualität der Anwendung als auch die des Providers prüft. Ob die Sicherheit und der Datenschutz von Anwendung und Provider sichergestellt werden kann, wird in acht Unterkategorien betrachtet:

- **Access Control & Privilege (Rechte- und Rollenkonzept):** Richtlinien und Prozesse, die vom Provider verwendet werden, um sicherzustellen, dass nur das Personal, dem entsprechende Berechtigungen erteilt wurden, die Daten nutzen oder ändern kann [Clo14].
- **Data Geographic/Political (Lokalität):** Die Bewertung der individuellen Sicherheitsrisiken bezogen auf den Standort der Daten und des Services basieren auf geografischen oder politischen Belangen [Clo14].
- **Data Integrity (Datenintegrität):** Die Daten, die erstellt, verwendet und gespeichert werden, müssen so aufbewahrt werden, dass sie korrekt und gültig bleiben [Clo14].
- **Data Privacy & Data Loss (Datenschutz):** Datenschutzbestimmungen für die Verwendung und Weitergabe von Kundendaten werden vom Provider durchgesetzt. Jeder Ausfall dieser Schutzmaßnahmen wird sofort erkannt und an den Kunden gemeldet [Clo14].
- **Physical & Environmental Security (Physische Sicherheit):** Die vom Provider verwendete Richtlinien und Prozesse zum Schutz der Provider-Einrichtungen vor unberechtigtem physischem Zugriff, Beschädigung oder Interferenz [Clo14].
- **Proactive Threat & Vulnerability Management (Gefährdungsmanagement):** Es werden Vorkehrungen getroffen, um sicherzustellen, dass der Dienst vor wiederkehrenden bekannten Bedrohungen sowie vor neuen Schwachstellen geschützt ist [Clo14].

- **Retention/Disposition (Vorratsdatenspeicherung):** Die Datenaufbewahrungs- und Dispositionsprozesse des Providers erfüllen die Anforderungen des Kunden [Clo14].
- **Security Management (Sicherheitsmanagement):** Die Fähigkeit des Providers, die Anwendungs-, Daten- und Infrastruktur-Sicherheit basierend auf den Anforderungen des Kunden zu gewährleisten [Clo14].

In der Kategorie *Usability* (Benutzerfreundlichkeit) soll bewertet werden, wie einfach es ist, die Anwendung zu nutzen und zu erlernen. Diese Kategorie bezieht sich fast ausschließlich auf die Qualität der entwickelten Anwendung. Dennoch können einige Kategorien durch Infrastruktur Themen unterstützt werden. Die Unterkategorien der *Benutzerfreundlichkeit* sind:

- **Accessibility (Zugänglichkeit):** *Zugänglichkeit* zeigt auf, inwieweit die Anwendung von Benutzern mit Einschränkungen betrieben werden kann [Clo14].
- **Client Personnel Requirements (Personalanforderungen):** Die Anzahl an Mitarbeitern, die vonseiten des Entwicklungsteams notwendig sind, um die Anwendung zu betreiben. Hierbei werden Rollen, Fähigkeiten, Erfahrungen, Schulungen und Zertifizierungen mit einbezogen [Clo14].
- **Installability (Installierbarkeit):** Die *Installierbarkeit* kennzeichnet den Zeit- und Arbeitsaufwand, der erforderlich ist, um die Anwendung für die Bereitstellung vorzubereiten [Clo14].
- **Learnability (Erlernbarkeit):** Die Erlernbarkeit bezeichnet den Aufwand, den Benutzer benötigen, um die Verwendung der Anwendung zu erlernen [Clo14].
- **Operability (Bedienbarkeit):** *Bedienbarkeit* bedeutet, dass die Anwendung vom Kunden leicht genutzt werden kann [Clo14].
- **Transparency (Transparenz):** Transparenz verdeutlicht, inwieweit der Benutzer feststellen kann, wann Änderungen an einem Feature oder einer Komponente der Anwendung aufgetreten sind und ob diese Änderungen die Benutzerfreundlichkeit beeinflussen [Clo14].
- **Understandability (Verständlichkeit):** *Verständlichkeit* ist die Einfachheit, mit der die Benutzer die Fähigkeiten und den Betrieb der Anwendung begreifen können [Clo14].

Je nach Projekt und Geschäftsfeld sind unterschiedliche Attribute bzw. Messgrößen wichtiger. Da das SMI unterschiedliche Projekte unterstützen soll, können die Kategorien je nach produktabhängiger Relevanz bewertet werden.

So sollte beispielsweise bei einer Cloudlösung für eine Bank, die Sicherheit als besonders wichtig gelten und dementsprechend hoch bewertet werden. Die Priorisierung hängt dabei vom eigenen Anwendungsfall ab. Kategorien können ebenfalls mit null Prozent bewertet werden, wenn sie keine Relevanz für den Anwendungsfall haben.

Der Index besteht insgesamt aus 51 Attributen in 7 Kategorien. Durch die große Menge an Attributen ist es sinnvoll nur die relevantesten zu nutzen. Das CSMIC schlägt vor, sich auf 5 bis 7 der wichtigsten Attribute zu beschränken.

2.4.4 Leistungsumfang

Der Leistungsumfang gibt an, welche Features durch eine Lösung zur Verfügung gestellt werden. Da sich das vorliegende Projekt um den Vergleich der Infrastrukturen und nicht um die darauf laufende Software dreht, verändert sich der Leistungsumfang der eigentlichen Software durch das Projekt nicht. Dennoch bieten die beiden Lösungen unterschiedliche Möglichkeiten, die gerade aus Sicht des Entwicklungs- und Wartungsteams relevant sind.

Der Leistungsumfang zeigt dabei, welche zusätzlichen Leistungen die Infrastrukturen anbieten. Vordergründig werden diejenigen betrachtet, die nicht für Anforderungen der Infrastruktur benötigt werden. Dabei ist die Quantität und die Diversität der Services relevant. Die zusätzlichen Leistungen können später zur Erweiterung der vorhandenen Software genutzt werden. Je mehr Leistungen vorhanden sind, desto mehr Möglichkeiten kann die Infrastruktur bieten.

2.5 Verwandte Arbeiten

Im Kontext der Qualitätsermittlung für Cloud-Computing haben Schubert, Kläs und Wagner das flexible Qualitätsmodell von Quamoco genutzt, um ein Framework zu entwickeln, das die Qualität von Software bewertet [Sch14b]. Mithilfe der erstellten Visualisierung, kann die Qualität der jeweiligen Software gut dargestellt werden. Außerdem wurde das Projekt *USecureD* von der HK Business Solutions GmbH, der Technische Hochschule Köln und dem Ministerium für Wirtschaft und Energie ins Leben gerufen, um ein Qualitätsmodell, mit dem Hauptmerkmal „Usable Security“, zu entwickeln [Sch15a]. Vor allem sollte hierbei eine Richtlinie für Hersteller- bzw. Anwenderunternehmen im Bereich *Usable Security* erstellt werden. Anlehnend an die genutzte Arbeit über den Service Measurement Index, haben Garg, Versteeg und Buyya für einzelne Punkte des SMI Funktionen entwickelt, um die Merkmale vergleichbarer zu gestalten [Gar13]. Diese Merkmale können jedoch lediglich bei der Cloud zugrunde gelegt werden und betrachten die Belange des Servers nicht. Hilfreich waren ebenfalls die Überlegungen von Vossen, Haselmann und Hoeren, die eine Entscheidungshilfen für Cloud-Computing entwickelt haben [Vos13]. Als Letztes sollte die Arbeit von Bachmann erwähnt werden, die die grundsätzliche Idee und Herangehensweise von einer Migration auf die Cloud vermittelt [Bac12].

Kapitel 3

Problemanalyse

In diesem Kapitel wird die Ausgangssituation des Meilenshops und seine Umsetzung auf technischer Ebene besprochen. Danach werden die daraus entstandenen Probleme thematisiert und eine Analyse der neuen Anforderungen angestoßen.

3.1 Technische Ausgangssituation

Wie in den Grundlagen beschrieben ist der Meilenshop ein Shopsystem, zur Anmeldung für Touren & Veranstaltungen, zum Kauf von Merchandise und zur Verwaltung des Vereins. Hier werden die Komponenten und verwendeten Techniken des Shops für *Frontend* und *Backend* vorgestellt.

Zur Gestaltung des Frontends wurden die Programmiersprache *TypeScript* verwendet, die auf *JavaScript* basiert. Als Framework wurde *Angular* genutzt. Zur Ausführung des *JavaEE*-Codes für das Backend wird Apache Tomcat genutzt. In der Abbildung 3.1 werden alle Schnittstellen dargestellt. Das Backend kommuniziert mit dem Frontend durch REST-Schnittstellen. Das Communication Package verwendet den externen Mailserver der Meilenwölfe. Um Bilder in Ordnern auf dem Dateisystem zu speichern, löschen und zu updaten (CRUD: Create, Read, Update, Delete) wird ein Util-Package genutzt. Alle Daten werden in einer MySQL-Datenbank gespeichert und verwaltet. Zur Kommunikation mit der Datenbank wird die Java Persistence API genutzt, die auf Java Database Connectivity (JDBC) basiert. Der Shop wird aktuell auf einem Linux-V-Server von Strato ausgeführt.

Der Server kostet 14,00 Euro pro Monat und die Vertragslaufzeit beträgt 12 Monate. Zur Verfügung stehen 2 CPU vCores, 4 GB RAM und als Speicher eine 500 GB SSD. Die Netzwerkanbindung liegt bei bis zu 500 Mbit/s. Außerdem ist ein SSL-Zertifikat inklusive, das zur Verschlüsselung der Verbindung benötigt wird. Hierbei ist es allerdings nicht möglich, eine Subdomain damit auszustatten, da das Zertifikat keine Wildcards unterstützt. Im Moment werden 10,8 GB für E-Mails und 801,6 MB als Speicher genutzt.

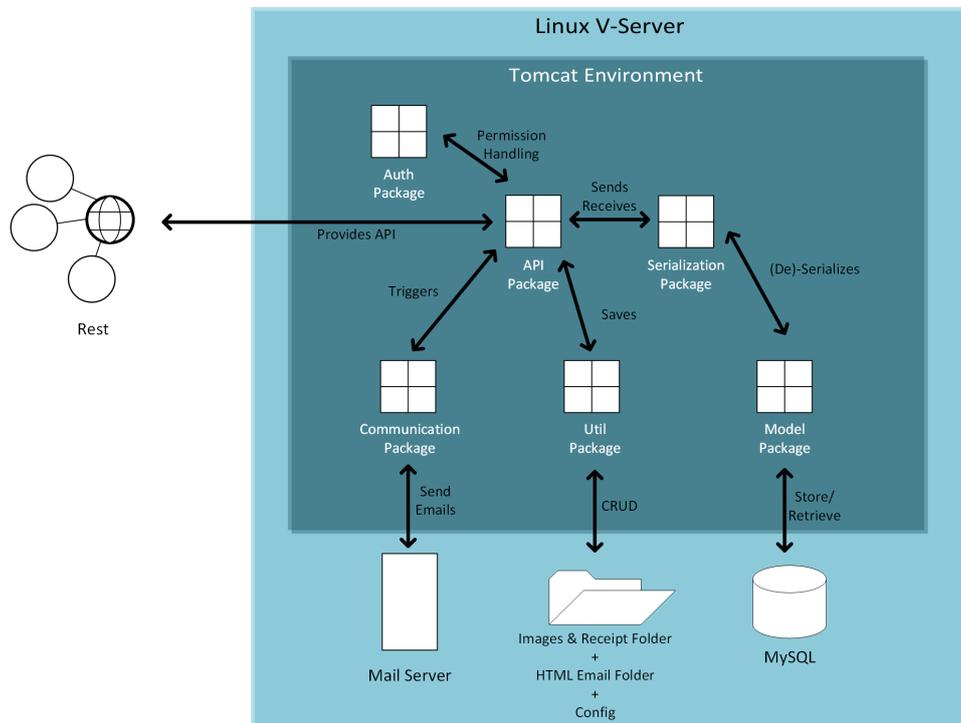


Abbildung 3.1:
Aufbau des Meilenshops

Außerdem laufen zwei Domains und acht Subdomains auf dem Server. Die eine Domain wird für die Website der Meilenwölfe genutzt, die Andere für den Shop. Ein 24/7 Experten Support kann für 8 Euro zusätzlich dazu gebucht werden. Die Zubuchung von Domains ist jederzeit kostenpflichtig möglich. Eine Veränderung des Speicherplatzes oder der CPU jedoch nur, wenn ein anderer Server ausgewählt wird.

Welche Anforderungen an die Cloudlösung vom Entwicklungsteam und den Meilenwölfen gestellt werden, wird im nächsten Kapitel besprochen.

3.2 Anforderungen an die neue Infrastruktur

Die neue Cloudlösung sollte einige Bedingungen erfüllen, welche in erster Linie die Sicherheit erhöhen und die ehrenamtlichen Mitarbeiter entlasten sollen. Außerdem gibt es noch grundlegende Anforderungen, die die Lösung erfüllen sollte. Die Anforderungen wurden vom Entwicklerteam aufgestellt.

3.2.1 Kernanforderungen

Die folgenden Punkte bilden die wichtigsten und grundlegendsten Anforderungen des Kunden an die Infrastruktur ab. Während einige von der aktuellen Lösung unterstützt werden, müssen andere erst durch den Wechsel erfüllt werden.

- **Der Standort des Cloud-Rechenzentrums sollte in einem EU-Mitgliedsstaat liegen**
Da die Meilenwölfe sensible Daten der eigenen Mitglieder und ggf. Mitfahrer verarbeiten, sind sie für den Schutz dieser Daten verantwortlich. Wenn das Rechenzentrum des Cloud-Providers in einem Land der Europäischen Union liegt, muss die *Datenschutzgrundverordnung* (DSGVO) eingehalten werden. Die DSGVO schützt die personenbezogenen Daten der Meilenwölfe und deren Verarbeitung vor Missbrauch.
- **Alle Daten und Entwicklungen bleiben im Eigentum**
Der Meilenshop, die personenbezogenen und alle anderen Anwendungsdaten sollten im Eigentum der Meilenwölfe bleiben. Wenn die Eigentumsrechte nicht bei den Meilenwölfen liegen, kann der Cloud-Anbieter die Daten verändern, verkaufen und weitergeben, was einen unerlaubten Eingriff in die Datenschutzvereinbarung der Meilenwölfe darstellt.
- **Verfügbarkeit von 99%**
Das Maß für den Grad, in dem festgelegten Anforderungen zwischen dem Anbieter und dem Kunden in einem vereinbarten Messzeitraum erfüllt werden, ist die Verfügbarkeit [Krc16]. Während die Verfügbarkeit der eigentlichen Anwendung bei einem ehrenamtlichen Team nicht vollständig sichergestellt werden kann, sollte zumindest die Verfügbarkeit der Infrastruktur bei einem respektablen Wert liegen. Bei dem vergleichbaren Strato-Server liegt die Verfügbarkeit bei mehr als 99% [Rei18a]. Die Eigenschaften der neue Lösung sollten nicht schlechter als die der Alten sein, daher sollte die Verfügbarkeit ebenfalls bei mindestens 99% liegen.

- **Kosten**

Die Kosten der neuen Lösung sollten die der Alten möglichst nicht oder nur zu einem respektablen Maß übersteigen. Da die Wirtschaftskraft der Meilenwölfe einzig durch laufende Beiträge gesichert wird, stellen die Kosten eine wichtige Anforderung dar.

- **Standard-Technologien**

Die Cloud und deren Applikationen sollten Technologien unterstützen, die auch in näherer Zukunft noch nutzbar und aktuell sind, wie beispielsweise JavaScript oder Java. So kann sichergestellt werden, dass der Meilenshop unterstützt wird. Selbstverständlich liegt hierbei das Hauptaugenmerk auf dem bestehenden Technologiestack.

- **Monitoring**

In der aktuellen Lösung gibt es keine einfache Möglichkeit, einen Überblick über die verbrauchten Ressourcen zu erhalten. Die neue Lösung sollte Monitoring unterstützen, da so nachvollzogen werden kann, wie die Ressourcen des Shops genutzt wurden. Ein mögliches Skalieren der Ressourcen wird so ebenfalls möglich.

3.2.2 Erhöhung der Sicherheit

Alle Anforderungen, die zum Anstieg der Sicherheit aufgestellt wurden, folgen nun. Der aktuelle Server ist nur durch das ehrenamtliche Team abgesichert und dabei auf regelmäßige Handarbeit angewiesen.

- **Automatisierte Backups**

Bei einem unvorhersehbaren Absturz des Systems, kann ein automatisiertes Backup den ehrenamtlichen Mitarbeitern viel Arbeit ersparen, da eine Wiederherstellung des Shops schwierig und zeitraubend ist. Nicht automatisierte Backups werden unregelmäßig durchgeführt und können so zu einem ungewollten Informationsverlust führen.

- **Zertifizierungen von Sicherheitsstandards**

Der Cloud-Anbieter sollte branchenübliche Zertifikate nachweisen können, die dem Kunden Vertrauen signalisieren. Da die Anzahl der möglichen Zertifikate hoch ist, wurden zwei Zertifizierungen ausgewählt, die der Anbieter nachweisen sollte: Das erste Zertifikat ist der internationale Sicherheitsstandard *ISO/IEC 27001: 2013*. Dieser wird von unabhängigen Auditoren in regelmäßigen Abständen überprüft. Die ISO-Norm soll das Vertrauen der Kunden in den Anbieter stärken und den Informationsbestand sichern, indem regelmäßige Sicherheitskontrollen durchgeführt werden. In der Norm werden die Anforderungen an dokumentierte Informationssicherheits-Managementsysteme beschrieben, die sich jeweils an die Risiken des Betriebs richten [Deu17b; Jen14].

Das zweite Zertifikat ist der *Cloud Computing Compliance Controls Catalogue* (C5). Dieser wurde vom *Bundesamt für Sicherheit in der Informationstechnik* (BSI) als eine Basis für Cloud Security entwickelt [Kom16]. C5 ist in 17 Bereiche eingeteilt, die an gängigen Standards, wie die *ISO/IEC 27001*, Controls Matrix der Cloud Security Alliance und diverse BSI-Veröffentlichungen angelehnt wurden [Kom16]. Das BSI hat verschiedene Anforderungen für die Zusammenarbeit mit Cloudanbietern aufgestellt, um für die Kunden des Providers ein Mindestmaß an Sicherheit zu gewährleisten [Bun17]. Damit stellt das C5 in gewisser Weise eine Erweiterung zur ISO-Norm dar. Da es sich hierbei um eine Zertifizierung für Cloud Infrastruktur handelt, ist eine Bewertung der Bestandslösung hiermit nicht möglich. Dennoch stellt dieses Zertifikat im Hinblick auf die kommenden Erweiterungen einen wichtigen Anhaltspunkt dar.

- **Sicherheitsmaßnahmen**

Während Zertifizierungen eine gewisse Basis an Sicherheitsmaßnahmen implizieren, gibt es jedoch eine Reihe von Tätigkeiten, die darüber hinaus Relevanz für die Sicherheit haben. Dazu gehören mögliche Maßnahmen, die vom physischen Schutz über Schulungen der Mitarbeiter bis hin zur Unterstützung bei dem Schutz der Daten des Kunden reichen. Diese gewisse Proaktivität ist schwer messbar, aber in diesem Kontext sehr wichtig, da die Meilenwölfe nur wenige eigene Sicherheitsmaßnahmen installieren können.

- **Skalierbarkeit & Elastizität von Ressourcen**

Die elementaren Fähigkeiten von Cloud-Services sind die Skalierbarkeit und Elastizität. Bei der Skalierbarkeit können die Ressourcen der Cloud in kürzester Zeit erhöht oder verringert werden, ohne dabei einen spürbaren Übergang zu erzeugen. Bei Veränderung der Auslastung, können beispielsweise Speicher oder Rechenleistungen verändert werden. Wenn der Shop stark ausgelastet wird, kann dies mithilfe der Elastizität der Infrastruktur ausgeglichen werden.

3.2.3 Entlastung der ehrenamtlichen Mitarbeiter

Die folgenden Anforderungen wurden aufgestellt, um die ehrenamtlichen Mitarbeiter zu entlasten. Da diese keinen Lohn oder sonstige Vergünstigungen für die Entwicklung und Wartung des Shops erhalten, sollte ihnen ein Teil der Arbeit durch den Serviceanbieter abgenommen werden.

- **Bedienbarkeit & Erlernbarkeit**

Die Lösung sollte für die ehrenamtlichen Mitarbeiter möglichst einfach zu erlernen und zu bedienen sein.

- **Wiederherstellbarkeit**

Es sollte möglichst einfach sein, den Shop nach ungeplanten Ausfällen und eventuellen Datenverlust ohne aufwendige Probleme wiederherzustellen.

- **Wartungsarbeiten**

Die Wartungsarbeiten an der Cloud sollten einfach und leicht durchführbar sein.

- **Automatisierte Erneuerung von Zertifikaten**

Ein Shopsystem funktioniert nur, solange der Anwender dem Anbieter vertraut. Eine wichtige Anforderung ist hier ein SSL-Zertifikat, welches die verschlüsselte Verbindung zu der Website ermöglicht. Da solche Zertifikate ein regelmäßiges Ablaufdatum besitzen, wird oft manuelle Arbeit notwendig. Eine Automatisierung dieses Austausches wäre wünschenswert.

- **Releasezyklen**

Durch die umfangreichen Arbeiten, die notwendig sind, um eine neue Version des Shops zu veröffentlichen, werden schnelle Anpassungen an der laufenden Software schwierig. In der Folge können nicht alle fertig entwickelten Features zeitnah in die Produktivversion übernommen werden, wodurch ebenfalls Fehler in der laufenden Anwendung oft schwer nachvollzogen werden können. Die häufige Lösung ist hier oft die Vereinheitlichung der verschiedenen Umgebungen durch häufigere Veröffentlichungen. Die Releasezyklen des Shops sollten durch die Einführung von Docker deutlich verkürzt und das Deployment vereinfacht werden [Kim16].

3.3 Wahl des Cloud-Anbieters

In diesem Kapitel wird untersucht, welche Cloud bei dem Vergleich zum vorhandenen Strato-Server genutzt werden soll. Dazu wurden die führenden Provider in der Cloud-Branche überprüft.



Abbildung 3.2:
Cloud Services – Gartner Quadrat 2017 [Hof18]

Das Marktforschungsunternehmen Gartner veröffentlichte 2017 das „Magic Quadrant of Cloud IaaS“, das die führenden Cloudanbieter bewertet [Hof18; Rix18]. Zur Bewertung nutzte Gartner 234 Kriterien, die die Cloudlösung Amazon Web Services (AWS) zu 92% erfüllt. AWS führt damit dicht gefolgt von Microsoft mit 88% und der Google Cloud Plattform mit 70% [Hof18]. Der Marktanteil von AWS lag 2018 bei 51% [Mus18]. Damit ist AWS der Marktführer im IaaS-/PaaS-Markt. Ein Grund für den großen Marktanteil ist die große Auswahl von Services, die der Kunde nutzen kann, um Applikationen zu entwickeln, zu testen und bereitzustellen [Hof18]. Außerdem hat Amazon eine sehr breite Infrastruktur mit 16 Regionen weltweit. Damit besaß AWS 2017 mehr Rechenkapazität in der Cloud als alle anderen Wettbewerber zusammen [Hof18]. Da Amazon Web Services bedeutende Vorteile besonders im Funktionsumfang besitzt, wird AWS in dieser Arbeit als Cloudlösung genutzt.

3.4 Auswahl der Qualitätsmerkmale

In diesem Abschnitt erfolgt die Analyse der Merkmale, die zur Qualitätsbewertung genutzt werden. Die Bewertung basiert auf den Anforderungen, die das Projektteam und die *Meilenwölfe* zusammen aufgestellt haben. Die Merkmale sollen zeigen, ob die Cloud oder der Server qualitativ geeigneter für die Belange und Anforderungen sind. Der Shop wird nicht bewertet, da dieser auf beiden Lösungen mit gleicher Version läuft. Die Bewertungsstufen sind „essenziell“, „nicht essenziell“ und „irrelevant“. *Irrelevant* eingestufte Merkmale werden nicht für die Bewertung der Qualität genutzt, da Themen besprochen werden, die keine Relevanz für das Projekt haben. *Nicht essenzielle* Merkmale decken wichtige Aspekte ab, die jedoch nicht den aufgestellten Hauptanforderungen entsprechen und daher nicht weiter betrachtet werden. Alle essentiellen Merkmale werden zu Bewertung der Qualität im Kapitel 5.3 genutzt. Im folgendem Abschnitt werden alle Unterkategorien bewertet.

Verantwortlichkeit

- **Auditierbarkeit (Nicht essenziell):**

Die Überprüfbarkeit von Standards, Prozessen und Grundsätzen des Providers ist ein sinnvolles Merkmal besonders für Themen wie Datensicherheit und -integrität. Damit kann die Frage geklärt werden, ob diese Standards, wie vom Kunden und Provider versprochen, eingehalten worden sind. In einem kleinen Projekt, wie dem Meilenshop, können diese Themen jedoch auch auf anderem Wege verifiziert werden, dafür sind keine Standards und Prozesse nötig. Daher wird die *Auditierbarkeit* mit *nicht essenziell* bewertet und deshalb nicht weiter betrachtet.

- **Rechtmäßigkeit (Irrelevant):**

Bei dem Merkmal der *Rechtmäßigkeit* wird überprüft, ob notwendige Standards, Prozesse und Grundsätze eingehalten werden. Für Vereine gibt es nur wenige notwendige Standards und diese sind meist sehr allgemein, wie die Ausstellung von Rechnungen. Da die *Rechtmäßigkeit* für Vereine aus diesem Grund kaum genutzt wird, wird das Merkmal als *irrelevant* bewertet.

- **Erfahrungen während des Vertragsschlusses (Irrelevant):**

Da das Entwicklungsteam den Vertrag abschließen wird und nicht der Verein, steht die Kundenzufriedenheit im Zusammenhang mit den Vertragsvereinbarungen nicht im Fokus der Interessen des Vereins. Die Erfahrungen während des Vertragsschlusses werden daher als *irrelevant* eingestuft.

- **Klarheit bei der Zusammenarbeit (Irrelevant):**

Die Zufriedenheit des Kunden, bei der Zusammenarbeit mit dem Provider, wird als *irrelevant* bewertet, da nur das Entwicklerteam mit dem Provider zusammenarbeiten wird und nicht der Verein als Kunde.
- **Kundenverwaltung (Nicht essentiell):**

Die *Kundenverwaltung* bezieht sich vor allem auf den Support des Cloudanbieters. Ein vorhandener Support ist nicht Teil der Anforderungen, da sich das Entwicklerteam um die Belange des Shops und der Service kümmern sollte. Daher ist sie hier *nicht essentiell*.
- **Inhaberverhältnis (Essentiell):**

Das *Inhaberverhältnis* ist ein *essenzieller* Punkt des Projekts. Alle Daten und Entwicklungen müssen im Besitz der Meilenwölfe bleiben.
- **Wirtschaftliche Sicherheit (Essentiell):**

Da die Meilenwölfe eine langfristige Geschäftsbeziehung mit dem Anbieter pflegen wollen, sollte die *wirtschaftliche Sicherheit* gewährleistet sein. Beim Verkauf des Unternehmens kann es zu Datenverlust, Betriebsstörungen, Vertragsveränderungen oder -kündigungen kommen [Hän13]. Dies sollte auf jeden Fall verhindert werden, daher wird die wirtschaftliche Sicherheit als *essenziell* bewertet.
- **Zertifizierung des Providers (Essentiell):**

Der Provider sollte aktuelle Zertifizierungen nachweisen können. Die Zertifizierung zeigen, dass der Anbieter unter anderem effizient und vertrauenswürdig ist [Möl16]. Dieses Merkmal wird mit *essenziell* bewertet.
- **Service Level Agreements (Essentiell):**

Service Level Agreements sind eine Gütevereinbarung zwischen dem Provider und dem Kunden [Krc16]. Die Höhe der Service Level Ziele kann je nach Anforderung vereinbart werden. Da die Meilenwölfe keine Kosten hierfür vorsehen, sollten die Kosten so gering gehalten werden wie möglich. Die Service Level Agreements sind ein *essenzieller* Punkt des SMI.
- **Ethik des Providers (Nicht essentiell):**

Da diese Arbeit vom Vergleich der Cloud- und der Server-Lösung handelt, ist die *Ethik des Providers* schwer zu differenzieren. Die ethischen Richtlinien zum Anbieten von Rechenkapazität sind weniger kritisch als etwa im produzierenden Gewerbe. Zusätzlich lässt sich mithilfe der Standortwahl hier schon einiges beeinflussen, was schon in einem anderen Merkmal abgedeckt ist. Deshalb ist dies *kein essenzielles* Merkmal.

- **Anforderung an Mitarbeiter des Providers (Irrelevant):**
In diesem Projekt sind keine Mitarbeiter vom Provider erforderlich, deshalb wird diese Unterkategorie nicht benötigt und als *irrelevant* bewertet.
- **Lieferkette (Nicht essentiell):**
Dass die Zulieferer die vereinbarten Service Level Agreements ebenfalls einhalten ist ein Hygienefaktor. Die Lieferkette wird als *nicht essenzielles* Merkmal bewertet, da davon ausgegangen wird, dass diese Unterkategorie in den Service Levels enthalten ist.
- **Support (Nicht essentiell):**
Der Anbieter sollte einen Support für seine Lösung anbieten. Die Qualität des Supports muss dabei nicht hoch sein, beispielsweise ist eine besonders schnelle Antwortzeit nicht nötig. Daher wird dieses Merkmal als *nicht essenziell* eingestuft.
- **Nachhaltigkeit (Nicht essentiell):**
Wie die Auswirkungen des Providers auf die Umwelt und die Gesellschaft aussehen, ist ein *nicht essenzielles* Merkmal. Während das Thema der Nachhaltigkeit in den vergangenen Jahren enorme Relevanz erfahren hat, bietet das reine zur Verfügung stellen, eines Servers nur wenig Möglichkeiten, um sich hier verbindlich von Wettbewerbern abzuheben.

Für die *Verantwortlichkeit* wurden vier Merkmale als *essenziell* bewertet. Diese werden zur Bewertung der Qualität im Abschnitt 5.3 herangezogen.

Agilität

- **Anpassbarkeit (Nicht essentiell):**
Die *Anpassbarkeit* muss der Provider nicht gewährleisten, da das Entwicklungsteam für Anpassungen am Shop verantwortlich ist. Diese sind jedoch außerhalb des Scopes dieser Arbeit. Die Anpassbarkeit gehört daher nicht zu den Anforderungen und wird als *nicht essenziell* bewertet.
- **Elastizität (Essentiell):**
Die *Elastizität* der Ressourcen ist eine Kernanforderung des Projekts, daher wird sie als *essenziell* bewertet. Eine Anpassung der Ressourcen, sodass jeder Zeit neue Projekte und Tools umgesetzt werden können, ist das Ziel dieser Anforderung.
- **Erweiterbarkeit (Essentiell):**
Die *Erweiterbarkeit* ist eine *essenzielle* Anforderung. Diese sollte durch die neue Architektur des Shops unterstützt werden.

- **Flexibilität (Nicht essentiell):**

Vordefinierte Funktionen zu dem bestehenden Dienst hinzuzufügen ist ein gutes Feature. Es ist jedoch nicht geplant worden, diese Funktion in näherer Zukunft zu nutzen, daher wird die *Flexibilität* als *nicht essenziell* eingestuft.

- **Portabilität (Nicht essentiell):**

Da eine langfristige Geschäftsbeziehung geplant ist, wird ein Wechsel des Anbieters in näherer Zukunft nicht in Erwägung gezogen, daher ist diese Unterkategorie *nicht essenziell*. Die Möglichkeit einer Portierung auf eine andere Plattform ist jedoch sinnvoll.

- **Skalierbarkeit (Essentiell):**

Bisher war der Umfang der Systemauslastung schwer einzuschätzen, da wenig Monitoring möglich war. Daher ist dieser Punkt ein Kernmerkmal und wird als *essenziell* bewertet.

Die *Agilität* hat drei Merkmale, die mit *essenziell* bewertet worden sind und in die Bewertung mit einfließen.

Verlässlichkeit

- **Verfügbarkeit (Essentiell):**

Die *Verfügbarkeit* sollte überprüft werden, da die Cloud bzw. der Server nur hilfreich ist, wenn die Verfügbarkeit hoch genug ist. Diese Unterkategorie wird als *essenziell* bewertet.

- **Wartbarkeit (Essentiell):**

Wenn die Durchführung von Wartungsarbeiten vom Provider übernommen wird, werden die ehrenamtlichen Mitarbeiter entlastet und die Motivation steigt, daher ist die *Wartbarkeit* ein *essenzielles* Merkmal.

- **Wiederherstellbarkeit (Essentiell):**

Wenn der Dienst nach einer ungeplanten Unterbrechung den Betriebszustand nicht wieder aufnehmen kann, bedeutet dies Zeitverlust und Stress für die ehrenamtlichen Mitarbeiter, daher ist es *essenziell*, dass der Dienst schnell wieder benutzbar ist.

- **Zuverlässigkeit (Irrelevant):**

Während die *Zuverlässigkeit* einige infrastrukturelle Anteile besitzt, werden diese schon in anderen Kategorien abgehandelt. Alles Weitere bezieht sich hier auf den Shop. Da dieser nicht verändert wird, muss das Merkmal nicht zwischen den beiden Infrastrukturlösungen verglichen werden.

- **Widerstandsfähigkeit / Fehlertoleranz (Irrelevant):**

Wie widerstandsfähig der Shop ist, muss nicht bewertet werden, da dieser sich bei beiden Lösungen nicht unterscheidet und sich daher gleich verhält.

- **Servicestabilität (Irrelevant):**

Das Merkmal der *Servicestabilität* muss nicht besprochen werden, da dieses sich auf den Shop bezieht. Während beispielsweise die – ohnehin schon bewertete – Verfügbarkeit hier eine Rolle spielt, sind die meisten Faktoren direkt von der Codebasis abhängig, die nicht Thema dieser Evaluation ist.

- **Betriebstauglichkeit (Essentiell):**

Damit die ehrenamtlichen Mitarbeiter wenig Zeit in die Wartung und Lösung von Problemen investieren müssen, sollten alle Arbeiten einfach und effizient durchführbar sein. Da die *Betriebstauglichkeit* eine Kernanforderung ist, wird sie als *essenziell* bewertet.

Vier Merkmale wurden als „*essenziell*“ bewertet und sollten bei der Bewertung betrachtet werden.

Ökonomie

Die Kategorie *Ökonomie* wird nicht zur Bewertung der Qualität genutzt, da alle Kosten im Zuge der Evaluierung mithilfe des Teufelsquadrats erläutert werden.

Performanz

- **Exaktheit, Funktionalität, Eignung, Interoperabilität (Irrelevant):**

Die Merkmale der *Exaktheit*, *Funktionalität*, *Eignung* und *Interoperabilität* beziehen sich auf den Shop, da dieser nicht verändert wird, sind diese Unterkategorien *irrelevant* und müssen nicht weiter betrachtet werden.

- **Antwortzeit (Nicht essentiell):**

Die *Antwortzeit* erfasst die Performanz der Cloud bzw. des Servers. Da die Latenz eines Servers aber im Bereich weniger Millisekunden liegt und die restliche Codebasis nicht verändert wurde, wird dieses Merkmal deswegen als *nicht essenziell* bewertet.

In der Kategorie *Performanz* wird kein Merkmal in die tiefer gehende Bewertung übernommen.

Sicherheit & Privatsphäre

- **Rechte- und Rollenkonzept (Irrelevant):**

Das Entwicklungsteam besteht aus wenigen Mitgliedern. Da nur das Team auf die Lösung zugreifen wird, ist ein *Rechte- und Rollenkonzept* für die Infrastrukturverwaltung nicht nötig. Innerhalb des Shops ist ein solches bereits implementiert und ist aus diesem Grund in dieser Transformation nicht enthalten.

- **Lokalität (Essentiell):**

Da die Datenschutzgrundverordnung (DSGVO) erfüllt werden muss, sollte der Server in Europa stehen, damit die Kunden- und Mitglieder-daten bestmöglich geschützt sind.

- **Datenintegrität (Irrelevant):**

Die *Datenintegrität* wird durch die eigene Datenbank gewährleistet. Hier sollte es keinen Unterschied zwischen den beiden Lösungen geben, daher wird das Merkmal nicht bewertet.

- **Datenschutz (Nicht essentiell):**

Die Regelungen des *Datenschutzes* unterliegen gesetzlichen Bestimmungen und müssen daher von den Anbietern eingehalten werden. Der Datenschutz ist also ein Hygienefaktor und wird deshalb nicht weiter betrachtet. Die wichtigen Aspekte wie beispielsweise die Datenlokalität werden schon in anderen Merkmalen bewertet.

- **Physische Sicherheit (Essentiell):**

Der Provider sollte seine Einrichtungen vor physischem Zugriff, Beschädigung oder Interferenzen schützen. Da der Kunde keinen Schutz gewährleisten kann, muss der Provider diese Aufgabe übernehmen. Daher ist dies ein *essenzielles* Merkmal.

- **Gefährdungsmanagement (Essentiell):**

Bedrohungen und Schwachstellen können zu Problemen und Ausfällen führen. Wenn der Provider passende Vorkehrungen trifft, werden die ehrenamtlichen Mitarbeiter entlastet. *Gefährdungsmanagement* ist daher ein *essenzielles* Merkmal.

- **Vorratsdatenspeicherung (Irrelevant):**

Da das Entwicklungsteam keine Anforderungen für Datenaufbewahrungs- und Dispositionsprozesse aufgestellt hat, wird diese Unterkategorie nicht weiter betrachtet.

- **Sicherheitsmanagement (Nicht essentiell)**

Da es keine speziellen Sicherheitsanforderungen des Entwicklungsteams gibt, wird dieses Merkmal als *nicht essenziell* gekennzeichnet.

Drei Merkmale wurden zur Überprüfung der Qualität im Bereich *Sicherheit & Privatsphäre* ausgewählt.

Benutzerfreundlichkeit

- **Zugänglichkeit (Nicht essentiell)**

Ob der Shop auch von Benutzern mit Einschränkungen genutzt werden kann, ist kein Teil dieser Arbeit. Der Shop ist zwar nicht auf barrierefreie Benutzung optimiert, wird in diesem Zuge aber nicht verändert, womit sich hier keine Verbesserung einstellen wird.

- **Personalanforderungen (Essentiell):**

Die bevorzugte Lösung sollte mit minimalen Mitarbeiterstunden auskommen, da nur ein kleines Team mit begrenzter Zeit an der Anwendung arbeitet. Dieses Merkmal sollte auf jeden Fall überprüft werden, da es *essenziell* zur Verbesserung der Arbeitsmoral beiträgt.

- **Installierbarkeit (Nicht essentiell)**

Der Aufwand der Installation sollte gering gehalten werden. Da die Installation jedoch nur selten durchgeführt wird, ist sie im Vergleich zu den anderen Merkmalen ein *nicht essenzieller* Punkt.

- **Erlernbarkeit, Bedienbarkeit, Transparenz, Verständlichkeit (Nicht essentiell)**

Die Merkmale *Erlernbarkeit, Bedienbarkeit, Transparenz* und *Verständlichkeit* beziehen sich auf den Shop, dessen Weiterentwicklung kein Teil dieser Arbeit ist.

In der Kategorie *Benutzerfreundlichkeit* wurden die Personalanforderungen als *essenzielles* Merkmal bewertet.



Abbildung 3.3:
Essenzielle Merkmale des SMI

Im Kapitel 5.3 werden die beiden Lösungen mithilfe der essentiellen Merkmale bewertet. Die essentiellen Merkmale in der Kategorie *Verantwortlichkeit* sind das Inhaberverhältnis, die wirtschaftliche Sicherheit, die Zertifizierungen des Providers und die Service Level Agreements. Die Elastizität, die Erweiterbarkeit und die Skalierbarkeit sind die essentiellen Merkmale der Kategorie *Agilität*. Die zu bewertenden Merkmale der *Verlässlichkeit* sind die Verfügbarkeit, Wartbarkeit, Wiederherstellbarkeit und Betriebstauglichkeit. In der Kategorie *Sicherheit & Privatsphäre* sind die essentiellen Merkmale die Lokalität, die physische Sicherheit und das Gefährdungsmanagement. Die *Benutzerfreundlichkeit* hat das essentielle Merkmal Personalanforderungen.

Kapitel 4

Umsetzung

Nach der Klärung von Anforderungen, Rahmenbedingungen und Akzeptanzkriterien im letzten Kapitel beschreibt dieser Abschnitt deren Umsetzung. Dabei wird im Folgenden auf die beiden integralen Bestandteile *Entwicklung* und *Betrieb* eingegangen, die im Rahmen des aktuellen Trends zum sogenannten *DevOps* ineinander verschmelzen sollen.

4.1 Entwicklung

Die Entwicklung der neuen Lösung umfasst alle Schritte von der Aufnahme der Anforderungen bis hin zur Auslieferung der neuen Software. Hierbei wird besonders auf die Virtualisierung – mithilfe von Containern und die Einrichtung einer *Continuous Integration Pipeline* – eingegangen.

4.1.1 Containervirtualisierung

Die Virtualisierung bringt vollkommen neue Chancen im Bereich der IT-Infrastruktur. Hierbei werden mehrere physische Ressourcen zusammengefasst und als logische Einheit behandelt. Dabei findet eine Abstraktion auf die Einheiten innerhalb des Clusters statt. Dieses Cluster kann in mehrere virtuelle – physisch nicht notwendigerweise zusammenhängende – Ressourcen geteilt werden, wobei jede virtuelle Einheit eine individuelle Aufgabe übernehmen kann. Auf diese Weise können Ressourcen besser ausgelastet werden. Schwach ausgelastete Einheiten können mit anderen freien Kapazitäten gebündelt und anderweitig verwendet werden. Nicht zuletzt lässt eine solche Architektur das horizontale Skalieren von Anwendungen – also das Ausführen von mehreren baugleichen Knoten derselben Anwendung – zu und lässt den Betreiber somit viel flexibler auf veränderte Lastverteilung reagieren. Hierdurch wird sichergestellt, dass jede Anwendung genau die benötigten oder vertraglich vereinbarten Ressourcen erhält [Ben08].

In der Folge ergeben sich einige Vorteile: Die Administration logischer Cluster ist bedeutend einfacher und auch die Bereitstellung lässt sich vollständig von der physischen Ebene abstrahieren. Durch die einfachere Ressourcenverteilung wird eine Replikation und in der Folge eine hohe Verfügbarkeit bedeutend begünstigt. Außerdem wird der Versionswechsel, sowohl auf der Ebene des Betriebssystems als auch der Anwendung, erleichtert [Ben08].

Die einfachste Form der Virtualisierung stellt die virtuelle Maschine dar. Hierbei wird ein virtueller Bereich auf einem Rechner mit einem eigenen Betriebssystem ausgestattet und wie ein vollwertiger Rechner verwaltet. Wie in Abschnitt 3 beschrieben, können auf jedem physischen Server mehrere virtuelle Maschinen laufen und sich die Ressourcen im laufenden Betrieb teilen. Diese Verteilung wird von einem sogenannten Hypervisor zur Verfügung gestellt [Bau09; Kir17a].

In der Praxis wird Virtualisierung jedoch oft mit Containern durchgeführt. Diese funktionieren analog zu den namensgebenden Schiffscontainern. In der Logistik war das Verladen von Gütern über Jahrhunderte eine schwere und langwierige Aufgabe. Während der eigentliche Transport immer effizienter wurde, dauerte das Be- und Entladen der Schiffe einige Zeit. Verärgert von dieser Tatsache begann der Reeder Malcom McLean größere, standardisierte Behälter für die eigentliche Fracht zu entwickeln und damit eine größere Effizienz zu erreichen. Im Jahr 1956 baute er einen Frachter um, sodass dieser genormte, stapelbare Metallkisten transportieren konnte. Diese können wesentlich platzsparender gepackt und leichter bzw. schneller geladen und gelöscht werden, wodurch die Schifffahrt und somit die gesamte Logistikbranche einen großen Effizienzgewinn verzeichnen konnte [Jäh16; Ran12].

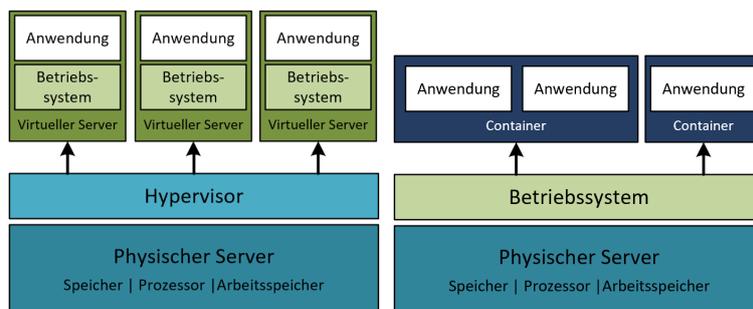
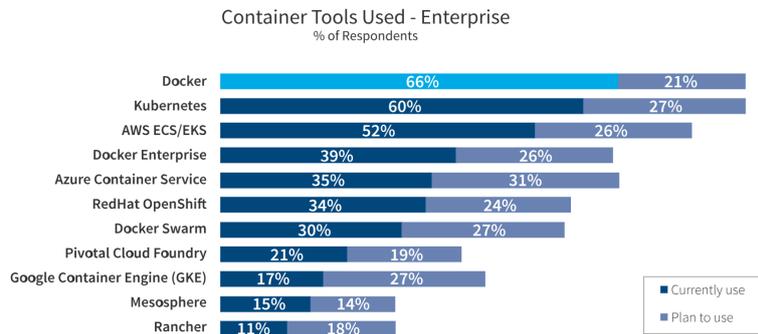


Abbildung 4.1:
VM- & Container-Architektur

Von diesen Vorteilen können auch virtuelle Container profitieren. Die oben genannte Virtualisierung findet hierbei oberhalb der Betriebssystemebene statt. Hierbei können sich sämtliche Container die physische Infrastruktur der virtualisierten Umgebung teilen und jeweils eine eigene Laufzeitumgebung in einem geschlossenen, isolierten Bereich zur Verfügung stellen. Dieser sprichwörtliche Sandkasten lässt es zu, dass Anwendungen ohne unbeabsich-

tigte Wechselwirkungen zu anderen Anwendungen in einer vordefinierten Umgebung laufen. Jede Anwendung kann einen eigenen Container mit allen notwendigen Konfigurationsparametern erhalten und auf diese Weise wiederhol- und testbar nach vordefiniertem Verhalten ausgeliefert und bereitgestellt werden. Durch die klare Trennung zwischen unterschiedlichen (Teil-)Anwendungen können auch die jeweiligen Ressourcen besser überwacht und gemanaged werden. Größere Anwendungen werden manchmal auf mehrere Container verteilt, sodass einzelne Pakete wie z. B. Datenpersistenz oder bestimmte Schnittstellen einzeln skaliert und im Falle eines Ausfalls behandelt werden können [Mou16; Tur14].

Im Vergleich zu virtuellen Maschinen sind Container schlanker, leichter zu replizieren und erfordern einen geringeren Konfigurationsaufwand [Wur17]. Durch die Unabhängigkeit von dem darunterliegenden Betriebssystem sind Containeranwendungen bedeutend einfacher zu warten. Im Vergleich zum Server, bei dem eine große Menge an eingebauten Standardfunktionen bis hin zur grafischen Oberfläche zur Verfügung gestellt wird, verwendet der Container nur die für die Anwendung relevanten Features und lässt sich im Fehlerfall leichter wiederherstellen. Aus diesem Grund wird in der vorliegenden Arbeit ein Ansatz mit virtuellen Docker-Containern verfolgt [Mou16].



Source: RightScale 2019 State of the Cloud Report from Flexera

Abbildung 4.2:
Container Tools [Rig]

Docker

Nach einer Umfrage von *RightScale* nutzen 66% der Unternehmen Docker. Somit ist Docker im Moment die verbreitetste Container-Software. Die Idee von Docker ist, die Linux-Containertechnologie benutzerfreundlicher zu gestalten. Deswegen wurde eine Komplettlösung vom Erstellen bis zum Verteilen von Containern geschaffen. Im März 2013 wurde die Software als *OpenSource* zur Verfügung gestellt, wodurch die Community schnell wuchs und die Unterstützung auch durch andere Firmen wie Amazon und Red Hat stieg [Tur14].

Docker hat mehrere Komponenten eingeführt, um alle Schritte möglichst benutzerfreundlich abzudecken. Das *Docker-Image* ist unveränderbar – jede Version einer Software hat sein eigenes Image. Images können auf dem *Docker-Hub* hochgeladen werden. Das Docker-Hub ist eine Plattform, auf der alle Images registriert werden können, entweder öffentlich oder privat [Sok17]. Alle Images können von dort heruntergeladen werden. Images lassen sich aus einem *Docker-File* bilden, dieses besteht aus einem Skript, das eine Folge von Befehlen abarbeitet, um das Images zu erzeugen.

```
FROM tomcat
```

```
RUN rm -rf /usr/local/tomcat/webapps/ROOT/  
ADD target/memo-0.2.war /usr/local/tomcat/webapps/ROOT.war
```

Programmcode 4.1: Docker-File des Meilenshops

Das Docker-File des Meilenshops besteht aus dem Aufruf des Tomcats, dem Löschen des Root-Ordners und dem Hinzufügen des Shops als neuen Root.

Wenn mehrere Container miteinander verbunden werden sollen, kann *Docker-Compose* benutzt werden [Sok17]. Compose verwendet eine YAML-Konfigurationsdatei, die aus zwei Abschnitten besteht. Als Erstes werden die Container-Namen genannt, womit die Container miteinander verbunden werden. Im zweiten Abschnitt können den Containern ihre Eigenschaften – wie beispielsweise freie Ports – zugewiesen werden [Mou16; Sok17]. Im bestehenden Fall könnte Docker Compose verwendet werden, um zwei Container – einer für die Anwendung und einer für die Datenbank – miteinander zu verbinden. Aus Gründen der Persistenz wurde jedoch ein nativer Datenbank-Container von AWS verwendet und mit dem Meilenshop verbunden. Um den Docker-Container auf der Cloud zu nutzen, wurde der Service *Elastic Compute Cloud* genutzt, dieser wird im Abschnitt 4.2 erläutert.

4.1.2 Development Pipeline

Im Rahmen der Umstellung auf die neue Cloud Umgebung bringt die Einrichtung einer *Pipeline* für *Continuous Integration* viele Vorteile. Hierzu wird ein gesonderter *Buildserver* dazu verwendet, den Lebenszyklus der einzelnen Softwareinkremente besser zu überwachen und zu kontrollieren. Dabei können nahezu alle Phasen des Software-Lebenszyklus automatisiert ausgeführt und die entsprechenden Stakeholder benachrichtigt werden [Fil16; Nak18]. Das weit verbreitetste Werkzeug in diesem Bereich stellt *Jenkins* dar, weshalb dieses Projekt darauf basiert [Gra15].

Auf einem Buildserver kann der Entwickler mehrere sogenannte *Jobs* mit unterschiedlichen aufeinander folgenden Phasen einrichten, sodass die Software beispielsweise zuerst getestet, dann die Qualität überprüft und anschließend das eigentliche Produkt erstellt werden kann. Hierbei kann eine solche Pipeline, ähnlich einer Fabrikhalle, mit unterschiedlichen Arbeitsstationen visualisiert werden. Während jede Arbeitsstation eine bestimmte Aufgabe erfüllt, verlässt das Endprodukt nach festgelegten Prozessen und mit kontrollierter Qualität die letzte Station, um ausgeliefert werden zu können. Sogar dieser letzte Schritt lässt sich mithilfe einer Development Pipeline automatisieren [Nak18].



Abbildung 4.3:
Beispiel einer Development Pipeline [Clo]

Die Verwendung einer solchen Pipeline geht mit einer Reihe von Vorteilen einher. Die Automatisierung kann – einmal korrekt eingerichtet – repetitive Schritte bei der Auslieferung der Software ersetzen und auf diese Weise Fehler vermeiden. Außerdem erhält der Entwickler durch die Einrichtung ein mächtiges Analysewerkzeug. So kann die Qualität der Software für jedes spezifische Inkrement direkt überwacht und ein schnelles Feedback an den Entwickler gegeben werden. Dadurch erhalten wichtige Metriken und Messgrößen wie

Testergebnisse und -abdeckung oder Checkstyles mehr Sichtbarkeit. Damit erlangt die Software langfristig eine höhere Qualität [Nak18].

Versionskontrolle

Das Ausgangsmaterial für die meisten Jobs einer Development Pipeline stellt der Programmcode in der Versionskontrolle dar. Oft wird ein Job automatisiert gestartet, wenn in der Versionsverwaltung – hier Git – Änderungen eingchecked werden. Meistens werden anschließend auf dem Buildserver einige Qualitätsmetriken ermittelt, um direktes Feedback zu dem neuen Inkrement geben zu können. Mithilfe dieser automatischen Rückmeldung lässt sich anschließend automatisiert entscheiden, ob die neu hinzugekommenen Änderungen schon qualitativ hochwertig genug sind, um per *Pull Request* auf den Hauptbranch gebracht zu werden. Hier lässt sich das obligatorische Review nicht ersetzen, dennoch gibt ein fehlgeschlagener Build meist einen Hinweis darauf, dass gewisse Probleme noch offen sind.

Testen

So wie das Testen von Software einen wichtigen Aspekt der Entwicklung betrifft, hilft die Automatisierung von Tests um Qualitätsprobleme in einer kontrollierbaren Laufzeitumgebung zu erkennen. Da die Qualität von Softwareprodukten oft nur sehr schwer ermittelt werden kann, lassen – erfolgte oder gescheiterte – Tests oft eine Aussage über die Lauffähigkeit der Software zu. Hierbei ist herauszuheben, dass durch die Verwendung eines zentralen Buildservers die Qualität eines Branches fast ohne Verzögerung sofort für alle Projektbeteiligten ersichtlich ist.

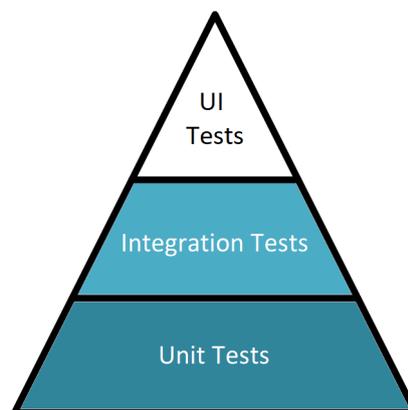


Abbildung 4.4:
Testpyramide von Mike Cohn angelehnt an [Wol16; Wit16]

Um das richtige Maß an Tests zu finden, kann die *Testpyramide* von Mike Cohn genutzt werden. Diese besteht aus drei Ebenen und zeigt das Verhältnis von Unit, Integration und UI Tests. Die Unit Tests stellen die Basis der Pyramide dar. Diese testen nur die Funktion einzelner Klassen und Komponenten innerhalb des Programms und sollten möglichst häufig genutzt werden, da sie einfach zu implementieren sind und keine Abhängigkeiten außerhalb ihrer Module haben. Die Integration Tests überprüfen das Zusammenspiel der Anwendung mit anderen Schnittstellen und sollten weniger genutzt werden, weil sie externe Last erzeugen. Daher sind diese die zweite Ebene. Die Tests haben längere Ausführzeiten und sind schwer zu pflegen. Deshalb ist eine zielgerichtete Nutzung für kritische Schnittstellen sinnvoll. Die dritte Ebene sind die UI Tests, diese sollten am wenigsten genutzt werden, da sie sehr viel Infrastruktur brauchen und langsam laufen [Wol16].

Build

Wenn die Software alle Qualitätstests erfolgreich durchlaufen hat, kann sie mithilfe des Buildservers direkt kompiliert und zu einem fertigen Produkt paketiert werden. Ähnlich wie in produzierenden Gewerben kommt am Ende dieses Schrittes eine fertige Datei heraus, die alle notwendigen Programmteile enthält, um in einer Produktivumgebung verwendet zu werden. Da die aktuelle Konfiguration auf Docker-Containern basiert, liegt hier ein zweistufiges Verfahren vor, bei dem aus der Software zunächst ein ausführbares Paket gebildet und anschließend ein umschließender Docker-Container verwendet wird. Diese notwendigen Schritte lassen sich mithilfe der Pipeline automatisieren und in einer standardisierten Umgebung durchführen. Damit werden unerwünschte Seiteneffekte, die durch äußere Umstände bedingt werden, bedeutend unwahrscheinlicher. Gleichzeitig lässt sich hier das Ergebnis oder eine mögliche Fehlerursache durch jeden Beteiligten direkt im Webbrowser einsehen.

Docker Registry Push

Damit der bestehende Stand auch für eine spätere Ausführung archiviert werden kann, wird der gebaute Docker-Container nach Docker-Hub hochgeladen. So lässt sich ein früherer Stand wiederherstellen und die Versionierung wird stark vereinfacht. In Docker kann – je nach Projekt – ein `private` oder ein öffentliches Repository gepflegt werden. Anschließend kann jeder, der dazu berechtigt ist, sich den Container herunterladen und mit Docker auf dem eigenen lokalen Computer ausführen und testen. Dadurch wird die Verteilung von Versionen entscheidend vereinfacht, da der Anwender keine Entwicklungsumgebung oder Ähnliches konfigurieren muss. Trotzdem ist es möglich, bei Bedarf eine bestimmte Softwareversion zu präsentieren.

Deployment

Wenn alle notwendigen Schritte getan sind, lässt sich die Anwendung in die Cloud hochladen und dort ausführen. Dank des bestehenden Docker-Containers muss sogar keine dedizierte Verbindung zwischen Buildserver und Cloud bestehen. Amazon AWS kann den Container aus der Registry laden und somit sicherstellen, dass immer die aktuelle Version läuft. Da sämtliche programmseitigen Konfigurationen direkt in Docker oder in der Anwendung erfolgen, lässt sich das Deployment ohne größeren Konfigurationsaufwand automatisieren und die Hostumgebung wechseln. Lediglich verbundene Services mit angeschlossenen Api-Schlüsseln müssen hinterlegt werden. Mithilfe dieser Schritte lässt sich eine Verteilung der aktuellen Softwareversion resistent gegen Fehler und vollkommen automatisiert verwirklichen.

4.2 Betrieb

Die Docker-Container können auf AWS mithilfe des Web-Service *Amazon Elastic Compute Cloud* (EC2) laufen. Es gibt zwei Modelle, die genutzt werden können, der „EC2-Start-Typ“ und der „Fargate-Start-Typ“. Der EC2-Start-Typ ist kostenlos, dafür muss das Cluster, auf dem die Container laufen, selbst konfiguriert werden [Amac]. Wenn alle administrativen Aufgaben von AWS übernommen werden sollen, ist Fargate der richtige Service. Für den Meilenshop wird der Start-Typ von EC2 genutzt, da die Kosten gering gehalten werden sollen. Um den Service zu nutzen, muss als Erstes die Umgebung definiert (Kapazitäten, Ports) und das Image hochgeladen werden. Danach muss der Service und dann das Cluster konfiguriert werden [Aman]. Die Datenbank kann mit dem Service *Amazon RDS for MySQL* erstellt werden und wird auf diesem Weg in den Meilenshop eingebunden [Amad].

Mit welchen Services der Meilenshop auf der Cloud in der Zukunft optimiert werden soll, folgt in den nächsten Abschnitten.

4.2.1 Monitoring

Zum Monitoring stellt Amazon den Service *CloudWatch* zur Verfügung, mit dem die eigenen Cloud-Ressourcen und Anwendungen überwacht werden können [Amaa]. Zur Überwachung werden Metriken genutzt. Diese können individuell generiert werden. Es können Metriken für Anwendungen, aber auch vordefinierte Metriken, wie beispielsweise für Datenbanktabellen, erzeugt werden [Aman]. So kann die Auslastung der Ressourcen nachverfolgt und gegebenenfalls darauf reagiert werden. Dafür bietet AWS Alarmer an, mit denen die Ressourcen je nach Bedarf automatisiert freigegeben werden können, um so einen störungsfreien Betrieb mit möglichst wenig Ressourcen zu gewährleisten [Aman].

4.2.2 Skalierbarkeit

Um die Infrastrukturkosten möglichst gering zu halten und die genutzten Anwendungen zu optimieren, kann die Cloud skaliert werden. Um die Ressourcen dem Bedarf genau anzupassen, bietet Amazon den Service *AutoScaling* [Bar] an. Als Erstes muss ein Skalierungsplan erstellt werden, in welchem festgelegt wird, wie jede Ressource für die Anwendung skaliert werden soll. Der Plan hat einen Sollwert, um den die Ressource skaliert wird. Es kann festgelegt werden, ob die Verfügbarkeit oder die Kosten optimiert werden sollen. Es ist aber ebenfalls möglich, einen Kompromiss zu nutzen, welcher über den festgelegten Sollwert reguliert wird [Bar]. Die Cloud regelt mithilfe des Plans die Skalierung automatisiert.



Abbildung 4.5:
AWS Auto Scaling Ablauf [Bar]

4.2.3 Sicherheit

Um den Meilenshop abzusichern, werden zwei zusätzliche Services von AWS genutzt, die im kommenden Abschnitt näher erklärt werden.

Zertifikate

Für den Shop und die Website der Meilenwölfe werden zwei SSL-Zertifikate benötigt. Um die Zertifikate auch in der Cloud zu nutzen, wird ein passender Service verwendet. Mit dem *AWS Certificate Manager* (ACM) können Zertifikate erstellt und genutzt werden. Gerade die Erstellung von Zertifikaten über ACM ist jedoch ein teurer und für die Meilenwölfe nicht benötigter Service. Die verwendeten Zertifikate können kostenlos bei externen Diensten wie Let's Encrypt erzeugt werden [Let19]. ACM kann jedoch auch extern ausgestellte SSL-Zertifikate verwalten, was keine weiteren Kosten verursacht. Wenn das externe Zertifikat erneuert werden muss, benachrichtigt der Service die Entwickler, die die Erneuerung aber trotzdem per Hand erledigen müssen [Amaf]

Backups

Um den reibungslosen Ablauf auch nach einem Absturz oder Problem zu gewährleisten, wird eine Sicherheitskopie vom Shop und von der Datenbank benötigt. Der Datenbank Service hat einen integrierten Backup-Service, der für 10 GB etwa 0,95\$ im Monat kostet. Mithilfe des *Amazon Data Lifecycle Managers* kann ein *Snapshot* – ein virtuelles Abbild - vom Shop gespeichert werden [Ama18d]. Dieser Snapshot wird als Basis genutzt, bei jeder weiteren Sicherung werden nur die Änderungen hinzugefügt, um Speicherplatz zu sparen. Die Kosten dieses Service betragen 0,054\$ pro GB und Monat der Datenspeicherung [Ama18d]. Eine Alternative könnte es sein, das aktuelle Image des Shops zu laden, da sowieso die aktuelle Version im Docker-Image gespeichert wurde.

Kapitel 5

Evaluation

In diesem Kapitel werden die Infrastrukturen der Cloud und des Servers bewertet. Bei dem Server handelt es sich dabei um den Linux Server von Strato auf dem der Meilenshop zu Projektbeginn ausgeführt wurde. Als neue Infrastruktur wird die AWS Cloud von Amazon genutzt, auf der der Meilenshop im Docker-Container läuft. Zur Bewertung wird das Teufelsquadrat herangezogen. Als Erstes werden die Kosten der Infrastruktur bewertet, danach wird die Zeit betrachtet. Die Qualität wird mithilfe des SMI untersucht und als Letztes wird der Leistungsumfang gegeneinander gestellt.

Bei jeder Kategorie wird herausgearbeitet, ob die Transformation in diesem Bereich einen Vorteil für den Meilenshop darstellt. Die Infrastruktur, die in mehr Kategorien überzeugt, sollte für den Meilenshop in Zukunft im Betrieb genutzt werden. Die Evaluation basiert auf den Aussagen der Betreiber, gemessenen Werten und eigenen Beobachtungen und Bewertungen.

5.1 Kosten

Die Kosten werden mithilfe des Gesamtkostenvergleichs von Lamberth und Weisbecker aufgeschlüsselt und bewertet. Alle Kosten werden pro Monat abgerechnet.

Als Erstes werden die direkten und indirekten Kosten des Servers, danach die der Cloudlösung evaluiert. Der Server verursacht keine Kosten aus der On-Demand-Nutzung, da dieser feste Ressourcen zugewiesen bekommt, die nicht kurzfristig verändert werden können. Die Hardwarekosten sind im Hostingbeitrag enthalten, da der Provider sich um die Pflege der Hardware und deren Austausch kümmert. Die genutzte Software wie MySQL oder Apache Tomcat sind Open-Source-Tools, der Meilenshop ist selbst entwickelt. So entstehen softwareseitig ebenfalls keine IT-Investitionskosten. In der Kategorie Implementierung & Design entstehen keine Kosten, da das Entwicklungsteam ehrenamtlich arbeitet. Um Kosten zu senken haben die Entwickler keine Schulungen besucht oder an Weiterbildungsmaßnahmen

KOSTEN	Server	Cloud
DIREKTE KOSTEN		
Externe Kosten aus On-Demand-Nutzung		
Kosten für Rechenleistung		-
Kosten für Bandbreite	-	-
Kosten für Speicher		-
IT-Investitionskosten		
Hardware	-	-
Software	-	- Amazon EC2 Container Service: 4,64\$ - Relational Database Service: 17,38\$ - AWS Backup: 0,95\$ - AWS Certificate Manager: kostenlos 22,97\$ ~ 20,58€
Implementierungs- & Designkosten	-	-
Schulungskosten	-	-
Kommunikationskosten	-	-
Laufende Kosten (Betrieb, Prozesse)		
Energiekosten	-	-
Hosting	14€	-
Jährliche Wartung & Weiterentwicklung	-	Siehe AWS Backups
Administration & Support	-	-
Kosten für Dienstleistungen	-	-
IT-Personalkosten	-	-
INDIREKTE KOSTEN		
Opportunitätskosten durch End-User-Operations	-	-
Ausfallzeit-bezogene Kosten	-	-
GESAMTKOSTEN	14€	20,58€

Abbildung 5.1:
Kostentabelle

teilgenommen. Folglich sind bei Entwicklung und Betrieb des Servers keine IT-Investitionskosten entstanden.

Alle laufenden Kosten (wie Energiekosten) werden durch die Hostinggebühr von 14 Euro im Monat abgedeckt. Die Wartung, Weiterentwicklung und Administration wird vom Entwicklungsteam ehrenamtlich übernommen. Alle direkten Kosten belaufen sich beim Server deshalb auf diese Summe. Bisher ist noch kein Fall bekannt in dem – aufgrund eines Serverausfalls – jemand nicht an einer Tour oder einer Veranstaltung teilnehmen konnte. Daher werden die Opportunitäts- und Ausfallzeit-bezogenen Kosten auf null geschätzt.

Die Gesamtkosten der Serverumsetzung belaufen sich somit auf die 14 Euro pro Monat für das Hosting.

Da die Kosten der AWS Cloud sich zwischen den Regionen unterscheiden, werden zur Bewertung nur die Kosten der Region Europa (Frankfurt) genutzt. Bei der AWS Cloud entstehen keine direkten Kosten aus der On-Demand-Nutzung, da keine Rechenleistung, Speicher oder Bandbreite gekauft wird, sondern Services genutzt werden, bei denen die Ressourcen ausgewählt werden können. Die Hardwarekosten für die Cloud verteilen sich gleichmäßig auf die Services.

Die Softwarekosten müssen aus zwei Perspektiven betrachtet werden. Der Meilenshop und der Open-Source-Technologiestack wurden um die Docker-Komponente erweitert, wodurch keine weiteren Kosten entstehen, weil Docker ebenfalls eine Open-Source-Software ist. In dieser Arbeit werden die AWS Services ebenfalls als Software gesehen, daher entstehen hier die Hauptkosten der Cloud. Der erste Service, der für den Meilenshop genutzt wird, ist der Service *Elastic Compute Cloud* (EC2). In diesem Service läuft der Docker-Container des Meilenshops. Dem Service müssen Ressourcen zugeordnet werden, für die Kapazität des Meilenshops wird *t3.nano* verwendet. Diese Umgebung hat 2 CPUs, 0,5 GB Arbeitsspeicher, 2 GB Storage und kostet in dieser Konfiguration 4,64\$ im Monat [Amab].

Modell	vCPU*	CPU Gutschriften/Stunde	Arbeitsspeicher (GiB)	Speicher	Netzwerkleistung (Gbit/s)
t3.nano	2	6	0,5	Nur EBS	Bis zu 5

Abbildung 5.2:
AWS – EC2 Spezifikationen [Amab]

Für den Meilenshop wird eine Datenbank benötigt. Daher wird der Service *RDS - Relational Database Service* genutzt. Mit dessen Hilfe kann die Datenbank in der Cloud eingerichtet, verwaltet und skaliert werden. Es kann zwischen bekannten Datenbankmanagementsystemen gewählt werden, zur Verfügung stehen *Amazon Aurora*, *PostgreSQL*, *MySQL*, *MariaDB*, *Oracle Database* und *Microsoft SQL Server*. Für den Meilenshop wird die Datenbankinstanz *db.t3.micro* genutzt, welche monatlich 14,64\$ kostet. Der Speicher von 10 GB kostet 2,74\$ extra. Die Backups der Datenbank verursachen zusätzliche Kosten, diese belaufen sich bei 10 GB auf 0,95\$. So entstehen für die Datenbank in der Cloud insgesamt Kosten von 17,38\$. Die angesetzten Ressourcen sind im Vergleich zu den des Servers überdimensioniert. Jedoch sind dies die kleinst auswählbaren Ressourcen für diesen Service. Eine mögliche Alternative wäre eine Datenbank, die in einem Docker-Container auf dem EC2 läuft.

Um die SSL-Zertifikate des Shops zu verwalten, wird der *AWS Certificate Manager* genutzt. Die Nutzung des Services ist kostenfrei, lediglich die Generierung eines Zertifikats erzeugt Kosten. Da die Zertifikate, wie in der bereits bestehenden Lösung, aber durch *Let's Encrypt* zur Verfügung gestellt und hier nur eingebunden werden, entstehen keine Kosten [Let19].

Die Energiekosten sind auf die einzelnen Services verteilt und schon in den Softwarekosten enthalten. Das Hosting entfällt bei der Cloud. Die Backups sind schon in den Softwarekosten enthalten, alle anderen Wartungen werden vom Entwicklungsteam durchgeführt. Der Basis-Support ist kostenlos verfügbar, dazu zählen *FAQs* für alle Services, *Best Practices* und Anleitungen. Nach Absprache mit dem Entwicklungsteam ist dieser Support vollkommen

ausreichend [Amah]. Es gibt jedoch noch drei andere Stufen von Support bei AWS – *Developer*, *Business* und *Enterprise*. Der Developer-Support ist ab 29\$ pro Monat verfügbar und enthält einen E-Mail-Zugang zu Support-Technikern während der Geschäftszeiten. Laut AWS ist dieser Support bei der Durchführung von AWS Experimenten oder Testverfahren empfehlenswert. Der Business Support wird empfohlen, wenn eine Umgebung auf AWS produktiv genutzt wird. Hier sinken die Reaktionszeiten bei bestimmten Fällen drastisch, außerdem kann der Support-Techniker über Telefon, E-Mail und Chat erreicht werden. Außerdem erhält der Kunde hier Hilfe beim Aufbau der Architektur und viele weitere Features. Diesen Support Service gibt es ab 100\$ pro Monat. Der Enterprise Support ist das größte Supportpaket und kostet mindestens 15.000\$ pro Monat. Dieses Paket wird empfohlen, wenn geschäfts- und/oder aufgabenkritische Arbeitslasten in AWS getätigt werden. Teil des Supports sind Online-Übungen fürs Selbststudium, proaktive Überwachung der Umgebung vom *Technical Account Manager* und viele weitere Features. Der Preis ist mindestens der oben angegebene Wert. Alle Preise steigen hier prozentual zu den Gesamtkosten [Amah]. Für den Meilenshop reicht jedoch der kostenlose Basis-Support aus.

Bei der Cloud werden ebenfalls alle Implementierungen und Entwicklungen von den ehrenamtlichen Mitarbeitern übernommen und so entstehen auch hier keine Personalkosten. Auch bei den Kommunikations- und Schulungskosten entstehen keine Defizite.

Indirekte Kosten konnten bisher noch nicht ermittelt werden, bei einer prognostizierten Ausfallwahrscheinlichkeit von weniger als 1% ist es sehr unwahrscheinlich, dass mehr Kosten als beim Server entstehen.

Die Kosten der AWS Cloud für den Meilenshop belaufen sich auf 22,97\$ (\approx 20,58 Euro) pro Monat.

Im Vergleich sind die Kosten des Strato Servers deutlich geringer (14 Euro pro Monat) als die Kosten der AWS Cloud (20,58 Euro). Hinzukommt, dass der Server deutlich einfacher kalkulierbare Kosten als die AWS Cloud hat, da die Kosten kontinuierlich gleichbleibend sind. Bei der Cloud können die Ressourcen automatisch angepasst werden, dies kann zu geringeren, aber auch zu deutlich höheren Kosten führen. Da die Cloud bei Amazon im ersten Jahr kostenlos genutzt werden kann, ist der Service im ersten Jahr deutlich attraktiver als in den folgenden Jahren. Jedoch darf der Service im kostenlosen Jahr nur 750 Stunden pro Monat genutzt werden, bevor auch hier Kosten entstehen. Wenn erhöhte Kosten durch die Skalierung entstehen, können extrem hohe Summen zustande kommen. Es gibt jedoch die Möglichkeit automatische Limitierungen einzurichten. Gerade Lasttests können durch automatische Skalierung zu enormen Kosten führen. Die bevorzugte Infrastruktur für den Meilenshop unter Berücksichtigung der Kosten ist der Server von Strato. Hierzu muss hinzugefügt werden, dass die von der Cloud zur Verfügung gestellten Ressourcen die des Servers deutlich übersteigen und daher ein erhöhter Preis zu rechtfertigen ist.

5.2 Zeit

Zur Bewertung wird einerseits die Entwicklungszeit für die Cloudtransition, andererseits die aufgewendete Zeit für wiederkehrende Aufgaben herangezogen. Die Entwicklungszeit des Shops wird hierzu nicht betrachtet, da dieser auf beiden Infrastrukturen mit der gleichen Version läuft und die Aufwendungen somit für beide Alternativen identisch wären. Die bewertete Zeit wird in drei Phasen unterteilt: Einmalige Investitionen zur Schulung, einmaliger Entwicklungsaufwand und wiederkehrende Aufgaben.

Während das reine Anlernen der Serveradministration mit geschätzten 5 *Story Points* (SP) bedeutend weniger Aufwand umfasst als die Schulungen zu AWS (3 SP), Docker (3 SP) und angeschlossenen Services (3 SP), liegen die Investitionszeiten beim Server mit 5 Story Points deutlich geringer als bei der Cloud mit 9 Story Points.

Ein ähnliches Bild ergibt sich bei der Entwicklungszeit. Während auf dem Server nur kleinste Änderungen mit 0,5 geschätzten Story Points vorgenommen werden müssen, gibt es auf der Cloudseite bedeutende Aufwände. Diese werden mit 2 Story Points für die Entwicklung von Docker und 2 weiteren Punkten für Anpassungen am Code und der Konfiguration bewertet, womit die Entwicklung der Cloudlösung eine zusätzliche Komplexität von 4 Story Points erfordert. Zusammengezählt betragen die einmaligen Kosten, also 5,5 Story Points auf dem Server und 13 Story Points auf der Cloud.

Zeit	Server	Cloud
+ Schulung:	5	3+3+3
+ Entwicklungsaufwand:	0,5	2+2
Einmalige Investitionen:	5,5	13
Wiederkehrender Aufwand	2	0,5

Tabelle 5.1: Evaluation der Zeit mit Story Points

Anders sieht es bei dem wiederkehrenden Aufwand für neue Auslieferungen aus. Während ein Deployment auf der Cloud nur geringste Anpassungen mit 0,5 Story Points erfordert, muss auf dem Server das Betriebssystem konfiguriert, die Programme und Laufzeitumgebungen installiert (1 SP) und das eigentliche Deployment durchgeführt werden (1 SP). Während ein Deployment in der Cloud somit eine Komplexität von 0,5 Story Points hat, liegt die verglichene Komplexität auf dem Server bei 2 Story Points.

Auch wenn Story Points nicht direkt vergleichbar und addierbar sind, lässt die Bewertung einen Rückschluss darauf zu, dass die aufgewendete Entwicklungszeit sich bereits nach wenigen Auslieferungen amortisiert hat. In dem Rechenbeispiel passiert dies nach der fünften Auslieferung. Bei durchschnittlichen Deploymentzyklen von rund einem Monat, ist diese Investition also bereits im ersten Jahr der Umsetzung, wieder im Gewinnsektor.

5.3 Qualität

Zur Bewertung der Qualität wird der Service Measurement Index (SMI) herangezogen. Im Kapitel 3.4 wurden die Merkmale des Index mithilfe der Anforderungen des Meilenshops bewertet. Im folgenden Abschnitt werden nur die als „essenziell“ bewerteten Merkmale zur Bewertung der Qualität genutzt.

Verantwortlichkeit

In der Kategorie *Verantwortlichkeit* werden das *Inhaberverhältnis*, die *wirtschaftliche Sicherheit*, die *Zertifizierung des Providers* und die *Service Level Agreements* bewertet.

Die erste Unterkategorie ist das *Inhaberverhältnis*. Um mehr über die Rechte an den Daten, Lizenzen und dem geistigen Eigentum zu erfahren, wurden die *Allgemeinen Geschäftsbedingungen* (AGB) herangezogen. In den AWS Kundenvereinbarungen steht:

„8. Eigentumsrechte.

8.1 Ihre Inhalte. Wenn nicht in dieser Ziffer 8 vorgesehen, erhalten wir nach dieser Vereinbarung keine Rechte von Ihnen (oder Ihren Lizenzgebern) an Ihren Inhalten. Sie stimmen zu, dass wir Ihre Inhalte nutzen dürfen, um Ihnen und allen Endnutzern die Serviceangebote zur Verfügung zu stellen.“ [Ama18c, S. 7]

Daher hat Amazon keine Rechte an den Daten der Meilenwölfe, wenn diese die AWS Cloud nutzen. In den AGBs von Strato wird nicht erwähnt, ob die Daten, Lizenzen, etc. im Eigentum der Meilenwölfe bleiben, womit ebenfalls davon ausgegangen werden kann, dass die Daten im Besitz der Meilenwölfe bleiben. Da AWS eine klare Definition vorweisen kann, ist die AWS Cloud im Bereich Inhaberverhältnis die bessere Wahl.

Verantwortlichkeit	Server	Cloud
Inhaberverhältnis		+
Wirtschaftliche Sicherheit		+
Zertifizierung des Providers		+
Service Level Agreements	+	+

Tabelle 5.2: Evaluation der Verantwortlichkeit

In der nächsten Kategorie wird die *wirtschaftliche Sicherheit* beleuchtet. AWS wurde 2006 als Tochterunternehmen von *Amazon.com* gegründet und wächst seitdem jedes Jahr [Ama18a]. Strato wurde 1997 gegründet und 2009 an die Telekom verkauft. Diese verkaufte Strato 2017 weiter an *United Internet* [Strd]. Bei jedem Verkauf ändern sich die AGBs, Verträge, Produkte, etc. wodurch die wirtschaftliche Sicherheit auf lange Perspektive geschmälert

wird. Strato hat einen Marktanteil von 13% in Deutschland und somit den drittgrößten Marktanteil in Deutschland im Bereich Hosting [Hos19]. Der Marktanteil von AWS lag 2018 bei 51% [Mus18]. Damit ist AWS der Marktführer im IaaS-/PaaS-Markt. Wegen der Unternehmensgeschichte wird AWS als wirtschaftlich sicherer betrachtet.

Eine andere Unterkategorie der Verantwortlichkeit ist die **Zertifizierung des Providers**. In den Anforderungen wurden zwei Zertifizierungen festgelegt, die die Infrastruktur Anbieter abgeschlossen haben sollten. Die erste Zertifizierung ist die ISO/IEC 27001:2013, diese ist Teil der zweiten Zertifizierung: Dem Testat *Cloud Computing Compliance Controls Catalogue* (C5). Strato kann die ISO/IEC 27001:2013 nachweisen, die vom TÜV SÜD ausgestellt wurde [TÜV16]. Da Strato einen Server anbietet, kann dieser das Cloudzertifikat C5 nicht erfüllen, allerdings kann Strato kein weiteres Zertifikat vorweisen. Eine sinnvolle Erweiterung wäre zum Beispiel ISO 22301 für ein ganzheitliches Management der betrieblichen Kontinuität im Unternehmen, um mögliche Schäden im Falle einer Störung zu minimieren. Strato wirbt mit Testsiegeremblem, die mithilfe von Kundenumfragen erstellt wurden. Dies sind keine Zertifizierungen, können aber auf gewisse Weise ein „sicheres Gefühl“ vermitteln.

AWS kann die ISO 27001-Zertifizierung nachweisen, der Anbieter wurde von *EY CertifyPoint* zertifiziert [EY 19]. Das Testat C5, durch das Bundesamt für Sicherheit in der Informationstechnik geprüft, hat AWS erhalten. Das BSI prüft in regelmäßigen Abständen, ob die Anforderungen auch weiterhin erfüllt werden. AWS kann auch noch weitere Zertifizierungen in Europa nachweisen. Beispiele hierfür sind 27017:2015, 27018:2014 und 9001:2015 [Ama19] und andere [Amaj].

Da die minimale Anforderung mit der ISO/IEC 27001 von beiden Infrastrukturen erfüllt wurde, entscheiden die weiteren Zertifizierungen des jeweiligen Anbieters über diese Kategorie. Strato kann keine weiteren Zertifizierungen nachweisen, während AWS mit C5 und noch mehreren anderen Zertifizierungen deutlich die Anforderungen dieser Kategorie erfüllt.

Das letzte notwendige Kriterium in diesem Komplex beinhaltet vorhandene **Service Level Agreements**. Beide Anbieter haben SLAs, die bei Strato pro Server gelten [Stra; Str15] und bei AWS pro Service [Ama15], daher sind beide geeignet für den Meilenshop. Zwar sind die Kosten bei beiden Lösungen unterschiedlich, dies wird jedoch durch die „Kosten“-Dimension des Teufelsquadrats abgedeckt.

In der Kategorie *Verantwortlichkeit* überzeugt AWS deutlich mit allen essenziellen Unterkategorien, Strato kann nur bei den Service Level Agreements überzeugen, weshalb AWS diese Kategorie für sich entscheidet.

Agilität

Ob die Cloud oder der Server besser an die Bedürfnisse des Meilenshops anpassbar sind, entscheidet sich in der Kategorie *Agilität*. Die zu überprüfenden Unterkategorien sind Elastizität, Erweiterbarkeit und Skalierbarkeit.

Die **Elastizität** wird bei AWS pro Service verändert. Wenn durch erhöhten Zugriff mehr Ressourcen benötigt werden, kann der Nutzer die Ressourcen automatisiert erhöhen lassen [Ama18b; Bar]. Die Automatisierung der Ressourcenanpassung muss vorher eingestellt werden, da so weitere Kosten entstehen können. Beim Server gibt es festgelegte Ressourcen, welche bei erhöhter Nutzung nicht reguliert werden. Wenn es beispielsweise zu viele Anfragen gibt, werden diese im Zweifelsfall nicht weitergeleitet. Da AWS Elastizitätsanpassungen zulässt, punktet der Cloud-Anbieter in diese Unterkategorie.

Agilität	Server	Cloud
Elastizität		+
Erweiterbarkeit		+
Skalierbarkeit		+

Tabelle 5.3: Evaluation der Agilität

Die nächste Unterkategorie **Erweiterbarkeit** soll die Möglichkeit aufzeigen, die bestehende Anwendung durch weitere Features zu erweitern. In der Bestandlösung ließen sich neue Features immer nur schwer in die Produktivumgebung integrieren und testen. Mithilfe von Docker können neue Images nun schnell und einfach erzeugt werden. Außerdem ist ein detailliertes Monitoring durch einen Service möglich. Durch die Verbindung der Cloud-Eigenschaften und der Funktionalität von Docker ist die Cloud hier deutlich im Vorteil und gewinnt die Kategorie *Erweiterbarkeit*.

Die **Skalierbarkeit** ist ein entscheidendes Merkmal von Cloud-Computing. Services können je nach Bedarf skaliert werden. Diese Möglichkeiten kann der Server nicht bieten, da er ausschließlich feste Ressourcenzuteilungen beinhaltet.

Bei Elastizität, Erweiterbarkeit und Skalierbarkeit überzeugt die Cloud und kann daher die *Agilität* für sich entscheiden.

Verlässlichkeit

Welche Infrastruktur die verlässlichere ist, wird im nächsten Abschnitt diskutiert. Dabei werden die Unterkategorien Verfügbarkeit, Wartbarkeit, Wiederherstellbarkeit und Betriebstauglichkeit genutzt.

Der erste Punkt in diesem Bereich ist die **Verfügbarkeit**. Das Ziel hier lag bei mindestens 99%. Beide Anbieter übertreffen diese Erwartungen [Rei18b; Die15], womit beide in dieser Unterkategorie punkten können.

Als Nächstes wird die **Wartbarkeit** diskutiert. Beim Server müssen Backups, Updates für das Betriebssystem und die Laufzeitumgebung per Hand erneuert werden. Dies bedeutet viel Zeit, die das Entwicklungsteam in die Wartung investieren muss. Durch die Abstraktionsebenen einer Cloud kann der Administrations- und Wartungsaufwand drastisch gesenkt werden. Beispielsweise muss das Betriebssystem nicht mehr aktualisiert werden [Rao16]. Außerdem kann der administrative Aufwand durch Docker weiter verringert werden, da die aktuellen Images einfach herunterladbar sind. Die Cloud erspart dem Entwicklungsteam viel Wartungsaufwand und kann so diese Unterkategorie für sich entscheiden.

Verlässlichkeit	Server	Cloud
Verfügbarkeit	+	+
Wartbarkeit		+
Wiederherstellbarkeit		+
Betriebstauglichkeit		+

Tabelle 5.4: Evaluation der Verlässlichkeit

In der nächsten Unterkategorie wird die **Wiederherstellbarkeit** der Anwendung auf den Infrastrukturen bewertet. Jeder Linux-Server von Strato ist mit *BackupControl* ausgestattet, bei der zehn unterschiedliche Sicherungen gleichzeitig gespeichert werden können. Einer dieser Slots kann als Langzeitspeicher genutzt werden. Das heißt, diese Sicherung wird nicht weiter überschrieben. Diese Leistungen sind bei Strato kostenfrei [Strc]. AWS stellt einen Backup-Service zur Verfügung, welcher die Daten aller genutzten Services und derer Daten sichern kann. Die Konfiguration des Services erfolgt selbstständig und bietet viele Möglichkeiten [Amae]. Vorteile dieses Services sind die zentrale Sicherung und Verwaltung sowie die Automatisierung des Vorgangs. Der Backup-Service kostet jedoch pro GB Datensicherung und pro wiederherstellbaren GB. AWS gibt bei diesem Service eine Beispielrechnung an, bei der ein deutlich größerer Service monatlich 30,20\$ kostet. Dies lässt sich nicht direkt mit dem Meilenshop vergleichen, gibt aber eine grobe Größenordnung an. Beide Infrastrukturen bieten eine Datensicherung an. Während die Lösung von Strato kostenfrei zur Verfügung steht, bietet AWS bedeutend mehr Funktionsumfang und Flexibilität zu einem respektablen Preis, was der Gesamtqualität zuträglich ist. Deshalb punktet AWS in diesem Bereich.

Die Messung der **Betriebstauglichkeit** des Strato Servers ist sehr unständig, da das Monitoring wenig Informationen zeigt. Die Auslieferung des Shops muss per Hand durchgeführt werden und um eine losgelöste Testumgebung zu erhalten, müsste ein weiterer Server angemietet werden. Diese Problematiken werden durch die Anwendung von Docker reduziert, denn mithilfe eines neuen Docker-Containers kann eine andere Testumgebung

geschaffen werden. Außerdem kann der Shop mit der neuen Development Pipeline leicht ausgeliefert und getestet werden. Durch die unterschiedlichen Metriken beim Monitoring-Service können deutlich mehr Aspekte verfolgt werden. Da die Betriebstauglichkeit ein wichtiger Grund für die Umstellung auf Docker war und die erhofften Verbesserungen umgesetzt worden sind, ist die Kombination von Cloud und Docker deutlich dem Server überlegen.

AWS kann bei Wiederherstellbarkeit, Wartbarkeit und Betriebstauglichkeit überzeugen und entscheidet so die *Verlässlichkeit* für sich.

Sicherheit & Privatsphäre

In der Kategorie *Sicherheit & Privatsphäre* werden die Unterkategorien Lokalität, physische Sicherheit und Gefährdungsmanagement bewertet.

Zunächst wird die **Lokalität** bewertet. Aufgrund der *DSGVO* ist es sinnvoll, dass die physischen Geräte in der Europäischen Union stehen. Dies ist auch eine Anforderung aus Kapitel 3.2. Durch den in AWS auswählbaren Standort ist es auch möglich die Zone Europa oder spezieller Frankfurt zu wählen. Da alle Server von Strato in Deutschland stehen, erfüllt auch Strato die Anforderungen und beide Infrastrukturen liegen bei dieser Thematik gleichauf.

Um die **physische Sicherheit** hier greifbarer zu machen, wird die Sicherheit als gewährleistet gesehen, wenn die ISO/IEC 27001 erfüllt wird. Diese Norm gibt vor, dass standardisierte Richtlinien und Prozesse genutzt werden müssen, um die Rechenzentren zu schützen [Kos]. Da diese ISO-Norm bei beiden Anbietern vorliegt, können sie in dieser Unterkategorie punkten.

Sicherheit & Privatsphäre	Server	Cloud
Lokalität	+	+
Physische Sicherheit	+	+
Gefährdungsmanagement		+

Tabelle 5.5: Evaluation der Sicherheit & Privatsphäre

Die Basis des **Gefährdungsmanagement** ist ebenfalls durch ISO/IEC 27001 gewährleistet. Da der Server jedoch vom Entwicklungsteam lauffähig und sicher gehalten werden muss, sind die Risiken der Cloud im Bereich Gefährdungsmanagement geringer und daher punktet AWS.

Im Bereich der **Sicherheit & Privatsphäre** erhält die Cloud knapp mehr Punkte, da diese allen drei Unterkategorien für sich entscheiden kann. Strato ist beim Kriterium Sicherheit jedoch fast genauso überzeugend.

Benutzerfreundlichkeit

Die letzte Kategorie der Qualität ist die *Benutzerfreundlichkeit*, welche aus genau einer essentiellen Unterkategorie besteht, der Personalanforderung.

Personalanforderung beschreibt, wie viel Personalaufwand für die Entwicklung mit dem Technologiestack notwendig ist. Da das Team eine feste Größe von drei Personen hat, gehen solche Aufwände direkt von dem möglichen Entwicklungsaufwand ab. Auf dem Server müssen Tests von Hand ausgeführt werden und Monitoring ist kaum gegeben. Die Serveradministration erfolgt via SSH.

Die Cloud dagegen verfügt über ein Dashboard mit allen notwendigen Informationen und möglichen Aktionen. Diese sind über die Benutzeroberfläche oder über ein Command Line Interface möglich. Außerdem gibt es viele FAQs und einen eigenen Wikibereich. Zusätzlich erzeugt der Buildserver Transparenz.

Aus diesem Grund liegt die Cloudlösung weit vor der Ausgangslösung.

Zusammenfassung

Die Cloud punktet in den Kategorien Verantwortlichkeit, Agilität, Verlässlichkeit und Sicherheit & Privatsphäre, ebenso wie bei den Personalanforderungen. Da die Cloud in allen fünf Kategorien überzeugt, ist sie die qualitativ bessere Wahl als neue Infrastruktur.

Gesamt	Server	Cloud
Verantwortlichkeit		+
Agilität		+
Verlässlichkeit		+
Sicherheit & Privatsphäre		+
Benutzerfreundlichkeit		+

Tabelle 5.6: Evaluation der Qualität

5.4 Leistungsumfang

Beide zu untersuchende Infrastrukturen haben neben ihrer Hauptfunktionalität andere Leistungen, welche hier untersucht werden. Die Infrastruktur, die quantitativ überzeugt, punktet in dieser Kategorie.

Der Strato Server hat mehrere zusätzliche Leistungen, die überwiegend kostenlos genutzt werden können. Strato bietet eine grafische Oberfläche zur Verwaltung und Konfiguration des Servers – *Plesk Onyx*. Zur Datensicherung und zum Schutz vor Datenverlust gibt es die Software *BackupControl*. Damit der Server überwacht werden kann, wird eine Monitoringsoftware angeboten. Außerdem bietet Strato den *RecoveryManager* an, der als Server-Versicherung angepriesen wird. Damit kann jederzeit der Server neu gestartet werden, selbst wenn es mit Plesk Onyx Probleme gibt. *TrafficControl* bietet einen Überblick über den Datenverkehr auf dem Server. Außerdem bietet Strato *ClusterIP* an, was für eine höhere Ausfallsicherheit sorgen soll. Durch eine zweite IP-Adresse kann zwischen zwei Servern umgeschaltet werden, um so eine höhere Flexibilität und Ausfallsicherheit zu garantieren [Stre]. Diese Software ist nicht kostenlos. Um das SSL-Zertifikat zu verwalten, bietet Strato ebenfalls eine Software an. Die Verwaltung des ersten Zertifikates ist kostenlos, jedes weitere kostet Geld. Da der Meilenshop und die Website der Meilenwölfe jeweils ein eigenes Zertifikat benötigen, ist dieses Tool nur teilweise nützlich. Außerdem enthält das kostenlose Zertifikat keine Möglichkeit Wildcard Domains zu unterstützen, womit eine Absicherung von Subdomains ausgeschlossen ist. Strato bietet also als zusätzliche Leistungen sieben unterschiedliche Tools an. Diese sind streng proprietär und lassen sich weder erweitern, noch wird ausreichende Dokumentation dazu gestellt.

Estimating your costs

Choose your region and fulfillment option to see the pricing details. Then, modify the estimated price by choosing different instance types.

Region

Fulfillment Option

Software Pricing Details

WordPress Certified by Bitnami and Automatic **\$0 /hr** >
running on t2.nano

Infrastructure Pricing Details

Estimated Infrastructure Cost **\$0.007 EC2/hr** >

Free Tier < EC2 charges for Micro instances are free for up to 750 hours a month if you qualify for the [AWS Free Tier](#).

The table shows current software and infrastructure pricing for services hosted in EU (Frankfurt). Additional taxes or fees may apply.

EC2 Instance type	Software/hr	EC2/hr	Total/hr
<input checked="" type="radio"/> t2.nano	\$0	\$0.007	\$0.007
<input type="radio"/> t2.micro	\$0	\$0.013	\$0.013
<input type="radio"/> t2.small	\$0	\$0.027	\$0.027
<input type="radio"/> t2.medium	\$0	\$0.054	\$0.054
<input type="radio"/> t2.large	\$0	\$0.107	\$0.107
<input type="radio"/> t2.xlarge	\$0	\$0.214	\$0.214

Abbildung 5.3:
Beispielpreise im AWS Marketplace [Amag]

Amazon bietet etwa 120 selbst entwickelte Services für die AWS Cloud an, die jeweils mit nutzungsabhängigem Preis vergütet werden [Amal]. Außerdem stellt Amazon einen Marketplace zur Verfügung, auf dem Entwickler selbst entwickelte Lösungen anbieten können [Amag]. Diese Services basieren auf der AWS Cloud, so kann der AWS-Account und die AWS-Zahlungsmethoden genutzt werden. Amazon überprüft alle externen Lösungen auf Sicherheit und Funktionalität [Amag]. Firmen und Privatpersonen können den Marketplace nutzen, um die Lösungen zu kaufen und einzurichten. Der Kaufpreis besteht aus den Kosten der Software und der AWS-Basis, welche nutzungsbasiert pro Stunde berechnet wird. Manche Lösungen werden als Open-Source-Lösungen angeboten, dann muss lediglich die AWS-Basis gezahlt werden. Durch die klar definierte Beschreibung im Marketplace, die gut verständlichen Preise und die große Vielfalt an unterschiedlichen Services, entsteht eine Vielzahl von nutzbaren Lösungen.

Es gibt zehn Kategorien im Marketplace: *Sicherheit, BI & Big Data, Internet of Things, Netzwerkinfrastruktur, Datenbank, Mobilgeräte, Speicherung, Geographic Information Systems (GIS), Anwendungsentwicklung* und *Open-Source-Software*. Diese sind nicht notwendigerweise disjunkt und unterteilen sich jeweils wiederum in Unterkategorien. Dadurch wird deutlich die Vielfalt an Lösungen sichtbar, die AWS durch die externen Services im Marketplace anbieten kann.

AWS bietet durch interne und externe Services eine große Anzahl an Lösungen, die durch ihre Diversität bestehen. Strato kann mit sieben Tools nur wenig Möglichkeiten zu Erweiterungen bieten. Um den Meilenshop zu erweitern oder auch ganz neue Services einzuführen, ist die AWS Cloud die bessere Alternative.

5.5 Fazit

Die Infrastrukturen wurden mit den vier Dimensionen des Teufelsquadrats bewertet. Dabei tendieren die Bewertungen der Qualität und des Leistungsumfangs stark zur Cloud-Infrastruktur. Die Kosten sind bei der Serverlösung etwas niedriger, dafür verfügt die Cloud im kleinsten möglichen AWS Paket über mehr Kapazitäten. Dies kann positiv sein, wenn mehr Rechenleistung benötigt wird, aber momentan steigt dadurch vor allem der Preis. Beim Faktor Zeit kann schwer eine Infrastruktur als Sieger festgelegt werden, da die Umsetzung der Änderungen viel Zeit gekostet hat. Durch den Gewinn an Qualität kann die Ausführungszeit aber deutlich verringert werden und so amortisiert sich die Entwicklungszeit schnell.

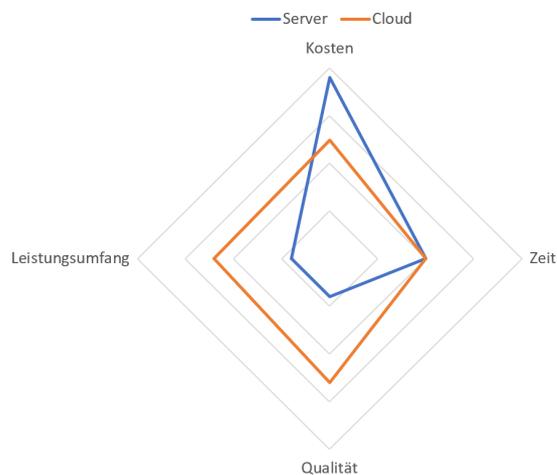


Abbildung 5.4:
Evaluation im Netzdiagramm

Bei den Dimensionen Kosten & Zeit wurde eine Differenz gebildet.

Vieles spricht für die Cloud, beispielsweise das schnellere und häufigere Deployment und die Möglichkeit, die Anwendung zu skalieren. Außerdem vermittelt die Cloud mehr Sicherheit und eine beliebige Erweiterbarkeit in alle Richtungen durch den Marketplace. Durch das Monitoring der eigenen Anwendung kann spezifisch auf alle Belange eingegangen werden. Zwar ist die Abrechnung der Cloud flexibler, dadurch können jedoch auch Risiken entstehen. Die Flexibilität der Kosten macht eine engmaschige Kostenkontrolle notwendig. Durch diese Skalierung kommt schnell das Gefühl von Unsicherheit über die Kosten auf.

Alles in Allem kann gesagt werden, dass es eine klare Empfehlung für die Transition des Meilenshops auf die Cloud gibt. Die Vorteile überwiegen im Vergleich zur Serverlösung, jedoch müssen die Risiken im Bereich der Kosten überwacht werden.

Kapitel 6

Zusammenfassung & Ausblick

In diesem letzten Kapitel werden die Erkenntnisse der vorliegenden Arbeit zusammengefasst und Schlussfolgerungen abgeleitet. Außerdem wird ein Überblick über kommende Weiterentwicklungen auf diesem Gebiet gegeben.

Zusammenfassung

Bisher wurde der Meilenshop auf einem On-Premise Linux Server von Strato ausgeführt. Wegen der starren Ressourcen, die zum Teil nicht benötigt wurden und der schlechten Übersicht eines Servers, kam die Idee einer Cloud-Transformation auf. Gemeinsam mit den Meilenwölfen und dem gesamten Entwicklungsteam wurden die Anforderungen an die neue Infrastruktur aufgestellt. Nach Konsultation des Gartner Quadrats wurde die AWS Cloud als zu untersuchende Infrastruktur ausgewählt. Der Wechsel auf die neue Infrastruktur wurde genutzt, um einige Änderungen im Bereich DevOps durchzuführen. Besonders die Einführung einer Development Pipeline und die Benutzung von Docker-Containern wurden in diesem Bereich forciert. Welche Infrastruktur besser zu den Bedürfnissen des Meilenshops passt, wurde anhand des Teufelsquadrats für Projektmanagement evaluiert. Die vier Dimensionen des Quadrats sind die Qualität – die mithilfe des Service Measurement Indexes evaluiert wurde – der Leistungsumfang, die Zeit und die Kosten – die nach dem TCO konzipiert wurden. In der Qualität und dem Leistungsumfang konnte die Cloud vollkommen überzeugen, wohingegen die Kosten ein wenig höher als beim Server waren. Die verwendbaren Ressourcen sind aber gerade bei der Datenbank auch bedeutend großzügiger. Beim Server wurde keine zusätzliche Entwicklungszeit benötigt, während für die Cloud Transformation eine Umsetzung in Docker notwendig war. Diese Entwicklungszeit wird langfristig durch die Ausführungszeit amortisiert. Die Transition des Meilenshops auf die Cloud kann als Erfolg gesehen werden, da das Entwicklungsteam damit entlastet wird. Außerdem kann die Cloud individuell auf die Anwendung und den Verbrauch skaliert werden. Durch die Anwendung von Docker konnte die Qualität der Lösung stark verbessert und

Ausführungszeit reduziert werden. Jedoch muss stark auf den Kostenrahmen geachtet werden, da – durch die Skalierbarkeit – schnell mehr Ressourcen genutzt werden können als vorgesehen. Nach dem Teufelsquadrat kann hier eine klare Empfehlung für Infrastruktur der Cloud im Falle des Meilenshops gegeben werden.

Ausblick

Das Ergebnis dieser Arbeit kann als Grundlage für die Frage genutzt werden, wann eine Cloud Transformation erfolgversprechend ist. Um genauere Ergebnisse zu erhalten, muss der Shop auf der Cloud im Langzeitbetrieb getestet werden. Außerdem müssen die theoretischen Grundlagen der Bewertung, wie die Zertifizierungen, Normen und Ausfallraten in der Realität überprüft werden. Wichtig wäre herauszufinden, wie groß der Anlernaufwand für neue Entwickler ist und, ob die Meilenwölfe ohne IT-Hintergrund ebenfalls die Lösung verwenden können.

Die vorliegende Arbeit gibt an, dass eine Transition in die Cloud ebenfalls für kleine Projekte sinnvoll sein kann. Da es sich hier allerdings um eine Fallstudie handelt, ist eine Verallgemeinerung auf andere Projekte und Anwendungsfälle nicht zwingend gegeben. Es sollte überprüft werden, ob die Annahmen auch für Projekte mit einem anderen Technologiestack gelten und wie weitere Indikatoren, wie Budget oder besondere Projektanforderungen die Wahl beeinflussen. Gerade Themen wie Support und Beratung wurden bei diesem kleinen Projekt nicht betrachtet. Wie sich die Nutzung einer Cloud Instanz ohne persönliche Beratung auf Projekte in diesem Bereich auswirkt, muss gesondert eruiert werden. Genauso handelt es sich bei dem vorliegenden Projekt um ein Beispiel, welches selbst für andere Projekte in einem ähnlichen Umfeld nicht zwingend aussagekräftig ist.

In der vorliegenden Fallstudie wurden einige Rahmenbedingungen im Vorfeld festgelegt. So würde sich die Bewertung der unterschiedlichen Kriterien sicherlich ändern, wenn eine andere Cloud-Plattform oder sogar ein anderes Servicemodell verfolgt worden wäre. Außerdem hat der Meilenshop praktisch keine externen Abhängigkeiten zu anderen Anwendungen. Bei anderen Programmen mit umfassender Einbettung in eine Anwendungslandschaft könnten vollkommen neue Probleme hinzukommen.

Unabhängig davon sollte betrachtet werden, wann die Cloud nicht die richtige Lösung ist. Beispielsweise sollte es bei enorm sicherheitskritischen Anwendungen noch weitere Erörterungen geben. Dennoch lässt die vorliegende Arbeit darauf schließen, dass viele dieser Eventualitäten trotzdem mithilfe einer Cloudlösung gewinnbringend umgesetzt werden können.

Literatur

- [Aic14] Aichele, C.; Schönberger, M. *IT-Projektmanagement: Effiziente Einführung in das Management von Projekten*. Berlin: Springer Vieweg, 2014.
- [Amaa] Amazon Web Services Inc. *Amazon CloudWatch - Der vollständige Überblick über Ihre Cloud-Ressourcen und -Anwendungen*. URL: <https://aws.amazon.com/de/cloudwatch/> (besucht am 30.04.2019).
- [Amab] Amazon Web Services Inc. *Amazon EC2*. URL: <https://aws.amazon.com/de/ec2/> (besucht am 30.04.2019).
- [Amac] Amazon Web Services Inc. *Amazon Elastic Container Service - Preise*. URL: <https://aws.amazon.com/de/ecs/pricing/> (besucht am 30.04.2019).
- [Amad] Amazon Web Services Inc. *Amazon RDS for MySQL*. URL: <https://aws.amazon.com/de/rds/mysql/> (besucht am 30.04.2019).
- [Amae] Amazon Web Services Inc. *AWS Backup*. URL: <https://aws.amazon.com/de/backup/> (besucht am 30.04.2019).
- [Amaf] Amazon Web Services Inc. *AWS Certificate Manager*. URL: <https://aws.amazon.com/de/certificate-manager/> (besucht am 30.04.2019).
- [Amag] Amazon Web Services Inc. *AWS Marketplace: Benutzerhandbuch für AWS Marketplace-Anbieter*. URL: https://aws.amazon.com/marketplace/pp/B00NN8Y43U/ref=mkt_ste_catgtm_oss1p (besucht am 30.04.2019).
- [Amah] Amazon Web Services Inc. *AWS Support - Häufig gestellte Fragen*. URL: <https://aws.amazon.com/de/premiumsupport/faqs/> (besucht am 30.04.2019).
- [Amal] Amazon Web Services Inc. *AWS Support-Stufen im Vergleich*. URL: <https://aws.amazon.com/de/premiumsupport/plans/> (besucht am 30.04.2019).

- [Amaj] Amazon Web Services Inc. *AWS-Compliance-Programme*. URL: <https://aws.amazon.com/de/compliance/programs/> (besucht am 30.04.2019).
- [Amak] Amazon Web Services Inc. *AWS-Fallstudie: General Electric*. URL: <https://aws.amazon.com/de/solutions/case-studies/general-electric/> (besucht am 30.04.2019).
- [Amal] Amazon Web Services Inc. *AWS-Preise*. URL: <https://aws.amazon.com/de/pricing/> (besucht am 30.04.2019).
- [Amam] Amazon Web Services Inc. *Bereitstellen von Docker-Containern - auf Amazon Elastic Container Service (Amazon ECS)*. URL: <https://aws.amazon.com/de/getting-started/tutorials/deploy-docker-containers/> (besucht am 30.04.2019).
- [Aman] Amazon Web Services Inc. *Erstellen eines CloudWatch-Alarm für eine Instance*. URL: https://docs.aws.amazon.com/de_de/AWSEC2/latest/WindowsGuide/using-cloudwatch-createalarm.html (besucht am 30.04.2019).
- [Amao] Amazon Web Services Inc. *Netflix-Fallstudie*. URL: <https://aws.amazon.com/de/solutions/case-studies/netflix/> (besucht am 30.04.2019).
- [Ama15] Amazon Web Services Inc. *AWS Service-Level-Agreements*. Apr. 2015. URL: <https://aws.amazon.com/de/legal/service-level-agreements/> (besucht am 30.04.2019).
- [Ama18a] Amazon. *Unsere Geschichte: Was aus einer Garagen-Idee werden kann?* 2018. URL: <https://www.aboutamazon.de/unsere-geschichte-was-aus-einer-garagen-idee-werden-kann> (besucht am 30.04.2019).
- [Ama18b] Amazon Web Services Inc. *Automating Elasticity*. März 2018. URL: <https://docs.aws.amazon.com/whitepapers/latest/cost-optimization-automating-elasticity/introduction.html> (besucht am 30.04.2019).
- [Ama18c] Amazon Web Services Inc. *AWS Kundenvereinbarung*. Nov. 2018. URL: [https://d1.awsstatic.com/legal/aws-customer-agreement/AWS%20Customer%20Agreement%20-%20%5C%20DE_%5C%20\(2018-11-01\).pdf](https://d1.awsstatic.com/legal/aws-customer-agreement/AWS%20Customer%20Agreement%20-%20%5C%20DE_%5C%20(2018-11-01).pdf) (besucht am 30.04.2019).
- [Ama18d] Amazon Web Services Inc. *Einführung von Amazon Data Lifecycle Manager für EBS-Snapshots*. Juli 2018. URL: <https://aws.amazon.com/de/about-aws/whats-new/2018/07/introducing-amazon-data-lifecycle-manager-for-ebs-snapshots/> (besucht am 30.04.2019).

- [Ama19] Amazon Web Services Inc. *ISO-Zertifizierungen und -Dienste von AWS*. Jan. 2019. URL: <https://aws.amazon.com/de/compliance/iso-certified/> (besucht am 30.04.2019).
- [Bac12] Bachmann, T. *Entwicklung einer Methodik für die Migration der Datenbankschicht in die Cloud*. 2012.
- [Bar] Barr, J. *AWS Auto Scaling*. URL: <https://aws.amazon.com/de/autoscaling/> (besucht am 30.04.2019).
- [Bau09] Baun, C.; Kunze, M.; Ludwig, T. „Servervirtualisierung“. In: *Informatik-Spektrum* (2009), S. 197–205.
- [Ben08] G.; et al. Bengel. *Masterkurs Parallele und Verteilte Systeme: Grundlagen und Programmierung von Multicoreprozessoren, Multiprozessoren, Cluster und Grid*. Wiesbaden: Vieweg+Teubner / GWV Fachverlage GmbH Wiesbaden, 2008.
- [Bie12] Biebl, J. „Wofür steht Cloud Computing eigentlich?“ In: *Wirtschaftsinformatik & Management* 4.1 (2012), S. 22–29.
- [Bru09] R. Brugger. *Der IT Business Case: Kosten erfassen und analysieren, Nutzen erkennen und quantifizieren, Wirtschaftlichkeit nachweisen und realisieren*. 2. Aufl. Springer-Verlag, 2009.
- [Bun17] Bundesamt für Sicherheit in der Informationstechnik. *Anforderungskatalog Cloud Computing (C5)*. Sep. 2017. URL: https://www.bsi.bund.de/DE/Themen/DigitaleGesellschaft/CloudComputing/Anforderungskatalog/Anforderungskatalog_node.html (besucht am 30.04.2019).
- [Clo] CloudBees. *Pipeline*. URL: <https://www.cloudbees.com/sites/default/files/blog/pipeline-vis.png> (besucht am 30.04.2019).
- [Clo14] Cloud Service Measurement Initiative Consortium. *Service Measurement Index: Framework Version 2.1*. 2014.
- [Clo17] Cloud Industry. *Cloud Computing Models Demystified*. Juni 2017. URL: <https://www.cloudindustry.com/au/cloud-computing-models-demystified/> (besucht am 30.04.2019).
- [Con] Consulting & Solutions Software GmbH. *Cloud-Migration für kleinere und mittlere Unternehmen (KMU)*. URL: <https://www.cio.de/a/zitate-von-steve-jobs,2291212> (besucht am 30.04.2019).
- [Coo18] Coolman, A. *5 Hauptgründe, warum Projekte scheitern und wie man Katastrophen verhindern kann*. März 2018. URL: <https://www.wrike.com/de/blog/hauptgruende-warum-projekte-scheitern-und-wie-man-katastrophen-verhindern-kann/> (besucht am 30.04.2019).

- [Deu17a] Deutsche Gesellschaft für Qualität. *Heute ist Weltqualitätstag: Was macht Qualität aus?* Nov. 2017. URL: <https://www.dgq.de/aktuelles/news/heute-ist-weltqualitaetstag-was-macht-qualitaet-aus/> (besucht am 30.04.2019).
- [Deu17b] Deutsches Institut für Normung e. V. *DIN EN ISO/IEC 27001*. Juni 2017. URL: <https://www.din.de/de/mitwirken/normenausschuesse/nia/wdc-beuth:din21:269670716> (besucht am 30.04.2019).
- [Die15] Dietl, W. *Downtime und Cloud-Verfügbarkeit: Google, Amazon und Microsoft im Vergleich*. Jan. 2015. URL: <https://www.computerweekly.com/de/news/2240238335/Downtime-und-Cloud-Verfuegbarkeit-Google-Amazon-und-Microsoft-im-Vergleich> (besucht am 30.04.2019).
- [Exo] Exone. *Der Server – Funktionsweise, Typen und Aufbau einfach erklärt*. URL: <https://www.exone.de/ratgeber/der-server-einfach-erklaert/> (besucht am 30.04.2019).
- [EY 19] EY CertifyPoint. *Certificate number: 2013-009*. Jan. 2019. URL: https://d1.awsstatic.com/certifications/iso_27001_global_certification.pdf (besucht am 30.04.2019).
- [Fil16] Filho, W. *What is a deployment pipeline and how does it help software development teams?* Juli 2016. URL: <https://medium.com/the-making-of-appear-in/what-is-a-deployment-pipeline-and-how-it-helps-software-development-teams-6cb29917ceea> (besucht am 30.04.2019).
- [Fra15] Franz, K. *Handbuch zum Testen von Web- und Mobile-Apps: Testverfahren, Werkzeuge, Praxistipps*. 2., aktualisierte und erw. Aufl. Berlin: Springer Vieweg, 2015.
- [Gar13] Garg, S.; Versteeg, S.; Buyya, R. „A Framework for Ranking of Cloud Computing Services“. In: *Future Gener. Comput. Syst.* 29.4 (Juni 2013), S. 1012–1023.
- [Gei18] Geißler, O.; Ostler U. *Was ist ein Dedicated Server?* Feb. 2018. URL: <https://www.datacenter-insider.de/was-ist-ein-dedicated-server-a-686962/> (besucht am 30.04.2019).
- [Gra15] Grabner, A. *Continuous Delivery in der Praxis*. Sep. 2015. URL: <https://www.computerwoche.de/a/continuous-delivery-in-der-praxis,3215540,2> (besucht am 30.04.2019).
- [Gri04] Grimm, S. *Marketing für High-Tech-Unternehmen: Wie Sie Markt- und Technologiezyklen strategisch nutzen und beeinflussen*. Wiesbaden: Gabler Verlag, 2004.

- [Ham10] Hammerschmid, C. *Server: Wie Sie alle Kosten im Voraus berechnen*. Dez. 2010. URL: https://www.chip.de/artikel/Server-Wie-Sie-alle-Kosten-im-Voraus-berechnen-3_140145768.html (besucht am 30.04.2019).
- [Hän13] Hänig, S. *Cloud Computing – was Juristen raten*. Dez. 2013. URL: <https://www.computerwoche.de/a/cloud-computing-was-juristen-raten,2549013,3> (besucht am 30.04.2019).
- [Han16] Hanschke, I. „Agile Planung — nur so viel planen wie nötig“. In: *Wirtschaftsinformatik & Management* 8 (Aug. 2016).
- [Hof13] Hoffmann, D. *Software-Qualität. 2.*, aktualisierte u. korr. Aufl. 2013. Berlin und Heidelberg: Springer, 2013.
- [Hof18] A. Hoffmann. *Public-Cloud-Anbieter im Vergleich: AWS, Azure und Google*. Feb. 2018. URL: <https://blog.unbelievable-machine.com/public-cloud-anbieter-vergleich-1> (besucht am 30.04.2019).
- [Hos19] HostAdvice.com. *Deutschland Web Hosting Marktanteil 2019*. 2019. URL: <https://de.hostadvice.com/marketshare/de/> (besucht am 30.04.2019).
- [Hum11] Hummel, O. *Aufwandsschätzungen in der Software- und Systementwicklung kompakt*. Heidelberg: Spektrum Akademischer Verlag, 2011.
- [IDG18] IDG Business Media GmbH. *Zitate von Steve Jobs*. Jan. 2018. URL: <https://www.cio.de/a/zitate-von-steve-jobs,2291212> (besucht am 30.04.2019).
- [Int14] International Organization for Standardization. *System und Software-Engineering - Qualitätskriterien und Bewertung von System- und Softwareprodukten (SQuaRE)*. 2014.
- [Jäh16] Jähn, T. *Wie die Welt in die Kiste kam*. Mai 2016. URL: <https://www.ndr.de/kultur/geschichte/Wie-die-Welt-in-die-Kiste-kam,container368.html> (besucht am 30.04.2019).
- [Jen14] Jendrian, K. „Der Standard ISO/IEC 27001: 2013“. In: *Datenschutz und Datensicherheit - DuD* 38.8 (2014), S. 552–557.
- [Joh15] Johner, C. *ISO 25010 und ISO 9126*. Aug. 2015. URL: <https://www.johner-institut.de/blog/iec-62304-medizinische-software/iso-9126-und-iso-25010/> (besucht am 30.04.2019).
- [Kim16] Kim, G. *Projekt Phoenix: Der Roman über IT und DevOps - Neue Erfolgsstrategien für Ihre Firma*. Heidelberg: O'Reilly, 2016.
- [Kir17a] Kirsch, M. *Was ist ein virtueller Server?* Okt. 2017. URL: <https://www.datenschutzbeauftragter-info.de/was-ist-ein-virtueller-server/> (besucht am 30.04.2019).

- [Kir17b] Kirsch, M.; et al. *Was ist ein virtueller Server?* Okt. 2017. URL: <https://www.datenschutzbeauftragter-info.de/was-ist-ein-virtueller-server/> (besucht am 30.04.2019).
- [Kom16] Kompetenznetzwerk Trusted Cloud e.V. *Cloud-Standards und Zertifizierungen im Überblick*. Feb. 2016. URL: https://www.trusted-cloud.de/sites/default/files/beitrag-cloud-standards_und_zertifizierungen_im_ueberblick.pdf (besucht am 30.04.2019).
- [Kos] Kosutic, D. *Überblick über ISO 27001:2013 Anhang A*. URL: <https://advisera.com/27001academy/de/knowledgebase/ueberblick-uber-iso-270012013-anhang-a/> (besucht am 30.04.2019).
- [Krc16] Krcmar, H.; Leimeister, J.; et al. *Cloud-Services aus der Geschäftsperspektive*. Wiesbaden: Springer Gabler, 2016.
- [Lan13] Lansing, J.; Schneider, S.; Sunyaev, A. *Cloud Service Certifications: Measuring Consumers' Preferences For Assurances*. ECIS 2013 Completed Research, Jan. 2013.
- [Let19] Let's Encrypt. *Let's Encrypt is a free, automated, and open Certificate Authority*. 2019. URL: <https://letsencrypt.org/> (besucht am 30.04.2019).
- [Lig09] P. Liggesmeyer. *Software-Qualität: Testen, Analysieren und Verifizieren von Software*. 2. Aufl. s.l.: Spektrum Akademischer Verlag, 2009.
- [Mei18a] Meilenwölfe e.V. *Meilenwölfe Blog*. Dez. 2018. URL: <https://meilenwoelfe.de/meilenblog/> (besucht am 30.04.2019).
- [Mei18b] Meilenwölfe e.V. *Meilenwölfe Shop*. Dez. 2018. URL: <https://shop.meilenwoelfe.de/> (besucht am 30.04.2019).
- [Mei18c] Meilenwölfe e.V. *Meilenwölfe Website*. Dez. 2018. URL: <https://meilenwoelfe.de/> (besucht am 30.04.2019).
- [Min] Mindlands Solutions UG. *Netzwerksicherheit beim Hosting im Firmennetzwerk und bei externen Anbietern*. URL: <https://www.netzwerke.com/netzwerksicherheit-beim-hosting.html> (besucht am 30.04.2019).
- [Möl16] Möller, S. *ISO-Zertifikat für die Cloud stärkt Vertrauen in Datensicherheit*. Dez. 2016. URL: <https://www.digitale-exzellenz.de/iso-zertifikat-fur-die-cloud-starkt-vertrauen-in-datensicherheit/> (besucht am 30.04.2019).
- [Mou16] Mouat, A. *Docker: Software entwickeln und deployen mit Containern*. Heidelberg: dpunkt.verlag, 2016. URL: <http://proquestcombo.safaribooksonline.com/9781492019688>.

- [Mus18] Musiienko, I. *Das Beste wählen: Vergleichende Analyse zu AWS vs. Azure vs. Google Cloud*. Juni 2018. URL: <https://www.infopulse.com/de/blog/das-beste-waehlen-vergleichende-analyse-zu-aws-vs-azure-vs-google-cloud/> (besucht am 30.04.2019).
- [Nak18] O. Nakar. *New Docker-based Dev Pipeline: Microservice Projects Just Got A 'Speed-Boost'*. Nov. 2018. URL: <https://www.imperva.com/blog/new-docker-based-dev-pipeline-microservice-projects-just-got-a-speed-boost/> (besucht am 30.04.2019).
- [Obe09] Obermeier, T.; Gasper, R. *Investitionsrechnung und Unternehmensbewertung*. München: Oldenbourg, 2009.
- [ODo15] O'Donnell, J. *iceberg-graphic*. Aug. 2015. URL: <https://arrayarchitects.com/iceberg-graphic/> (besucht am 30.04.2019).
- [Pie10] Pietsch, W.; Krams, B. *Vom Projekt zum Produkt: Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschaftsinformatik*. Bd. 178. Ges. für Informatik, 2010.
- [Prö18] Pröpfer, N. *Agile Techniken für klassisches Projektmanagement: Qualifizierung zum PMI-ACP*. mitp Professional. mitp, Dez. 2018.
- [Ran12] Randelhoff, M. *Malcolm McLean: Der Vater des Containertransports*. Apr. 2012. URL: <https://www.zukunft-mobilitaet.net/9121/vergangenheit-verkehrsgeschichte/malcolm-mclean-containerschiff-erfinder-container-teu/> (besucht am 30.04.2019).
- [Rao16] Raoul, H.; Leyh, C. „Cloud Computing: Gestern, heute, morgen“. In: *HMD Praxis der Wirtschaftsinformatik* 53.5 (2016), S. 563–579.
- [Rei18a] Reimers, N. *Strato Erfahrungen - Webhosting*. Aug. 2018. URL: <https://www.webhosterwissen.de/webhosting/webhosting-vergleich/strato/> (besucht am 30.04.2019).
- [Rei18b] Reimers, N. *Strato Erfahrungen - Webhosting*. Aug. 2018. URL: <https://www.webhosterwissen.de/webhosting/webhosting-vergleich/strato/> (besucht am 30.04.2019).
- [Rep10] Repschläger, J.; Pannicke, D.; Zarnekow, R. „Cloud Computing: Definitionen, Geschäftsmodelle und Entwicklungspotenziale“. In: *HMD Praxis der Wirtschaftsinformatik* (2010), S. 6–15.
- [Rig] RightScale. *Get the RightScale State of the Cloud Report*. URL: <https://www.rightscale.com/lp/state-of-the-cloud> (besucht am 30.04.2019).

- [Rix18] Rixecker, K. *Gartner: AWS und Microsoft liegen weit vor der Cloud-Konkurrenz*. Juni 2018. URL: <https://t3n.de/news/gartner-aws-microsoft-cloud-831619/> (besucht am 30.04.2019).
- [Sch14a] Schouten, E. *Cloud computing defined: Characteristics & service levels*. Jan. 2014. URL: <https://www.ibm.com/blogs/cloud-computing/2014/01/31/cloud-computing-defined-characteristics-service-levels/> (besucht am 30.04.2019).
- [Sch14b] Schubert, C.; Kläs, M.; Wagner, S. „Softwarequalität erfassen und vergleichen: Von der Messung bis zum abstrakten Qualitätsattribut“. In: *Objektspektrum* 2014.01/2014 (Jan. 2014), S. 50–55.
- [Sch15a] Schmitt, H.; Gorski, P. *USecureD - Qualitätsmodell*. Okt. 2015.
- [Sch15b] Schmitt, H.; Gorski, P.; Lo Iacono, L. *Qualitätsmodell: Deliverable E 1.4 (Version 2)*. 2015.
- [Sch15c] Schneider, S.; Sunyaev, A. *Cloud-Service-Zertifizierung: Ein Rahmenwerk und Kriterienkatalog zur Zertifizierung von Cloud-Services*. Berlin: Springer Gabler, 2015.
- [Sch19] Schmerer, K. *Gartner: Public Cloud Services wachsen 2019 um 17,5 Prozent*. Apr. 2019. URL: <https://www.zdnet.de/88357429/gartner-public-cloud-services-wachsen-2019-um-175-prozent/> (besucht am 30.04.2019).
- [Sie12] Siegel, J.; Perdue, J. *Cloud Services Measures for Global Use: The Service Measurement Index (SMI)*. 2012.
- [Sie14] J. Siegel J.; Perdue. „Service Measurement Index - Framework Version 2.1“. In: (2014).
- [Sok17] Sokenou, D. „Testen und Docker anhand eines Beispiels aus der Praxis“. In: *Softwaretechnik-Trends* 37.1 (2017).
- [Sta18a] Stadtjugendring Wolfsburg. *Stadtjugendring Wolfsburg Website*. Dez. 2018. URL: <https://www.stjr.de/news/> (besucht am 30.04.2019).
- [Sta18b] Statista. *Bevölkerung in Deutschland nach Einstellung zur Aussage „Bei den meisten Produkten kommt es mir eher auf die Qualität als auf den Preis an“ von 2015 bis 2018 (in Millionen)*. Nov. 2018. URL: <https://de.statista.com/statistik/daten/studie/172177/umfrage/einstellung-qualitaet-wichtiger-als-preis/> (besucht am 30.04.2019).

- [Stra] Strato AG. *Häufige Fragen zu den STRATO Service Level Agreements (SLA)*. URL: <https://www.strato.de/faq/server/haeufige-fragen-zu-den-strato-service-level-agreements-sla/> (besucht am 30.04.2019).
- [Strb] Strato AG. *Managed Server*. URL: <https://www.strato.de/server/managed-server/> (besucht am 30.04.2019).
- [Strc] Strato AG. *Mit einem Backup vom Server Website und Daten sichern*. URL: <https://www.strato.de/server/backup-server/> (besucht am 30.04.2019).
- [Strd] Strato AG. *Über uns - Von der Domain bis zur High-End-Lösung*. URL: <https://www.strato.de/ueber-uns/> (besucht am 30.04.2019).
- [Stre] Strato AG. *V-Server Linux*. URL: <https://www.strato.de/server/linux-vserver/> (besucht am 30.04.2019).
- [Str15] Strato AG. *Service Level Agreement (SLA)*. Apr. 2015. URL: https://www.strato.de/_assets_de/download/server/texte/service-level-agreement.pdf (besucht am 30.04.2019).
- [Tag19] Tagesschau.de. *Digitalisierung - Volkswagen baut Cloud mit Amazon*. März 2019. URL: <https://www.tagesschau.de/wirtschaft/vw-amazon-103.html> (besucht am 30.04.2019).
- [Tur14] Turnbull, J. *The Docker Book: Containerization Is the New Virtualization*. James Turnbull, 2014.
- [TÜV16] TÜV SÜD Management Service GmbH. *Zertifikat*. Juli 2016. URL: https://www.strato.de/_assets_de/download/sicherheit/strato-iso-27001.pdf (besucht am 30.04.2019).
- [VfL18] VfL Wolfsburg-Fußball GmbH. *VfL Wolfsburg Website*. Dez. 2018. URL: <https://www.vfl-wolfsburg.de/> (besucht am 30.04.2019).
- [Vos13] Vossen, G.; Haselmann, T.; Hoeren, T. *Cloud-Computing für Unternehmen: Technische, wirtschaftliche, rechtliche und organisatorische Aspekte*. 1. Aufl. Heidelberg: dpunkt-Verl., 2013.
- [Wil04] S. Wild M.; Herges. *Total Cost of Ownership (TCO): Ein Überblick*. Mainz, Juni 2004.
- [Wit16] Witte, F. *Testmanagement und Softwaretest: Theoretische Grundlagen und praktische Umsetzung*. 1. Auflage. Wiesbaden: Springer Vieweg, 2016.
- [Wöh08] Wöhe, G.; Döring, U. *Einführung in die allgemeine Betriebswirtschaftslehre*. 23., vollst. neu bearb. Aufl. Vahlens Handbücher der Wirtschafts- und Sozialwissenschaften. München: Vahlen, 2008.

- [Wol16] Wolff, E. *Continuous Delivery: Der pragmatische Einstieg*. 2. Aufl. dpunkt, 2016.
- [Wur17] Wurbs, P. *Docker versus VM: Wie Container die heutige IT verändern*. Aug. 2017. URL: <https://jaxenter.de/docker-vs-vm-54816> (besucht am 30.04.2019).