



Diplomarbeit

Konzeptionierung und Bewertung einer Cloud-basierten Multimodellspeicherung

Christian Braun

Geboren am: 30. Juni 1998 in Dresden

Matrikelnummer: 4631398

Immatrikulationsjahr: 2016

zur Erlangung des akademischen Grades

Diplom-Ingenieur (Dipl.-Ing.)

Betreuer

Dr.-Ing. Iris Braun

Dipl.-Medieninf. Michael Polter

Betreuender Hochschullehrer

Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill

Eingereicht am: 13. Januar 2023



Aufgabenstellung für die Diplomarbeit

Für: Christian Braun

Matrikel-Nr.: 4631398

Studiengang: Informationssystemtechnik

Thema: Konzeptionierung und Bewertung einer Cloud-basierten Multimodellspeicherung

Im Rahmen der Arbeit sollen folgende Aufgaben bearbeitet werden (Listung der Teilaufgaben):

Multimodelle werden im Bauingenieurwesen benutzt, um domänenübergreifende Informations- räume miteinander zu verbinden [1]. Dies geschieht durch die Verknüpfung von Elementen verschiedener Modelle. Beispielsweise lassen sich somit Bauteile aus einem Gebäudemodell mit Kostenelementen eines anderen Modells verknüpfen.

Moderne Software-Plattformen zur Berechnung komplexer Probleme beim Entwurf und Betrieb von Gebäuden nutzen Multimodelle, um Informationen verschiedenster Fachdisziplinen (z.B. Architekt, Statiker, TGA-Planer) zu verknüpfen. Die Multimodelle werden dabei meist dateibasiert im lokalen Verzeichnissystem abgelegt. Die Nutzung von Cloud-basierten Rechen- und Speicherressourcen ermöglicht es Unternehmen, solche Plattformen ohne Investitionen in neue Hardware zu nutzen. Jedoch stellt sich gerade bei kontinuierlichem Langzeiteinsatz solcher Plattformen die Frage nach Rentabilität des Cloud-Ansatzes gegenüber dem Hosting auf eigener Hardware.

Im Hinblick auf beispielhafte Workflows aus dem Bereich der Gebäudesimulation und die Kostenmodelle von Cloud-Anbietern soll in dieser Arbeit untersucht werden, wie Multimodelle Cloudbasiert am effektivsten und kostengünstigsten gespeichert werden können. Dabei sind insbesondere voneinander abhängige Simulationsaufgaben zu berücksichtigen, welche ein Vielfaches Lesen und Schreiben der Teilmodelle eines Multimodells verursachen. Für das beispielhafte Einsatzszenario ist eine geeignete Datenspeicherstrategie zu entwerfen und prototypisch zu implementieren, um die Umsetzbarkeit des entwickelten Ansatzes zu evaluieren.

Angestrebte Arbeitsergebnisse:

- Gegenüberstellung der Kostenmodelle gängiger Cloud-Anbieter
- Untersuchung cloudbasierter Datenbanksysteme (z.B. Amazon S3, Google Cloud Storage)
- Entwurf und Gegenüberstellung von mindestens zwei Cloud-basierten Datenspeicherstrategien
- prototypische Implementierung einer ausgewählten Strategie

[1] [https://tu-](https://tu-dresden.de/bu/bauingenieurwesen/ressourcen/dateien/postgraduales/pdf/Fuchs.pdf?lang=de)

[dresden.de/bu/bauingenieurwesen/ressourcen/dateien/postgraduales/pdf/Fuchs.pdf?lang=de](https://tu-dresden.de/bu/bauingenieurwesen/ressourcen/dateien/postgraduales/pdf/Fuchs.pdf?lang=de)

Betreuer/in: Dr.-Ing. Iris Braun, Michael Polter (TUD Baulng)

1. Prüfer/in: Prof. Dr. Alexander Schill

2. Prüfer/in: Dr.-Ing. Iris Braun

Die Arbeit wird in deutscher / englischer Sprache verfasst.

Ausgehändigt: 22.07.2022

Einzureichen: 23.12.2022

Vorsitzender des Prüfungsausschusses

Verantwortlicher Hochschullehrer

Martin

Wollschlaeger

Herr Prof. Dr.-Ing. habil. Martin Wollschlaeger

Digital unterschrieben von Martin
Wollschlaeger
Datum: 2022.07.18 12:58:18 +02'00'

A. Schill

Digital unterschrieben
von Alexander Schill
Datum: 2022.07.14
16:06:55 +02'00'

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich das vorliegende Dokument mit dem Titel *Konzeptionierung und Bewertung einer Cloud-basierten Multimodellspeicherung* selbstständig und ohne unzulässige Hilfe Dritter verfasst habe. Es wurden keine anderen als die in diesem Dokument angegebenen Hilfsmittel und Quellen benutzt. Die wörtlichen und sinngemäß übernommenen Zitate habe ich als solche kenntlich gemacht. Es waren keine weiteren Personen an der geistigen Herstellung des vorliegenden Dokumentes beteiligt. Mir ist bekannt, dass die Nichteinhaltung dieser Erklärung zum nachträglichen Entzug des Hochschulabschlusses führen kann.

Dresden, 13. Januar 2023

Christian Braun

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Ziele der Arbeit	2
1.3	Abgrenzung der Arbeit	2
1.4	Überblick über die Arbeit	3
2	Grundlagen und verwandte Arbeiten	4
2.1	Datenmodelle im Bauwesen	4
2.2	Multimodelle	4
2.2.1	Linkmodelle	5
2.2.2	Fachmodelle	5
2.3	Cloud-Anbieter/-Technologien	5
2.3.1	Service-Modelle	5
2.3.2	Kostenmodelle	7
2.4	Sicherheitsvorkehrungen	8
2.4.1	Georedundanz	8
2.4.2	RAID-Verbund	8
2.4.3	C5-Zertifikat	8
2.5	Verwandte Arbeiten	9
2.5.1	Anwendung von Techniken des Cloud Computing zur Visualisierung und Bearbeitung von BIM	9
2.5.2	Kontextabhängiger Informationsraum für Informationsprozesse im Bau- wesen	9
2.5.3	Alles an seinem (Schreib-)Platz: Abstraktion von Cloud-Speichern mit- tels Objekt-Speicher	10
3	Analyse von Multimodellen und Cloud-Anbietern	11
3.1	Aufbau eines Multimodell-Containers	11
3.2	Speicherarten von Modelltypen	12
3.2.1	Index-Datei	12
3.2.2	Fachmodelle	14
3.2.3	Linkmodelle	14
3.3	Vergleich von Cloud-Anbietern	16
3.3.1	Kriterien	16
3.3.2	Auswahl von Cloud-Anbietern	18
3.4	On-Site Speicherung	25

4	Konzept	26
4.1	Kollaboratives Arbeiten	26
4.2	Speichermethoden	26
4.3	Strategien zur kostenoptimierten Cloudspeicherung von Multimodellen . . .	28
4.3.1	Modellorientierte Kostenoptimierung	29
4.3.2	Matrix-basierte Kostenoptimierung	36
4.4	Fazit	44
5	Implementierung	45
5.1	Umsetzung der Eingabeparameter	47
5.2	Initialisierung des Produktportfolios	48
5.3	Analyse des Multimodell-Containers	49
5.3.1	ICDD-Framework	49
5.3.2	XML-Parsing	50
5.4	Mapping von Modell und Cloud-Produkt	53
5.5	Cloud-Produkte mit Festpreis-Kostenmodell	55
5.6	Cloud-Produkte mit Pay-per-Use-Kostenmodell	55
5.7	Ausgabe als Datei	56
6	Evaluation	58
6.1	Evaluation der Implementierung	58
6.2	Evaluation der Speicherstrategie	62
7	Zusammenfassung und Ausblick	63
7.1	Zusammenfassung der Diplomarbeit	63
7.2	Ausblick und zukünftige Themen	64

1 Einleitung

Das Errichten von Gebäuden ist seit Beginn des Bauingenieurwesens ein sehr umfangreiches und zeitaufwendiges Unterfangen. Aufgrund dessen Komplexität und der Vielzahl beteiligter Parteien ist eine gute Planung und Kommunikation für das Gelingen eines solchen Unterfangens notwendig. Fehler in der Planung, welche erst während der Bauphase bemerkt werden, sind nur sehr schwer und kostspielig zu korrigieren.

Die Modellbasierte Planung hat eine sehr lange Tradition im Bauingenieurwesen; dabei gewinnen die am Computer erstellten Modelle immer mehr an Bedeutung. Die verschiedenen Modelle bilden jeweils fachlich relevante Informationen virtuell ab und können so bereits vor dem eigentlichen Errichten des Gebäudes geprüft und gegebenenfalls korrigiert werden, zum Beispiel durch umfangreiche Simulationen. Für ein Gebäude entsteht während des Lebenszyklus eine Vielzahl an Modellen. Die Meisten davon im Vorhinein während der Planung, jedoch kommen bei Änderungen am Gebäude oder dessen Abriss weitere Modelle und Simulationen hinzu.

Die Anzahl der Modelle und Arten von Modellen steigen stetig. Daher ist es von besonderer Relevanz, dass alle Modelle fachlich eindeutig und unmissverständlich sind. Nur wenn alle beteiligten Parteien Informationen einheitlich interpretieren, ist ein Gelingen von Bauvorhaben möglich.

1.1 Motivation

Das Zusammenführen von Modellen verschiedener fachlicher Aspekte kann zu einem deutlichen Gewinn an Informationen führen, zum Beispiel welche Ressourcen und deren Lieferanten in jeder Phase des Baufortschritts benötigt werden. Das Multimodell-Konzept erlaubt es erstmals, verschiedene Fachmodelle lose zu koppeln. Das bedeutet, die Informationen zur Verlinkung mehrerer Modelle werden außerhalb, in separaten Linkmodellen, gespeichert. Fachbezogene Anwendungen können deren Modelle ohne Beeinträchtigungen weiterverwenden, weil am Modell keine Veränderungen vorgenommen werden. Das Konzept der losen Kopplung ermöglicht, das gesamte Multimodell-Konzept in nahezu jeden Fachbereich und jede Domäne zu übertragen.

In der Domäne des Bauingenieurwesens können durch die Kopplung von Fachmodellen zusätzliche Informationen über den gesamten Bauprozess gewonnen werden. Dadurch ist eine bessere Planung und Abstimmung aller Vorgänge am Gebäude möglich. Zum Beispiel können Simulationen auf verschiedene Phasen des Baufortschritts angewandt werden. Durch diese Informationen können mögliche Komplikationen noch während der Planungsphase erkannt und behoben werden.

Für das kollaborative Arbeiten an einem Multimodell müssen die einzelnen Bestandteile zuverlässig und an zentralen Orten gespeichert werden. Es ist besonders wichtig, dass für alle Nutzer zu jeder Zeit die aktuellste Version eines jeden Modells zur Verfügung steht. Die Speicherung der Modelle kann auf unterschiedliche Art und Weise erfolgen, muss aber ein kollaboratives Arbeiten unterstützen. Geeignet dafür wären zum Beispiel firmeneigene Server und Netzwerke oder verschiedene Cloud-Lösungen. Insbesondere Cloud-Anbieter im Bereich der Public Cloud bieten das Potential, einen unkomplizierten Zugriff für alle Parteien zu ermöglichen.

Eine Option der lokalen Speicherung bietet der Multimodell-Container (MMC), dieser ist ein ZIP-Archiv einschließlich festgelegter Verzeichnisstrukturen. Das beinhaltet zum Beispiel eine Index-Datei mit allen Modellen inklusive deren Ablageorten innerhalb des Archivs. Aufgrund der steigenden Anzahl von Fachmodellen ist eine lokale Speicherung für kollaboratives Arbeiten nur begrenzt geeignet.

Es werden Alternativen für die Speicherung der Bestandteile von Multimodellen benötigt, welche auf das kollaborative Arbeiten ausgerichtet sind.

1.2 Ziele der Arbeit

Das Ziel dieser Arbeit ist die Entwicklung einer Strategie für die Speicherung von Multimodellen und deren enthaltenen Modellen mittels geeigneter Produkte von Public Cloud-Anbietern. Für die Vergleichbarkeit der verschiedenen Cloud-Produkte und deren Kostenmodellen, ist die Entwicklung einer einheitlichen Darstellung dieser notwendig. Die Auswahl von Cloud-Produkten soll die individuellen Eigenschaften eines jeden Modelltyps berücksichtigen und einer geeigneten Lösung aus dem Angebot an Speicherlösungen zuordnen. Die Speicher-Strategie darf die Anforderungen des kollaborativen Arbeitens nicht benachteiligen.

1.3 Abgrenzung der Arbeit

Diese Arbeit untersucht beispielhaft für die Domäne des Bauingenieurwesens die Überführung eines MMC in die Public Cloud.

Alle Betrachtungen beziehen sich auf die Transformation und Möglichkeiten zur Speicherung von bestehenden MMC und deren Inhalten. Die Erstellung und inhaltliche Modifikation eines MMC wird nicht behandelt.

Der MMC ist ein Konzept der *Fakultät Bauingenieurwesen, Technische Universität Dresden*. In dieser Arbeit werden keine realen Nutzungsverhalten von MMC analysiert. Es werden ausschließlich Annahmen zur Nutzung der einzelnen Bestandteile des MMC getroffen und anhand von Beispielen begründet. Die in einem MMC enthaltenen Fachmodelle werden als Blackbox behandelt. Das bedeutet, dass deren Inhalt und fachlichen Aussagen nicht weitergehend berücksichtigt werden.

Auf dem frei zugänglichen Markt existiert eine sehr große Anzahl von Cloud-Anbietern und Cloud-Produkten; ein vollständiges Abbild der verfügbaren Cloud-Anbieter wird nicht verfolgt. In dieser Arbeit werden beispielhaft einige Cloud-Anbieter und Cloud-Produkte für einen Vergleich herangezogen. Die beispielhaft herangezogenen Cloud-Produkte werden jeweils in deren Basis-Konfiguration betrachtet. Verbesserte Cloud-Produkte können, falls benötigt, als separates Cloud-Produkt dem Produktportfolio hinzugefügt werden.

Cloud-Produkte werden in dieser Arbeit nicht hinsichtlich ihrer technischen Leistung beurteilt. Das beinhaltet zum Beispiel die Übertragungsgeschwindigkeit vom Cloud-Anbieter zum Nutzer.

Die von Cloud-Anbietern genutzten Kostenmodelle sind in vielen Fällen individuell. Für den

Vergleich der Cloud-Produkte über mehrere Cloud-Anbieter hinweg werden diese angenähert abgebildet. Die in dieser Arbeit gewählte Annäherung kann unter Umständen Abweichungen zu dem tatsächlichen Kostenmodell enthalten. Es ist ebenfalls möglich, dass einzelne Kostenmodelle nicht abbildbar sind.

Diese Arbeit behandelt die Möglichkeit der Überführung von MMC in die Public Cloud mittels der Auswahl geeigneter Cloud-Produkte. Eine Bewertung von Alternativen, zum Beispiel einer Speicherung On-Site, wird nicht vorgenommen. Die weitergehende Betrachtung für notwendige Vorkehrungen, welche das kollaborative Arbeiten an einem MMC ermöglichen, sind nicht Bestandteil dieser Arbeit. Das betrifft beispielsweise ein Versionsmanagement und das Verwalten von Schreibzugriffen.

Der Umfang dieser Arbeit behandelt das Zusammenstellen von Cloud-Produkten, angepasst an die Anforderungen eines gestellten MMC. Das Ergebnis ist ein Vorschlag für eine verteilte Speicherung des MMC in der Public Cloud. Der Prozess zur Transformation des MMC hin zu einer verteilten Speicherung ist nicht Teil dieser Arbeit. Das betrifft auch die Integration der Schnittstellen verschiedener Cloud-Provider.

1.4 Überblick über die Arbeit

Das Kapitel zu den Grundlagen und verwandten Arbeiten gibt einen Überblick zum Stand der Forschung und Technik. Es werden wesentliche Begriffe und Eigenschaften der Modellbasierten Planung im Bauwesen und der Dienstleistungen von Cloud-Anbietern erläutert.

Im Kapitel zur Analyse von Multimodellen und Cloud-Anbietern werden die Eigenschaften von Multimodellen untersucht, damit die Funktionalität und Integrität aller Bestandteile bei einer verteilten Speicherstrategie sichergestellt ist. Ebenfalls wird das Angebot aktueller Cloud-Anbieter analysiert und welche Vor- und Nachteile dieses gegenüber dem Betreiben eigener Infrastruktur zur Speicherung von Modellen hat.

In dem Konzept-Kapitel werden die Möglichkeiten untersucht, die Bestandteile eines Multimodells mit dem Angebot bekannter Cloud-Anbieter zu kombinieren. Die dafür genutzte Speicherstrategie berücksichtigt die individuellen Eigenschaften eines jeden Modells. Dabei werden insbesondere die Anforderungen des kollaborativen Arbeitens und der technischen Machbarkeit berücksichtigt.

Das Kapitel der Implementierung behandelt die prototypische Realisierung der Speicherstrategie als Algorithmus am Beispiel der Domäne des Bauingenieurwesens sowie Entscheidungen zum Design der Software.

In der Evaluierung werden die Ergebnisse des Konzepts und dessen Implementierung zusammengefasst und ausgewertet. Auch wird dargelegt, ob sich die Vorteile der Cloud-Angebote auf das Konzept der Multimodelle übertragen lassen.

In dem Kapitel Zusammenfassung und Ausblick werden die Ergebnisse der Arbeit zusammengefasst dargestellt, des Weiteren werden einige Möglichkeiten für die Weiterentwicklung von Speicherstrategien für Multimodelle erläutert.

2 Grundlagen und verwandte Arbeiten

In diesem Kapitel werden die Grundlagen beschrieben, welche für das Verständnis der folgenden Kapitel und Konzepte benötigt werden. Zunächst wird dazu das Konzept von Multimodellen beziehungsweise Multimodell-Containern näher betrachtet. Anschließend werden die aktuellen Methoden zur Speicherung der verschiedenen Modelltypen und die Angebote verschiedener führender Cloud-Anbieter untersucht.

2.1 Datenmodelle im Bauwesen

Modelle sind eine vereinfachte Abbildung der Realität, welche das Original (bzw. Urbild) auf die für den Betrachter relevanten Informationen reduzieren[63, S. 131]. Aufgrund der Komplexität der Vorgänge innerhalb des Bauwesens ist seit "Anbeginn der Baukultur in allen Bereichen eine Arbeitsteilung" erforderlich[33, S. 10]. Schon lange vor dem Aufkommen der Rechentechnik wird eine modellbasierte Arbeitsweise verwendet. Zur Abbildung aller relevanten Informationen eines modernen Bauvorhabens sind eine Vielzahl von Modellen notwendig[33, S. 22]. Durch die Modellierung lassen sich Planungen aufeinander abstimmen, bevor eine kostspielige Realisierung stattfindet, zum Beispiel durch Simulationen für Statik, thermische Eigenschaften oder Szenarien zur Evakuierung. Für die Absprache mehrerer Parteien und das Treffen von Entscheidungen innerhalb der Bauprozesse muss ein einvernehmlich verständlicher Informationsaustausch sichergestellt sein[33, S. 39].

2.2 Multimodelle

Multimodelle ermöglichen die logische Verknüpfung von verschiedenen Fachmodellen und deren Elementen zur Bildung von domänenübergreifenden Informationsräumen. Innerhalb dieser domänenübergreifenden Informationsräume lassen sich weitere, für die Bauprozesse relevante, Informationen ableiten. Auf der Basis dieser zusätzlichen Informationen können weitere Optimierungen vorgenommen werden. Das Konzept der Multimodelle verwendet eine lose Kopplung von Fachmodellen, das bedeutet, dass Informationen zu Verknüpfungen von Fachmodellen außerhalb in separaten Linkmodellen gespeichert werden. Das Fachmodell wird nicht verändert und kann deshalb auch in der zugehörigen Fachanwendung ohne Einschränkungen weiter genutzt werden. Der Einsatz eines MMC ermöglicht den einheitlichen Datenaustausch eines solchen Multimodells[33, S. 41].

2.2.1 Linkmodelle

Linkmodelle bezeichnen die Modelle innerhalb eines Multimodells, welche die Verknüpfungen zwischen mehreren Elementarmodellen abbilden[33, S. 47]. Eine Verknüpfung referenziert mindestens zwei Elemente verschiedener Elementarmodelle[35, S. 207]. Ein Element ist ein Teil eines Modells, welches innerhalb des referenzierten Modells durch eine eindeutige ID identifiziert werden kann[34, S. 25].

2.2.2 Fachmodelle

Fachmodelle sind von Fachanwendungen erstellte Modelle. In einem Multimodell werden Fachmodelle mit beliebigen Datenformaten gebündelt. Alle Fachmodelle sind gleichgestellt und benötigen keinen Transformationsprozess, deshalb ist ein Weiterverwenden der Originaldaten problemlos möglich[33, S. 47].

2.3 Cloud-Anbieter/-Technologien

Der Begriff der Cloud beschreibt einen Pool von zusammengeschalteten virtuellen Computer-Ressourcen[14, S. 4]. Diese, als Verband koordinierten Ressourcen, werden von Cloud-Anbietern mittels verschiedener Service-Modelle zur Verfügung gestellt. Das Service-Modell definiert die Aufteilung der Verantwortlichkeiten zwischen dem Kunden und dem Cloud-Anbieter. Das Kostenmodell legt fest, welche Methode verwendet wird um den genutzten Service abzurechnen.

2.3.1 Service-Modelle

Service-Modelle legen fest, auf welcher technischen Ebene der Cloud-Anbieter die Leistung gegenüber dem Kunden erbringt. Die zu erbringende Leistung kann das Zurverfügungstellen von virtualisierten Ressourcen oder den vollständigen Betrieb eines Software-Produktes umfassen (s. Abbildung 2.1)[14, S. 4].

Viele Cloud-Produkte können in mehr als nur ein Service-Modell eingeordnet werden, fast jedes IaaS-Produkt kann Teil einer Plattform eines Cloud-Anbieters sein. Eine verwaltete Datenbank kann Teil des Produktportfolios einer Cloud-Plattform sein. Und ist zur gleichen Zeit gegebenenfalls auch ein SaaS-Produkt, welches eine durch den Cloud-Anbieter betriebene Software darstellt, welche dem Kunden zur Verfügung gestellt wird.

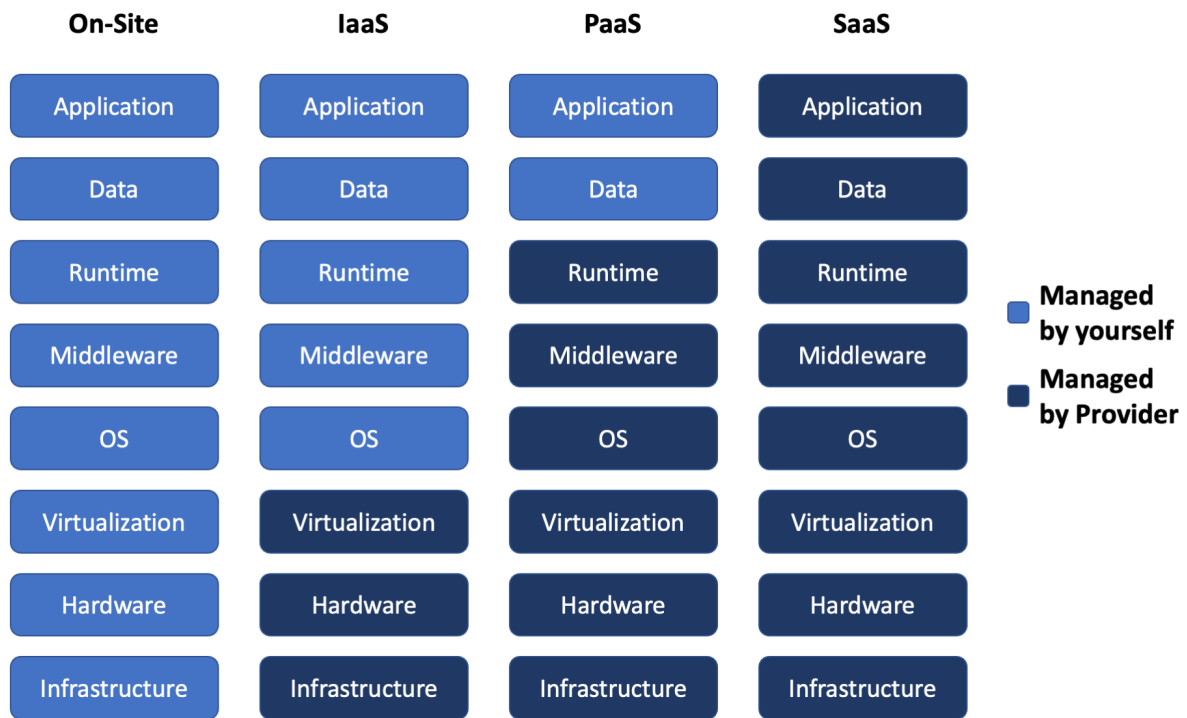


Abbildung 2.1: Übersicht Service-Modelle (Nach [68])

Infrastructure-as-a-Service (IaaS)

In dem Service-Modell Infrastructure-as-a-Service (IaaS) übernimmt der Cloud-Anbieter das Zurverfügungstellen von Ressourcen. Der Cloud-Anbieter verantwortet ausschließlich die für den Betrieb von Servern bzw. Virtuellen Maschinen notwendige Infrastruktur. Dies umfasst zum Beispiel die Stromversorgung, die Netzwerkinfrastruktur sowie den Betrieb der physischen Hardware einschließlich dessen Instandhaltung[69, S. 30]. Die Verantwortung für die Administration des Betriebssystems, der betriebenen Anwendungen sowie die dafür notwendigen Middleware-Komponenten (Software) verbleiben bei dem Kunden. Je nach Einsatz-Szenario ist eine aufwendige Betreuung der zu betreibenden Systeme durch den Kunden notwendig. Das Skalieren auf Infrastruktur-Ebene ist unter Umständen mit dem Aufwand der Einrichtung weiterer Systeme verbunden. Dieses Service-Modell ist weniger flexibel bei kurzfristigen Änderungen von benötigten Ressourcen.

Platform-as-a-Service (PaaS)

Bei dem Service-Modell Platform-as-a-Service (PaaS) stellt der Cloud-Anbieter dem Kunden eine Laufzeitumgebung für dessen Anwendungen zur Verfügung. Der Cloud-Anbieter übernimmt neben der Verantwortung für die Infrastruktur, ähnlich zu dem Service-Modell IaaS, ebenfalls die Verantwortung für den Betrieb und die Administration der Plattform. Eine Plattform beinhaltet ein Portfolio von Produkten, welche für den Betrieb einer Anwendung benötigt werden, zum Beispiel Datenbanken, Speicherplatz und eine Entwicklungsumgebung[69, S. 29]. Der Kunde ist für die Entwicklung und Administration seiner Anwendung sowie deren Kompatibilität zu der gewählten Plattform verantwortlich. Das Service-Modell PaaS bietet des Weiteren den Vorteil, dass die gesamte Administration zentral und die Skalierung automatisiert über den Cloud-Anbieter abgewickelt werden kann.

Bei der Anpassung von Anwendungen an die Plattform eines Cloud-Anbieters entsteht eine Abhängigkeit des Kunden gegenüber der Plattform. Die technische Realisierung von Plattformen verschiedener Cloud-Anbieter unterscheiden sich; deshalb wird bei dem Wechsel eines Cloud-Anbieters gegebenenfalls eine umfangreiche Anpassung der Anwendung notwendig.

Software-as-a-Service (SaaS)

In dem Service-Modell Software-as-a-Service (SaaS) übernimmt der Cloud-Anbieter alle für den Betrieb einer Anwendung notwendigen Aufgaben. Ähnlich zu dem Service-Modell PaaS stellt der Cloud-Anbieter die für den Betrieb von Anwendungen notwendige Infrastruktur und Plattform. Darüber hinaus ist dieser zuständig für die Administration, Aktualisierung, Lizenzierung und Fehlerbehebung der Anwendung. Das Service-Modell SaaS erlaubt eine sehr einfache Nutzung von Anwendungen in der Cloud, weil fast alle administrativen Aufgaben in der Verantwortung des Cloud-Anbieters liegen. Auch lassen sich Anwendungen hier gegebenenfalls effizienter und günstiger betreiben, weil mehrere Kunden sich Ressourcen effizient teilen können. Der Kunde profitiert ebenfalls von der flexiblen Skalierung durch den Anbieter, falls mehr Ressourcen benötigt werden[69, S. 28]. Bei der Nutzung des Service-Modells SaaS besteht gegebenenfalls eine direkte Abhängigkeit von einem Cloud-Anbieter und dessen Anwendung. Ein Wechsel zu einem anderen Cloud-Anbieter ist nur möglich, wenn dieser eine ähnliche Software anbietet.

2.3.2 Kostenmodelle

Ein Kostenmodell definiert die Methode zur Abrechnung von erbrachten Leistungen des Cloud-Anbieters gegenüber dem Kunden. Im Rahmen dieser Arbeit werden insbesondere die beiden Kostenmodelle *Festpreis* und *Pay-per-Use* betrachtet. Darüber hinaus finden weitere Kostenmodelle in der Praxis Anwendung, beispielsweise das Kostenmodell *Abonnement* (Microsoft Office 365), *Pay-per-Unit* (Kauf einer App) oder *Free* (Mail-Programm). Des Weiteren können mehrere Kostenmodelle miteinander kombiniert werden, zum Beispiel *Festpreis* + *Pay-per-Use* (Smartphone-Tarif mit Freiminuten)[13, S. 35ff.].

Festpreis

In dem Kostenmodell Festpreis, wird ein festgelegter Preis für eine bestimmte Leistung zur Verfügung gestellt. Wird eine Leistung über einen periodischen Zeitraum zur Verfügung gestellt, wird auch von dem Kostenmodell *Abonnement* gesprochen.

Der vereinbarte Preis beinhaltet das zur Verfügung stellen einer Leistung und ist unabhängig von der tatsächlichen Nutzung[13, Kap. 12.1]. Das Risiko für nicht zur Verfügung stehende Ressourcen liegt fast ausschließlich bei dem Kunden. Der Cloud-Anbieter erfüllt die der Produktbeschreibung entsprechende Leistung, welche zeitlich begrenzte Ausfälle toleriert (s. Abschnitt 3.3.1).

Pay-per-Use

Das Kostenmodell Pay-per-Use, auch *Nutzungsabhängige Abrechnung* genannt, basiert auf der Berechnung der tatsächlich genutzten Leistung. Der Preis für beanspruchte Ressourcen wird je Einheit festgelegt, zum Beispiel je Zeiteinheit oder Transaktion[13, Kap. 12.3]. Das Risiko für nicht zur Verfügung stehende Ressourcen befindet sich zum größten Teil bei dem Cloud-Anbieter. Neben den in der Produktbeschreibung zugesprochenen Verfügbarkeiten

(s. Abschnitt 3.3.1), kann dem Kunden nur eine Leistung berechnet werden, welche er auch nutzen kann.

2.4 Sicherheitsvorkehrungen

Durch die Nutzung von Cloud-Anbietern werden sensible Daten des Kunden an den Cloud-Anbieter übermittelt. Diese werden zusätzlich vom Cloud-Anbieter geschützt; dafür werden verschiedene Methoden eingesetzt, beispielhaft werden im Folgenden einige erläutert. Des Weiteren werden viele Cloud-Anbieter von unabhängigen Zertifizierungsstellen geprüft.

2.4.1 Georedundanz

Georedundanz ist ein Konzept, welches den Einsatz von mindestens zwei Rechenzentren an zwei voneinander entfernten Standorten vorsieht. Der Ausfall eines Standortes kann durch die anderen Standorte zu jeder Zeit kompensiert werden[36].

Durch die räumliche Distanz der beiden Standorte wird dem Einfluss überregionaler Ereignisse vorgebeugt, zum Beispiel Hochwasser oder Erdbeben.

Georedundant abgesicherte Cloud-Produkte sind insbesondere für Kunden interessant, deren Daten eine maximale Verfügbarkeit benötigen.

2.4.2 RAID-Verbund

Ein RAID-Verbund ist ein Konzept zum logischen Bündeln mehrerer physischer Festplatten mit dem Ziel, die Datensicherheit zu erhöhen und die Zugriffszeiten zu reduzieren[52, S. 110f.]. Ein RAID-Verbund kann verschieden konfiguriert werden, abhängig von der Anzahl der zur Verfügung stehenden Festplatten und dem gewünschten Ziel.

Bei der Nutzung eines RAID 5 wird zum Beispiel ein Kompromiss zu den beiden Zielen Datensicherheit und Zugriffszeit gewählt. Durch die Verteilung der Daten auf mehrere Festplatten können verschiedene Zugriffe zur gleichen Zeit stattfinden. Mindestens eine der genutzten Festplatten enthält Paritätsdaten, welche zur Wiederherstellung einer ausgefallenen Festplatte genutzt werden können.

2.4.3 C5-Zertifikat

Das C5-Zertifikat testiert, dass der vom Bundesamt für Sicherheit in der Informationstechnik (BSI) erstellte Kriterienkatalog C5 (*Cloud Computing Compliance Criteria Catalogue*) erfüllt wird.

Der Kriterienkatalog C5 wurde 2020 überarbeitet und an die aktuellen Anforderungen des Cloud-Computing angepasst. In erster Linie richtet sich der Kriterienkatalog an professionelle Cloud-Anbieter, deren Prüfer und Kunden[15].

Der Kriterienkatalog ist in 17 Anforderungsbereiche unterteilt, zum Beispiel physische Sicherheitsvorkehrungen, Konzepte zum Regelbetrieb, Identitäts- und Berechtigungsmanagement, Umgang mit Ermittlungsanfragen staatlicher Stellen und den Umgang mit Sicherheitsvorfällen[58, S. 16f.].

Das C5-Zertifikat wird von öffentlichen Einrichtungen und vielen Firmen in Deutschland als Grundlage für die Nutzung von Cloud-Produkten verwendet.

2.5 Verwandte Arbeiten

Im folgenden Abschnitt werden Arbeiten ausgewertet und deren Einfluss auf die Erstellung dieser Arbeit betrachtet. Alle Arbeiten behandeln ähnliche Themen in den Bereichen Multimodell oder Cloud-Speicherung.

2.5.1 Anwendung von Techniken des Cloud Computing zur Visualisierung und Bearbeitung von BIM

Im Jahr 2011 hat Chuang die auf dem offenen Markt erhältlichen Programme zur Erstellung von BIM-Modellen untersucht. Insbesondere wird die Möglichkeit des kollaborativen Arbeitens betrachtet. Es wird hervorgehoben, dass keines der untersuchten Programme auf den Zugriff von entfernten Ressourcen ausgelegt ist[18].

Als mögliche Lösung wird der Entwurf einer Cloud-BIM Architektur näher betrachtet. Dieser baut auf dem Konzept von Web-Anwendungen auf, welche für verschiedene Nutzer optimiert sind und auf einer zentralen Service-Schicht aufbauen. Das Zurverfügungstellen und Verarbeiten der Daten für die Service-Schicht wird durch einen *RealityServer*, welcher auf einer SQL-Datenbank aufbaut, umgesetzt.

Das Konzept für ein BIM-Programm in der Cloud zeigt eine mögliche Realisierung von Programmen aus dem Bauingenieurwesen, welche kollaboratives Arbeiten über Teams und Standorte hinweg ermöglicht. Die entworfene Architektur baut auf einer zentralen SQL-Datenbank auf. Das bedeutet, dass statt vollständiger Dateien ausschließlich die explizit benötigten Daten zu dem Nutzer übertragen werden.

Der genutzte Ansatz demonstriert die Möglichkeit, bestehende Programme an eine Nutzung in der Cloud anzupassen. Des Weiteren wird eine SQL-Datenbank zur Speicherung genutzt, welche die Nutzung verschiedener Systeme zur gleichen Zeit ermöglicht.

Das Konzept der Multimodelle bezieht sich auf die Verknüpfung verschiedener Fachmodelle, ein solches ist beispielsweise das Modell eines BIM-Programms. Die Strategie zur Speicherung von MMC erfordert, anders als dieses Konzept, keine Änderungen an den einzelnen Fachmodellen oder deren Programmen. Außerdem wird die Auswahl der benötigten Cloud-Produkte nicht behandelt, sondern nur eine mögliche Nutzung dieser.

2.5.2 Kontextabhängiger Informationsraum für Informationsprozesse im Bauwesen

Multimodelle für komplexe Bauvorhaben sind zunehmend sehr umfangreich, was den Umgang mit diesen schwierig gestaltet. Zum Beispiel wird für ein zu errichtendes Einkaufszentrum ein Multimodell mit mehr als 4.000 Dokumenten erwartet[39, S. 4].

Als eine mögliche Lösung hat Hilbert 2019 das Konzept zur Einführung eines *Multimodell Generator Dienstes* untersucht[39]. Dieser bildet aus umfangreichen Multimodellen sogenannte Teil-Multimodelle, welche ausschließlich die für einen angefragten Arbeitsschritt benötigten Dokumente enthalten.

Der Ansatz zur Bildung von Teil-Multimodellen befasst sich mit der Motivation, jedem Nutzer ausschließlich die Informationen zur Verfügung zu stellen, welche dieser benötigt. Dieses Vorgehen eignet sich für die Verwendung von entfernten Ressourcen, wobei nur die benötigten Informationen übertragen und gegebenenfalls weitere Informationen bei Bedarf angefordert werden.

Der Entwurf eines Multimodell-Generators eignet sich für die gemeinsame Nutzung mit einer verteilten Speicherung, welche sich zum Beispiel durch Cloud-Produkte realisieren lässt.

Wie die Dateien des MMC gespeichert und zur Verfügung gestellt werden wird nicht behandelt. Die Nutzung eines Generators zum Erstellen von Teil-Multimodellen ist eine Möglichkeit bestehende Programme an die Nutzung einer verteilten Speicherung anzupassen. Durch das Extrahieren der benötigten Daten aus dem gesamten Multimodell werden alle benötigten Daten in einem virtuellen Multimodell abgelegt. Dadurch wird ein lokal betriebenes Programm von der verteilten Speicherung entkoppelt, deren effiziente Nutzung darauf ausgelegt ist, Daten nur bei Bedarf (On-Demand) abzurufen.

2.5.3 Alles an seinem (Schreib-)Platz: Abstraktion von Cloud-Speichern mittels Objekt-Speicher

Der Blog-Eintrag aus dem Jahr 2022 erläutert den Entwicklungsprozess bei Dropbox von der ineffizienten Speicherung von Daten bei AWS hin zu einer Abstraktionsebene zur effizienten Nutzung verschiedener Speicherprodukte[32]. Dropbox analysiert die von Nutzern gespeicherten Dateien auf deren Format und Speicherplatz. Im Anschluss an die Verschlüsselung der Daten werden diese entweder in dem eigenen *Magic Pocket block store* oder in dem Cloud-Produkt AWS S3 abgelegt. Durch die Nutzung einer eigenen Datenbank sind alle Metadaten unabhängig des gewählten Speicherplatzes zentral verfügbar. Durch diese Entkopplung kann für jede Datei die beste Methode zur Speicherung ausgewählt werden, ohne dabei die Datensicherheit zu gefährden oder die Erfahrung für den Nutzer negativ zu beeinflussen.

Die von Dropbox gewählte Realisierung ermöglicht die interne Nutzung zweier Speicherprodukte zur Kombination von deren individuellen Vorteilen durch die allgemeine Analyse von Dateien. Die zu entwickelnde Speicherstrategie für einen MMC in der Cloud soll ebenfalls mehrere Cloud-Produkte zur Speicherung der einzelnen Bestandteile eines MMC kombinieren. Im Unterschied zu der gewählten Realisierung von Dropbox stehen bei einem bereits existierenden MMC mehr Informationen über deren Inhalt zur Verfügung. Diese zusätzlichen Informationen können genutzt werden, um das Portfolio an Speicherprodukten für diesen Anwendungsfall zu optimieren.

Fazit Die Arbeiten 2.5.1 und 2.5.2 beschäftigen sich mit dem Umgang von bestehenden MMC und wie dieser optimiert werden kann. Die jeweiligen Szenarien behandeln nicht die Auswahl von geeigneten Cloud-Produkten und haben daher keinen Einfluss auf die zu entwickelnde Speicherstrategie. Im Gegensatz dazu wird in der Arbeit 2.5.3 das Bündeln mehrerer Speicherprodukte unter einer einheitlichen Abstraktionsschicht untersucht. Je nach den Eigenschaften einer zu speichernden Dateien wird eines der zur Verfügung stehenden Speicherprodukte ausgewählt. Dieser Ansatz eignet sich als Orientierung für die zu entwickelnde Speicherstrategie, weil ebenfalls auf Basis von analysierten Dateien ein geeignetes Cloud-Produkt aus einem Produktportfolio ausgewählt werden soll.

3 Analyse von Multimodellen und Cloud-Anbietern

In dem folgenden Kapitel wird der Aufbau von Multimodellen und Multimodell-Containern näher betrachtet. Anschließend wird der Markt für Cloud-Anbieter und deren Cloud-Produkte untersucht, dafür werden zunächst die Bewertungskriterien für diese Arbeit definiert und anschließend bei der Untersuchung eines jeden Cloud-Anbieters angewandt.

3.1 Aufbau eines Multimodell-Containers

Es existieren verschiedene Formen für die Speicherung von Multimodellen, welche einen Austausch dieser ermöglichen. Eine Möglichkeit dieser Speicherung ist die Form eines MMC. Dieser ist in der Norm ISO 21597-1 standardisiert und daher weitestgehend in der Industrie unterstützt. Das in dieser Arbeit entwickelte Konzept basiert auf der Analyse bereits existierender MMC, andere Formen der Speicherung von Multimodellen werden hier nicht betrachtet.

Ein MMC nach der Norm ISO 21597-1 besitzt das Dateiformat ZIP64. Dass es sich bei dem ZIP-Archiv um einen MMC handelt, wird durch die Dateiergung *.icdd* gekennzeichnet. Die Dateistruktur des MMC ist ebenfalls in der ISO Norm definiert. Das Stammverzeichnis beinhaltet die Index-Datei (*Index.rdf*) und drei weitere Verzeichnisse, das sind *Ontology resources*, *Payload documents* und *Payload triples*.

Die Index-Datei ist das Inhaltsverzeichnis eines MMC und liegt in einem XML-basierten RDF-Format vor. Das Inhaltsverzeichnis enthält eine Liste aller in dem MMC enthaltenen Elementarmodelle. Des Weiteren sind zu jedem Elementarmodell Eigenschaften enthalten, zum Beispiel Autor, Dateiformat und Beschreibung. Neben den Elementarmodellen sind noch weitere Daten zum MMC und dessen Aufbau enthalten.

Das Verzeichnis *Ontology resources* des MMC, enthält die beiden Dateien *Container.rdf* und *Linkset.rdf*. Die Datei *Container.rdf* beschreibt die Datenstruktur der Objekte und deren Attribute innerhalb des MMC. Die Datei *Linkset.rdf* beschreibt die Datenstruktur der Linksets und die verschiedenen Varianten zur Verlinkung von Dokumenten und deren Elementen (s. Abschnitt 3.2.3).

Das Verzeichnis *Payload resources* enthält alle im ICDD-Archiv gespeicherten Dokumente. Die Dateistruktur lässt weitere Unterverzeichnisse zu, zum Beispiel um Dokumente thematisch zu strukturieren. Alle Modelle, inklusive ihrer Eigenschaften, sind dabei in der Index-Datei aufgelistet (s. Abschnitt 3.2.1).

Das Verzeichnis *Payload triples* enthält die Linkmodelle des MMC. Die Datenstruktur der Linkmodelle wird durch die Datei *Linkset.rdf* definiert. Unterverzeichnisse sind durch die Da-

teistruktur zugelassen und erlauben die Strukturierung der verschiedenen Linksets.

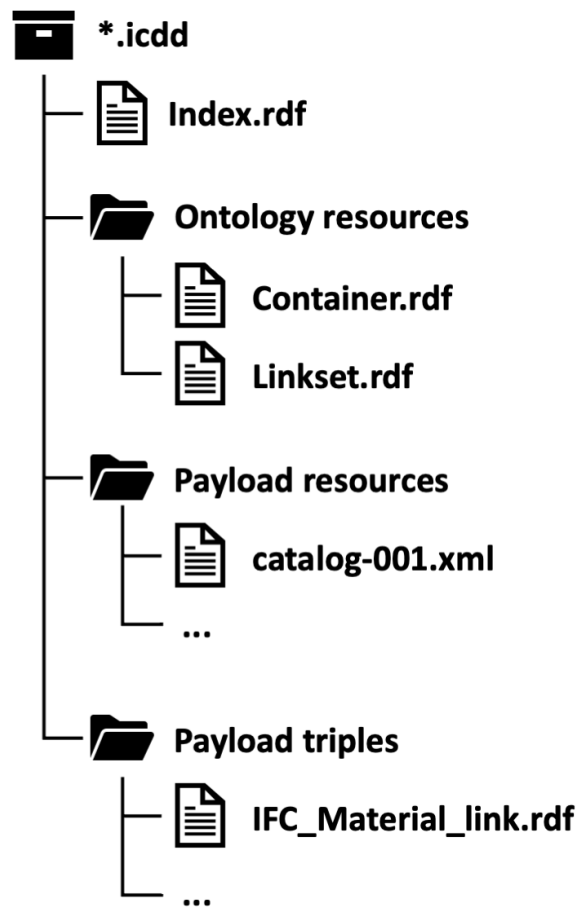


Abbildung 3.1: Dateistruktur eines MMC

3.2 Speicherarten von Modelltypen

Die verschiedenen Bestandteile eines MMC können mit unterschiedlichen Methoden gespeichert werden. Der MMC in Form eines ICDD-Archivs legt alle Bestandteile als Dateien in einem Archiv ab. Für die Speicherung eines Multimodells in der Public Cloud bieten sich gegebenenfalls alternative Methoden zur Speicherung von Informationen an, zum Beispiel in Form einer Datenbank.

Diese Arbeit befasst sich mit der Möglichkeit, die bekannte Dateistruktur eines MMC aufzulösen, um eine Speicherung in Produkten der Public Cloud zu ermöglichen. Der dafür notwendige Transformationsprozess wird hierbei nicht näher betrachtet.

3.2.1 Index-Datei

Die Index-Datei enthält eine Liste aller im MMC enthaltenen Dokumente und Linkmodelle. Sie ist ontologiebasiert und besitzt den Dateityp RDF, welcher auf dem Dateityp XML basiert. In der Index-Datei sind ebenfalls Metadaten zu dem Multimodell in einem Eintrag zusammengefasst (s. Listing 1).

Jedes Dokument ist als eigener Eintrag mit Metadaten in der Index-Datei dargestellt. Der Identifier eines Dokumentes wird als Attribut des Keys *rdf:about* abgelegt. Um welche Art von Eintrag es sich handelt, ist an dem Namespace des Attributs *rdf:type* zu erkennen. In dem Beispiel eines Dokumenten-Eintrags endet dieser mit *Container.rdf#Document* (s. Listing 2 Zeile 2). Des Weiteren sind in dem Eintrag noch Name, Beschreibung, Dateiname und Dateiformat des Dokumentes festgehalten (s. Listing 2 Zeile 1-10). Es kann ebenfalls Einträge geben, welche als externe Dokumente gekennzeichnet sind, diese befinden sich außerhalb des Information Container for Data Drop (ICDD)-Archivs und werden in dieser Arbeit nicht weiter berücksichtigt.

Ein Linkmodell ist ebenfalls jeweils als Eintrag in der Index-Datei dargestellt, enthält aber deutlich weniger Informationen. Der Identifier und der Typ des Eintrags ist identisch zu dem Eintrag eines Dokumentes. Das ist an den Attributen *rdf:about* und *rdf:type* zu erkennen (s. Listing 2 Zeile 12-13). Darüber hinaus enthält der Eintrag nur den Dateinamen des Linkmodells.

Jeder Eintrag in der Index-Datei ist durch einen Identifier gekennzeichnet und damit im MMC eindeutig identifizierbar.

Der Aufbau der Index-Datei ermöglicht es, diese in eine relationale Datenbank zu überführen. Der Identifier kann beispielsweise als primärer Schlüssel verwendet werden. Für die Transformation von XML-basierten Dateien in eine SQL-Datenbank existieren bereits frei zugängliche Tools, zum Beispiel *SQLizer*¹.

Die Abbildung der Index-Datei in einer SQL-Datenbank ermöglicht die Nutzung von SQL-Abfragen. Des Weiteren kann der Betreiber der Datenbank die Zugriffe optimieren, zum Beispiel durch Caching.

```

1 <rdf:RDF
2   xmlns:rdf="https://www.w3.org/1999/02/22-rdf-syntax-ns#"
3   xmlns:owl="https://www.w3.org/2002/07/owl#"
4   xmlns:ct="https://www.iso-icdd.org/draft/Container.rdf#"
5   xmlns:rdfs="https://www.w3.org/2000/01/rdf-schema#"
6   xmlns:xsd="https://www.w3.org/2001/XMLSchema#" >
7 <rdf:Description rdf:about="https://tu-dresden.de/isigg/container#index.rdf">
8   <owl:imports rdf:resource="https://www.iso-icdd.org/draft/Container.rdf"/>
9   <rdf:type rdf:resource="https://www.w3.org/2002/07/owl#Ontology"/>
10  <rdf:type
11    rdf:resource="https://www.iso-icdd.org/draft/Container.rdf#ContainerDescription"/>
12  <ct:publisher rdf:resource="https://tu-dresden.de/isigg/container#{publisherID}"/>
13  <ct:containsDocument rdf:resource="https://tu-dresden.de/isigg/container#{documentID}"/>
14  <ct:containsLinkset rdf:resource="https://tu-dresden.de/isigg/container#{linksetID}"/>
15 </rdf:Description>
16
17 /.../
18
19 </rdf:RDF>

```

Listing 1: Aufbau der Index-Datei

¹<https://sqlizer.io/>

```

1 <rdf:Description rdf:about="https://tu-dresden.de/isigg/container#{documentID}">
2   <rdf:type rdf:resource="https://www.iso-icdd.org/draft/Container.rdf#Document"/>
3   <ct:belongsToContainer rdf:resource="https://tu-dresden.de/isigg/container#index.rdf"/>
4   <ct:creator rdf:resource="https://tu-dresden.de/isigg/container#{creatorID}"/>
5   <ct:name>Station_Material</ct:name>
6   <ct:description>MaterialDataModel</ct:description>
7   <ct:filename>Station_MaterialData.xml</ct:filename>
8   <ct:format>application/x-extension-owl</ct:format>
9   <ct:filetype>xml</ct:filetype>
10 </rdf:Description>
11
12 <rdf:Description rdf:about="https://tu-dresden.de/isigg/container#{linksetID}">
13   <rdf:type rdf:resource="https://www.iso-icdd.org/draft/Container.rdf#Linkset"/>
14   <ct:filename>IFC_Sprinkler_Link.rdf</ct:filename>
15 </rdf:Description>

```

Listing 2: Dokument- und Linkset-Eintrag in der Index-Datei

3.2.2 Fachmodelle

Fachmodelle, auch als Dokumente bezeichnet, sind sehr vielfältig und lassen sich nicht auf Dateitypen oder -größen einschränken. Einige Software-Hersteller nutzen ein proprietäres Dateiformat, weshalb Fachmodelle nachfolgend als geschlossene BlackBox behandelt werden.

Aufgrund der Komplexität und Vielfalt von Fachmodellen wird deren Zweck und fachliche Aussage nicht weitergehend in der Arbeit behandelt.

3.2.3 Linkmodelle

Linkmodelle verknüpfen mindestens zwei Dokumente oder Elemente eines Multimodells miteinander. Ein Element ist ein eindeutig bestimmbarer Teil eines Dokumentes. Die Art der verknüpften Elemente kann dabei variieren, Verlinkungen nach der ISO 21597-1 sind dabei zwischen:

1. einem Element eines Dokuments und mehreren Dokumenten
2. einem Element eines Dokuments mit mehreren Elementen anderer Dokumente
3. mehreren Elementen eines Dokuments mit mehreren Elementen anderer Dokumente

Linkmodelle sind in einem MMC in dem Ordner *Payload triples* in Form von Linksets abgelegt. Linksets, auch Linkmodelle genannt, besitzen das Dateiformat *RDF* und enthalten mehrere, meist logisch zusammenhängende Verknüpfungen. Der Aufbau eines Linkmodells ist in der Datei *Linkset.rdf* im Ordner *Ontology resources* definiert. Ein jedes Linkmodell besteht aus einer Liste von Objekten des Typs *Link*. Ein jeder *Link* enthält eine Liste von Objekten des Typs *LinkElement* (s. Listing 3 Zeile 1-5). Jedes *LinkElement* besitzt einen eindeutigen Identifier sowie den Pfad zu einem Dokument und gegebenenfalls den Identifier eines Elements (s. Listing 3 Zeile 6-15).

Aufgrund des XML-basierten Datenformats und der Datenstruktur mit Identifiern eignen sich Linkmodelle, ähnlich der Index-Datei, diese in Form einer Datenbank zur Verfügung zu stellen. XML-basierte Dateien lassen sich mittels öffentlich zugänglicher Tools in relationale Datenbanken überführen, siehe Abschnitt 3.2.1.

Mit der Speicherung von Linkmodellen in einer Datenbank ist es möglich, direkte SQL-Abfragen auf den Linkmodellen durchzuführen, zum Beispiel die Abfrage aller Verlinkungen eines bereits abgerufenen Dokuments.

In der Tabelle 3.1 ist eine mögliche Abbildung von Linkmodellen durch eine SQL-Datenbank dargestellt. Jede Zeile entspricht einem Linkelement, welches über das Feld *linkelementid* identifiziert werden kann sowie das referenzierte Dokument (*documentid*) und optional einen Identifier eines Elements (*elementid*). Die Linkelemente werden bei dieser Darstellung über das Feld *linkid* einem Link zugeordnet. Das Feld *linkid* ist daher nicht eindeutig und als Schlüssel ungeeignet.

Ein mögliches Szenario zur Nutzung von SQL-Abfragen ist das Auflisten aller Dokumente, welche mit dem bereits geöffneten Dokument *container#{documentA}* verknüpft sind. In Listing 4 wird in der inneren Abfrage jede *linkid* herausgesucht, welche das Dokument *container#{documentA}* enthält. Die äußere Abfrage gibt dann alle Einträge aus, welche die zuvor herausgesuchte *linkid* enthält.

```

1 <rdf:Description rdf:about="http://tu-dresden.de/isigg/linkset#{linkID}">
2   <ls:hasLinkElement rdf:resource="http://tu-dresden.de/isigg/linkset#{elementID_A}" />
3   <ls:hasLinkElement rdf:resource="http://tu-dresden.de/isigg/linkset#{elementID_B}" />
4   <rdf:type rdf:resource="https://wisib.de/ontologie/icddDraft/Linkset.rdf#Link" />
5 </rdf:Description>
6 <rdf:Description rdf:about="http://tu-dresden.de/isigg/linkset#{elementID_A}">
7   <ls:hasIdentifier rdf:resource="http://tu-dresden.de/isigg/linkset#{identifizierA}" />
8   <ls:hasDocument rdf:resource="http://tu-dresden.de/isigg/container#{documentA}" />
9   <rdf:type rdf:resource="https://wisib.de/ontologie/icddDraft/Linkset.rdf#LinkElement" />
10 </rdf:Description>
11 <rdf:Description rdf:about="http://tu-dresden.de/isigg/linkset#{elementID_B}">
12   <ls:hasIdentifier rdf:resource="http://tu-dresden.de/isigg/linkset#{identifizierB}" />
13   <ls:hasDocument rdf:resource="http://tu-dresden.de/isigg/container#{documentB}" />
14   <rdf:type rdf:resource="https://wisib.de/ontologie/icddDraft/Linkset.rdf#LinkElement" />
15 </rdf:Description>

```

Listing 3: Link eines Linkmodells

linkid	linkelementid	documentid	elementid
linkset#{linkID}	linkset#{elementID_A}	container#{documentA}	linkset#{identifizierA}
linkset#{linkID}	linkset#{elementID_B}	container#{documentB}	linkset#{identifizierB}

Tabelle 3.1: Abbildung eines Linkmodells in einer relationalen Datenbank

```

SELECT * FROM linkset WHERE linkid = (
  SELECT linkid FROM linkset WHERE documentID = "container#{documentA}"
)

```

Listing 4: SQL-Abfrage zur Auflistung aller verlinkten Dokumente

3.3 Vergleich von Cloud-Anbietern

In dem folgenden Abschnitt wird ein Auszug der auf dem Markt zur Verfügung stehenden Cloud-Anbieter untersucht, der Fokus liegt dabei auf den Marktführern. Es werden auch kleinere Cloud-Anbieter untersucht, wenn sich diese durch ein außergewöhnliches Produktportfolio sowie geografische oder ökonomische Gegebenheiten abheben. Die Fachmodelle eines MMC werden in dieser Arbeit als Blackbox betrachtet und sind häufig in einem proprietären Dateiformat. Fachmodelle werden deshalb in Tools zur Versionsverwaltung, wie zum Beispiel Git beziehungsweise Github als binäre Dateien behandelt. Die Vorteile von Git können bei binären Dateien nicht angewandt werden, weshalb Git beziehungsweise Plattformen welche Git verwenden nachfolgend nicht näher betrachtet werden.

In dieser Arbeit werden für die Vergleichbarkeit der Preise Standorte um Frankfurt am Main gewählt, anderenfalls der in Deutschland nächstmögliche Standort.

Die Speicherung der für die Bauprozesse notwendigen Modelle bei Public Cloud Anbietern kann zur Verbesserung von Verfügbarkeit, Vereinfachung des Zugriffs für beteiligte Parteien und einer Senkung der Kosten führen.

Zu beachten ist, dass jeweils nur das grundlegende Produkt für einen Vergleich herangezogen wird. Das bedeutet, dass mögliche Zusatzleistungen in dieser Analyse der Cloud-Anbieter nicht näher betrachtet werden.

3.3.1 Kriterien

Um das Potenzial der Public Cloud nutzen zu können, ist es wichtig das am besten geeignete Cloud-Produkt eines Cloud-Anbieters zu verwenden. Für eine Bewertung der Cloud-Produkte sind folgende Faktoren relevant:

Speicherkosten

Jede Datei eines Modells belegt einen gewissen Speicherplatz, so wird als grundlegender Faktor die Größe eines Modells bei der Wahl eines geeigneten Cloud-Produktes berücksichtigt. Je nach Cloud-Produkt des Cloud-Anbieters können die Preise für den verwendeten Speicherplatz stark variieren. Zum Vergleich wird die Einheit Euro je Gigabyte (€/GB) verwendet. Zu beachten ist, dass je nach Cloud-Anbieter und dessen Kostenmodell der verwendete Speicherplatz oder der zur Verfügung gestellte Speicherplatz abgerechnet wird.

So werden zum Beispiel bei einem Produkt mit einem monatlichen Festpreis 1000 GB zur Nutzung zur Verfügung gestellt. Die Kosten sind die gleichen, unabhängig wie viel des zur Verfügung gestellten Speicherplatzes tatsächlich genutzt wird.

Speicherkapazität

Ein Cloud-Produkt zum Speichern von Daten kann eine begrenzte Speicherkapazität besitzen. Das bedeutet, je nach Art des Cloud-Produktes und dessen Kostenmodell können nicht beliebig viele Modelle in einer Instanz eines Cloud-Produktes abgelegt werden. Für den Vergleich der Cloud-Produkte wird die Einheit Gigabyte (GB) verwendet.

Zum Beispiel können bei einem Produkt mit einem monatlichen Festpreis, welches 1000 GB Speicherplatz zur Verfügung stellt, keine weiteren Modelle abgelegt werden, wenn dieser zur Verfügung stehende Speicherplatz bereits vollständig belegt ist.

Transferkosten

Bei jedem Dateizugriff muss die Datei vom Cloud-Anbieter zum zugreifenden Nutzer transferiert werden. Dabei entsteht Netzwerkverkehr für den Cloud-Anbieter, dieser wird in den meisten Fällen berechnet. Je nach Anbindung des Cloud-Anbieters zum nächsten Internet-Anbieter entstehen Kosten für den Cloud-Anbieter. Für den Vergleich, der durch das Volumen des Netzwerkverkehrs anfallenden Kosten, wird die Einheit Euro je Gigabyte (€/GB) verwendet. Bei nahezu allen Cloud-Anbietern wird ausschließlich der ausgehende Netzwerkverkehr berechnet, auch werden Daten deutlich häufiger heruntergeladen als hochgeladen. Daher wird der eingehende Netzwerkverkehr nicht in den weiteren Vergleich mit einbezogen. Je nach Anzahl der Zugriffe auf eine Datei können die Transferkosten einen großen Anteil bei der Berechnung des Gesamtpreises ausmachen.

Transaktionskosten

Bei Nutzung eines SaaS- oder PaaS-Produktes, wie zum Beispiel Datenbanken oder Objekt-Speicher, werden Transaktionen in Rechnung gestellt. Als Transaktion gilt hierbei jede Art von Interaktion mit dem Cloud-Produkt, sowohl der eigentliche Zugriff auf die Dateien, als auch das Abrufen von Verzeichnissen. Für den Vergleich von Cloud-Anbietern wird die Einheit Euro je Zehntausend Transaktionen (€/10.000 Transaktionen) verwendet. Unterscheiden sich die Preise nach Art der Transaktion, werden die Kosten einer Transaktion zum Abrufen von Dateien verwendet.

Skalierbarkeit

Ein Vorteil großer Cloud-Anbieter ist die Möglichkeit der einfachen Skalierung von benötigten Ressourcen. Je nach konkretem Cloud-Produkt lassen sich Ressourcen verschieden einfach und schnell anpassen. Für den Vergleich der Cloud-Produkte wird eine Punkteskala mit dem Wertebereich von 0 bis 2 verwendet. Zwei Punkte stehen für ein besonders einfaches Skalieren der Ressourcen, zum Beispiel in Form eines Cloud-Produktes mit Pay-per-Use Kostenmodell. Ein Punkt bedeutet, dass eine begrenzte Skalierung der Ressourcen möglich ist, zum Beispiel existiert ein Kostenrahmen oder ein Upgrade auf ein größeres Produkt ist ohne weitere Umstände möglich. Null Punkte entsprechen der niedrigsten Bewertung, eine Skalierung der Ressourcen ist ohne einen erhöhten manuellen Eingriff nicht möglich.

Verfügbarkeit

Die Nutzung von Cloud-Anbietern kann zu einer Abhängigkeit der Nutzer von den Cloud-Anbietern führen, weil Anwendungen gegebenenfalls auf die Verfügbarkeit von Daten aus Cloud-Produkten angewiesen sind. Cloud-Anbieter geben die Verfügbarkeit der Cloud-Produkte in Form von Service-Level-Agreements (SLAs) an, also der garantierten Erreichbarkeit über einen Monat oder ein Jahr hinweg. Eine Übersicht der möglichen Ausfallzeiten wird in Tabelle 3.2 abgebildet. Für den Vergleich wird eine Punkteskala mit dem Wertebereich von 0 bis 3 verwendet. Drei Punkte entsprechen einer Verfügbarkeit von mindestens 99,99%. Zwei Punkte entsprechen einer Verfügbarkeit von mindestens 99,9%. Ein Punkt wird ab einer Verfügbarkeit von mindestens 99% vergeben. Null Punkte werden vergeben, wenn die garantierte Verfügbarkeit unterhalb von 99% liegt.

Das Sicherstellen hoher Verfügbarkeiten erfordert einen erhöhten Aufwand an Vorkehrungen, wie zum Beispiel die Replikation von Daten oder eine redundante Infrastruktur von Strom und Netzwerk.

Verfügbarkeit	99%	99,9%	99,99%
Verfügbar	29,7 Tage	29,97 Tage	29,997 Tage
Nicht Verfügbar	7,2 Stunden	43,2 Minuten	4,32 Minuten

Tabelle 3.2: Verfügbarkeit je Monat (30 Tage)

Datensicherheit

Zum Schutz der Daten von Kunden vor einem möglichen Datenverlust, zum Beispiel aufgrund technischer Störungen, nutzen viele Cloud-Anbieter verschiedene Formen von Backups oder stellen redundanten Systeme zur Verfügung. Für den Vergleich der Anbieter wird eine Punkteskala mit einem Wertebereich von 0 bis 2 Punkten verwendet. Null Punkte bedeuten, dass keine weiteren Vorkehrungen zur Sicherung von Daten existieren. Ein Punkt wird vergeben, wenn der Anbieter Vorkehrungen zur Sicherung der Daten am gleichen Standort getroffen hat, zum Beispiel in Form eines RAID-Verbundes oder eines weiteren Servers. Zwei Punkte werden vergeben, wenn der Anbieter eine georedundante Strategie zur Sicherung der Daten verwendet. Das bedeutet, die Daten sind an mindestens einem weiteren geografisch separaten Standort vorgehalten.

Datenschutz

Für viele Firmen sind die Datenhoheit, beziehungsweise die gesetzlichen Rahmenbedingungen, von großer Relevanz. Die Europäische Union hat mit der Datenschutz-Grundverordnung (DSGVO) Rahmenbedingungen für Unternehmen innerhalb der EU geschaffen. Je nach Cloud-Anbieter und dessen eigentlichem Firmensitz kann außerdem das nationale Recht anderer Länder Anwendung finden. Das kann unter Umständen zu Interessenkonflikten mit dem europäischen Datenschutz, beziehungsweise Anforderungen bei öffentlichen Ausschreibungen führen. Für den Vergleich wird der Sitz des Cloud-Anbieters herangezogen, dabei wird ausschließlich unterschieden zwischen dem Sitz innerhalb der Europäischen Union (EU) oder außerhalb (Nicht-EU) der Europäischen Union.

3.3.2 Auswahl von Cloud-Anbietern

Im Folgenden wird die Entwicklungsgeschichte eines jeden Cloud-Anbieters kurz betrachtet sowie eine Auswahl aktueller Produkte, welche sich zur Speicherung von Modellen im Rahmen dieser Arbeit eignen. Anschließend werden alle betrachteten Cloud-Anbieter hinsichtlich der in Abschnitt 3.3.1 definierten Kriterien bewertet und gegenübergestellt.

Amazon Web Services (AWS)

Amazon Web Services, auch AWS bezeichnet, ist der Marktführer im Bereich Cloud-Computing. AWS konnte im dritten Quartal 2022 weltweit einen Marktanteil von 32% erzielen[20]. Mit der Gründung im Jahr 2006 kann AWS auf über 16 Jahre Erfahrung und einen Kundenstamm von mehreren Millionen Nutzern zurückgreifen[70]. Zum aktuellen Produktportfolio zählen über 200 Dienste, welche jedes Servicemodell beinhalten.

Nach eigener Aussage orientiert sich AWS bei den Sicherheitsstandards für die Infrastruktur und Services an denen für militärische Einrichtungen, global tätige Banken und anderen Organisationen, für welche Sicherheit von höchster Bedeutung ist[70]. Zusätzlich erfüllt AWS viele nationale und internationale Standards, zum Beispiel in vielen europäischen Rechenzentren das C5-Zertifikat des BSI[17].

Die Service-Level-Agreement sind je nach Art des Servicemodells in den Zusagen an die Verfügbarkeit abgestuft. Grundlegend garantiert AWS eine Verfügbarkeit von IaaS-Produkten von mindestens 99,99% im Monat. Das beinhaltet zum Beispiel das Produkt *Amazon Elastic Block Store*[7]. Bei dem Produkt *Amazon S3* (Object Storage) gibt AWS eine Verfügbarkeit von 99,9% an[8]. Ebenfalls gibt AWS für Produkte des Servicemodells PaaS beziehungsweise SaaS, zum Beispiel das Datenbank-Produkt *Amazon Aurora* mit einer Verfügbarkeit von mindestens 99,9% an. Zu beachten ist jedoch, dass AWS zu großen Teilen eine hochverfügbare Variante ihres Produktes anbietet, in dem Beispiel von *Amazon Aurora* wird für diese eine Verfügbarkeit von mindestens 99,99% garantiert[6].

Aus dem Produktportfolio werden beispielhaft zwei Produkte näher betrachtet (s. Tabelle 3.3), *S3 Standard* und *Aurora Serverless v2 (Single-AZ)*. Das Produkt *S3 Standard* ist ein Objekt-Speicher, welcher nach dem realen Verbrauch von Ressourcen abgerechnet wird. Berechnet werden zum Beispiel der benötigte Speicherplatz, die Anzahl der Zugriffe und die transferierten Daten. Es eignet sich für das Datei-basierte Speichern von Modellen.

Das Produkt *Aurora Serverless v2 (Single-AZ)* ist eine mögliche Konfiguration des Produktes *Aurora*, welches eine von AWS verwaltete SQL-Datenbank darstellt. Das Kostenmodell von *Aurora* orientiert sich in der serverlosen Konfiguration ebenfalls an der Berechnung der real verbrauchten Ressourcen. Zur Berechnung wird die Einheit Aurora Capacity Unit (ACU) eingeführt, welche den Bedarf benötigter Rechenleistung zum Betrieb der Datenbank darstellt. Für das zur Verfügung stellen der Datenbank werden unabhängig der Nutzung 0,5 ACU benötigt, bei vielen Zugriffen wird die Anzahl der benötigten ACU's in kürzester Zeit dem Bedarf angepasst. Zusätzlich werden der benötigte Speicherplatz, die Zugriffe auf die Datenbank sowie die übertragenen Daten in Rechnung gestellt.

AWS wird in der Europäischen Union durch die Firma *Amazon Web Services EMEA SARL*, auch AWS Europe genannt, vertrieben. AWS und AWS Europe sind Tochterfirmen des US-Unternehmens *Amazon.com*.

	S3 Standard	Aurora Serverless v2 (Single-AZ)
Servicemodell	IaaS	PaaS/SaaS
Kostenmodell	Pay-per-Use	Fixkosten/Pay-per-Use
Fixkosten	-	mind. 0,5 ACU
Speicherkosten	<50TB: 0,0245€/GB	0,119€/GB
Speicherkapazität	-	-
Transferkosten	<10TB: 0,09€/GB	<10TB: 0,09€/GB
Transaktionskosten	0,00043€/1.000 Trans.	0,22€/1 Mio. Trans.
Skalierbarkeit	2	2
Verfügbarkeit	2	2
Datensicherheit	1	1
Sonstiges	-	0,14€/ACU/Stunde

Tabelle 3.3: Auszug - AWS Cloud-Produkte

Microsoft Azure

Microsoft Azure, auch nur Azure genannt, erreichte im dritten Quartal 2022 einen weltweiten Marktanteil von 22% und belegt damit den zweiten Platz der größten Anbieter im Bereich Cloud-Computing[20].

Die Entwicklung als Anbieter im Bereich Cloud-Computing begann Microsoft Ende 2008 als Anbieter für PaaS -Produkte, zu dem Zeitpunkt noch unter dem Namen *Windows Azure*.

Die erste Generation von Produkten, welche ab 2010 der Öffentlichkeit zur Verfügung standen, waren auf den Betrieb von ASP.NET Web-Anwendungen sowie zugrunde liegenden SQL-Datenbanken ausgerichtet. Ein wichtiger Zeitpunkt in der Entwicklung von Microsoft als Cloud-Anbieter war einige Jahre später der Wechsel der Strategie von einer Windows-Plattform hin zu einer Linux-basierten Cloud. Zur gleichen Zeit änderte sich der Name hin zu dem heutigen *Microsoft Azure*. Heute bietet Azure über 600 Services in über 200 Produkten aus den Service-Modellen IaaS, PaaS und SaaS an[40]. Microsoft hebt besonders hervor, dass die Rechenzentren, deren Infrastruktur und die Vorgänge in Azure selbst vollständig durch Microsoft realisiert werden. Dadurch ist es Microsoft möglich, für Azure die höchsten internationalen Standards an Sicherheit und Datenschutz zu gewährleisten[10].

Die Zusage von Microsoft Azure zur Verfügbarkeit sind dem Service-Level-Agreement nach verhältnismäßig hoch. Azure garantiert für deren IaaS-, PaaS- und SaaS-Produkte eine Verfügbarkeit von mindestens 99,9%[62][61]. Für einzelne Produkte, wie zum Beispiel *Microsoft Azure SQL*, sogar mindesten 99,99%. Darüber hinaus existieren zum Teil Optionen, welche die Verfügbarkeit auf mindestens 99,995% steigern[60].

Beispielhaft werden die beiden Produkte *Blob Storage (Hot)* und *Cosmos DB (Serverless)* näher betrachtet (s. Tabelle 3.4). Bei dem Produkt *Blob Storage (Hot)* handelt es sich um einen Objekt-Speicher, ähnlich zu dem Produkt *AWS S3*. Berechnet wird der reale Verbrauch von Ressourcen, das sind zum Beispiel der benötigte Speicherplatz, die Zugriffe und das Übertragen von Daten. Das Produkt *Cosmos DB (Serverless)* ist eine von Azure zur Verfügung gestellte SQL-Datenbank, welche ausschließlich nach dem realen Verbrauch von Ressourcen abgerechnet wird. Im direkten Vergleich zu Produkten wie *AWS Aurora* fallen die Transaktionskosten etwas höher aus, dafür wird die benötigte Rechenleistung zum Betrieb der Datenbank nicht separat berechnet.

Für den Betrieb von Produkten in der Europäischen Union bietet Microsoft Azure eine Besonderheit. Microsoft betreibt zwei souveräne Rechenzentren in Deutschland, welche digital und physisch von dem internationalen Azure-Netzwerk getrennt sind. Dieses Vorgehen stellt sicher, dass keine Daten die Europäische Union verlassen. Damit war Microsoft Azure der erste große Anbieter, welcher die Anforderungen des C5-Zertifikats (s. Abschnitt 2.4.3) erfüllen konnte[31]. Außerdem plant Microsoft Azure ähnliche Angebote für andere Länder und Regionen. Unter diesen Umständen ist es möglich, einzelne Cloud-Produkte der souveränen Rechenzentren gleichgestellt zu denen von europäischen Anbietern zu betrachten. *Microsoft Azure* ist eine Tochterfirma der amerikanischen Firma **Microsoft Corporation**.

	Blob Storage (Hot)	Cosmos DB (Serverless)
Servicemodell	IaaS	PaaS/SaaS
Kostenmodell	Pay-per-Use	Pay-per-Use
Fixkosten	-	-
Speicherkosten	<50TB: 0,0189€/GB	0,241€/GB
Speicherkapazität	-	-
Transferkosten	<10TB: 0,0837€/GB	<10TB: 0,0837€/GB
Transaktionskosten	0,0054€/10.000 Trans.	0,294€/1 Mio. Trans.
Skalierbarkeit	2	2
Verfügbarkeit	2	2
Datensicherheit	1	1
Sonstiges	-	-

Tabelle 3.4: Auszug - Azure Cloud-Produkte

Google Cloud Platform (GCP)

Google Cloud Platform, auch als Google Cloud bezeichnet, ist der dritte große Anbieter im Bereich Cloud-Computing. Google Cloud konnte im dritten Quartal 2022 einen Marktanteil von 9% weltweit erzielen[20]. Die Geschichte von Google Cloud beginnt im Jahr 2008 mit der *Google App Engine*, welche eine Sammlung von Tools der Öffentlichkeit zur Verfügung stellte. Mit diesen war es möglich, webbasierte Anwendungen für die von Google betriebene Infrastruktur zu entwickeln und anschließend zu betreiben[37]. Das angebotene Produktportfolio hat sich heute auf mehr als 100 Produkte erweitert, welche alle Servicemodelle abdecken[54]. Die Sicherheit der Google Cloud sichert alle Aspekte von physischen bis digitalen Eingriffen ab. Alle Rechenzentren sind mit mehreren Sicherheitszonen versehen, welche jeweils mehrere Autorisierungskonzepte besitzen[25]. Google Cloud hebt besonders hervor, dass die eingesetzte Hardware von Anfang bis Ende eines Lebenszyklus in den Sicherheitszonen eines Rechenzentrums von Google verbleibt. Jede Festplatte wird zum Beispiel in mehreren Schritten vernichtet bevor sie entsorgt wird. Des Weiteren ist die betriebene Hardware zu großen Teilen von Google mitentwickelt oder durch eigenen Code angepasst[38]. Des Weiteren erfüllt Google Cloud nationale und internationale Standards zum Datenschutz, beispielsweise das C5-Zertifikat des BSI[16].

Die Zusagen von Google an die Verfügbarkeit in den Service-Level-Agreement unterscheiden sich je Produkt. Bei einzelnen Instanzen von IaaS-Produkten, wie zum Beispiel *Google Cloud Compute Engine* oder *Google Cloud Storage*, gibt Google Cloud eine Verfügbarkeit von mindestens 99,5% oder 99,9% über einen Monat an. Google Cloud verweist auf die Möglichkeit der Replizierung von Instanzen, um eine Verfügbarkeit von 99,9% oder 99,99% zu erreichen[24][23]. Bei PaaS- beziehungsweise SaaS-Produkten, wie zum Beispiel *Google Cloud SQL* gibt Google Cloud eine Verfügbarkeit im Monat von mindestens 99,95% an[22].

Aus dem Produktportfolio werden beispielhaft zwei Produkte näher betrachtet (s. Tabelle 3.5), *Cloud Storage Standard* und *Cloud SQL*. Das Produkt *Cloud Storage Standard* ist ein Objekt-Speicher und wird nach dem realen Verbrauch von Ressourcen abgerechnet. Berechnet werden unter anderem der benötigte Speicherplatz, die Anzahl der Zugriffe und die zum Nutzer transferierten Daten.

Das Produkt *Cloud SQL* ist eine von Google Cloud verwaltete SQL-Datenbank und eignet sich zur Speicherung der Linkmodelle und Index-Dateien eines MMC. Berechnet wird die Datenbank nach den Ressourcen, welche für den Betrieb der Datenbank zur Verfügung gestellt werden. Dafür stellt der Nutzer aus dem Portfolio von IaaS-Produkten eine Konfiguration an Ressourcen zusammen, welche Google Cloud für den Betrieb der Datenbank verwenden darf. Der benötigte Speicherplatz und die zum Kunden übertragenen Daten werden separat in Rechnung gestellt.

Im Europäischen Wirtschaftsraum wird Google Cloud von *Google Ireland Limited* angeboten und vertrieben. Google Cloud ist eine Tochtergesellschaft der *Alphabet Inc.* mit Sitz in Kalifornien, USA.

	Cloud Storage Standard	Cloud SQL
Servicemodell	IaaS	PaaS/SaaS
Kostenmodell	Pay-per-Use	Fixkosten/Pay-per-Use
Fixkosten	-	59,1675€/mtl.
Speicherkosten	0,023€/GB	0,204€/GB
Speicherkapazität	-	-
Transferkosten	<1TB: 0,12€/GB	0,19€/GB
Transaktionskosten	0,004€/10.000 Trans.	-
Skalierbarkeit	2	1
Verfügbarkeit	2	2
Datensicherheit	1	1
Sonstiges	-	vCPU: 1; RAM: 3,75GB

Tabelle 3.5: Auszug - GCP Cloud-Produkte

OVHcloud

OVHcloud, ehemals nur OVH, ist nach eigener Aussage der größte Anbieter Europas im Bereich Cloud-Computing[67]. OVH wurde 1999 in Frankreich gegründet, um einen gemeinschaftlichen Bedarf von Internet-Hosting zu decken. Innerhalb kürzester Zeit ist OVH zu einem der beliebtesten Anbieter für IaaS-Produkte herangewachsen und expandierte bereits 2004 an weitere Standorte in Europa. Dabei setzte OVH immer wieder auf eigene Prozesse und Innovationen, bereits sehr früh kam eine Wasserkühlung der Server zum Einsatz und auch der gesamte Montage-Prozess der Server wurde intern bearbeitet. 2010 begann OVH mit dem Ausbau seines Geschäfts im nordamerikanischen Raum, anschließend folgten auch erste asiatische Regionen. 2019 benannte sich OVH um in OVHcloud und strukturierte sein Produktportfolio neu, um so den individuellen Anforderungen der Kunden besser entsprechen zu können[51]. Heute gelingt OVHcloud ein Spagat zwischen klassischen IaaS-Produkten bis hin zu PaaS - und SaaS-Produkten.

Der Aspekt der Sicherheit ist einer der wichtigsten Aspekte für OVHcloud und verweist auf die ständige Weiterentwicklung der eigenen Sicherheitsvorkehrungen. Auch entwickelt OVHcloud selbst Sicherheitsvorkehrungen, wie zum Beispiel eigene Systeme zur Abwehr von Distributed-Denial-of-Service Angriffen. OVHcloud kann die Erfüllung vieler internationaler Standards vorweisen, Besonders hervorzuheben ist hierbei das *BSI C5* Zertifikat, welches den Einsatz der OVHcloud-Produkte für öffentliche Einrichtungen zulässt[59].

In den Service-Level-Agreements gibt OVHcloud insbesondere für IaaS-Produkte besonders hohe Zusagen an. OVHcloud garantiert für fast alle IaaS-Produkte, zum Beispiel eine *Public Cloud General Purpose Instanz* eine Verfügbarkeit von 99,99%. Bei PaaS - und SaaS-Produkten variieren die Zusagen an die Verfügbarkeit deutlich. Zum Beispiel wird in dem Produkt *Database as a Service - Business* nur eine Verfügbarkeit von 99,9% zugesprochen[11].

OVHcloud hat den Firmensitz in Roubaix, Frankreich und unterliegt daher generell den Gesetzen der Europäischen Union. OVHcloud hebt besonders hervor, dass das Geschäft in anderen Regionen strikt getrennt von dem europäischen Geschäft abgewickelt wird[26].

Open Telekom Cloud

Die Open Telekom Cloud ist ein Angebot der *Deutschen Telekom AG*, welches seit 2016 erstmals Dienste direkt im Internet zur Verfügung stellt[29]. Betreut werden die eigenen Rechenzentren durch die Tochterfirma *T-Systems International GmbH*. Besonders wird der Standort Deutschland beziehungsweise Europa hervorgehoben. Die Rechenzentren befinden sich in Deutschland sowie in den Niederlanden, um ein georedundantes Angebot zu ermöglichen.

Des Weiteren bietet die *Deutsche Telekom AG* auch die *Swiss Open Telekom Cloud* an, ein identisches Angebot von Cloud-Produkten in der Schweiz. Bei der Sicherheit der Open Telekom Cloud wird neben dem Standort Deutschland beziehungsweise Europa, auf die *Twin-Core-Technologie* und die bewusste Auswahl der Standorte hingewiesen. Das bedeutet, dass bei der Wahl eines jeden Standorts das Risiko auf Erdbeben und Überflutungen mit einbezogen wurde. Außerdem besitzt jeder Standort zwei identisch gebaute Rechenzentren, welche sich wenige Kilometer entfernt voneinander befinden[55]. Die Open Telekom Cloud garantiert eine Verfügbarkeit aller Dienste von mindestens 99.95% im Kalendermonat[49]. Ebenfalls erfüllt die Open Telekom Cloud eine Reihe von nationalen und internationalen Standards, unter anderem den C5-Standard des BSI[30]. Das Angebot der Open Telekom Cloud wird rechtlich über die *T-Systems International GmbH* mit Sitz in Deutschland abgewickelt[46].

IONOS

IONOS Cloud, auch IONOS genannt, ist ähnlich wie die *1&1 Aktiengesellschaft* eine Tochterfirma der *United Internet AG*. IONOS Cloud war früher auch als *1&1 IONOS* bekannt und entstammte der 1988 gegründeten Firma *1&1 Internet*, welche sich 2018 mit der Firma *Profit-Bricks* zusammengeschlossen hat. Heute ist IONOS an 15 Standorten in neun Ländern vertreten und betreibt über zehn eigene geo-redundante Rechenzentren. Mit über 8 Millionen Kundenverträgen weltweit und über 90.000 Servern bezeichnet sich IONOS als eines der größten Hosting-Unternehmen Europas[65]. Das Produktportfolio von IONOS umfasst alle Service-Modelle, von dedizierter Hardware über Object-Storage bis hin zu Websites und Email-Services ist alles verfügbar[47]. IONOS garantiert für alle Dienste, zum Beispiel für den *IONOS Cloud S3 Object Storage* eine Verfügbarkeit von mindestens 99,5% im Monat. Für die meisten Produkte, zum Beispiel *IONOS Cloud Compute* oder *IONOS Cloud Database-as-a-Service*, garantiert IONOS eine Verfügbarkeit von mindestens 99,95% oder 99,9%[57]. IONOS Cloud wird rechtlich über die *IONOS SE* mit Sitz in Deutschland abgewickelt[5].

Hetzner

Hetzner zählt nach eigener Aussagen zu den größten Webhostern und Betreibern von Rechenzentren in Europa. Mit der Gründung im Jahre 1997 kann Hetzner auf über 25 Jahre Erfahrung im Bereich des Hostings zurückgreifen. Von Beginn an betreibt Hetzner sämtliche Infrastruktur selbst und ist stets darauf fokussiert, dem Endkunden ein Produkt mit dem besten Preis-Leistungs-Verhältnis zur Verfügung zu stellen[66]. Heute bietet Hetzner nahezu ausschließlich IaaS-Produkte an, welche sich durch ihr besonderes Preis-Leistungs-Verhältnis von der Konkurrenz abheben. Dabei wird auf die eigene Infrastruktur mit mehreren hunderttausend Servern in 29 Rechenzentren an drei Standorten zurückgegriffen[56]. Außerdem betont Hetzner, dass es internationalen Standards für Informationssicherheit entspricht und deren Rechenzentren die neusten Sicherheitsanforderungen erfüllen[71].

Bei der Verfügbarkeit garantiert Hetzner im Rahmen des Service-Level-Agreements eine Verfügbarkeit im Jahresmittel von mindestens 99,9%, wobei Hetzner nicht nach den einzelnen Produkten unterscheidet[2].

Als deutscher Anbieter entspricht Hetzner den Anforderungen an den Datenschutz der Europäischen Union. Besonders wird darauf hingewiesen, dass das Geschäft außerhalb der Europäischen Union vollständig separiert wird.

IBM Cloud

IBM Cloud ist der vierte große Anbieter aus Nordamerika und liegt nach der Statistik aus dem Jahr 2018 mit 1,8% Marktanteil weit hinter AWS, Azure oder Google Cloud[41]. Insbe-

sondere im Finanzsektor spielt IBM Cloud eine große Rolle. Des Weiteren sind viele Produkte im Vergleich zu anderen internationalen Cloud-Provider preiswert[27][21]. IBM Cloud entstammt der Firma *SoftLayer*, welche in dem Jahr 2005 gegründet wurde und besonders als Cloud-Provider für IaaS-Produkte in den USA bekannt war[42]. Im Jahr 2018 wurde das Cloudgeschäft von IBM in IBM Cloud umbenannt und betreibt heute 46 Rechenzentren in 27 Ländern[44]. Bei der Sicherheit der Rechenzentren und Daten verweist IBM Cloud auf die jahrelange Erfahrung sowie auf einen hohen Anteil eigener Infrastruktur, zum Beispiel eigener Netzwerkinfrastruktur zwischen den verschiedenen Standorten, um die Latenz zu verringern und die Zuverlässigkeit zu erhöhen[64]. Ebenfalls setzt IBM Cloud auf eine Reihe nationaler und internationaler Zertifikate, hierbei auch das C5-Zertifikat des Bundesamts für Sicherheit in der Informationstechnik[43].

IBM Cloud garantiert eine Verfügbarkeit über alle Produkte hinweg von mindestens 99,9%, bei vielen Produkten sogar 99,95%, zum Beispiel für *IBM Cloud Object Storage*. Außerdem bietet IBM Cloud für nahezu alle Produkte die Option, eine hochverfügbare Variante des Produktes zu buchen. Auch hier sind die Upgrades im Vergleich zu anderen internationalen Cloud-Anbietern preiswert[45].

IBM Cloud ist eine Marke der Firma International Business Machines Corporation (IBM) mit Sitz im Bundesstaat New York in den USA.

Oracle Cloud Infrastructure (OCI)

Oracle Cloud Infrastructure, kurz auch Oracle Cloud oder OCI genannt, ist der fünfte große Anbieter aus Nordamerika. Oracle Cloud startete sein Angebot im Bereich Cloud-Computing im Jahr 2016 unter dem Namen *Oracle Bare Metal Cloud Services* und beinhaltete ausschließlich IaaS-Produkte. Innerhalb weniger Jahre wurde der Name in *Oracle Cloud Infrastructure* geändert und das Angebot, einschließlich PaaS - und SaaS-Produkte, auf über 70 Produkte erweitert. Bei der Sicherheit setzt Oracle Cloud nach eigenen Angaben auf einen vollständigen *Security-Layer*, welcher von Grund auf alle Daten verschlüsselt. Auch hebt Oracle Cloud hervor, dass der Datenverkehr nach Möglichkeit in viele virtuelle Netzwerke aufgeteilt wird[50]. Oracle Cloud entspricht den internationalen Standards für Informationssicherheit und erfüllt viele regionale Anforderungen an den Datenschutz, beispielsweise das C5-Zertifikat des BSI[19]. Im Service-Level-Agreement spricht Oracle-Cloud den Produkten eine Verfügbarkeit von mindestens 99,9% zu, zum Teil sogar mindestens 99,95%. Dies gilt zum Beispiel für die Produkte *Oracle Cloud Infrastructure - Compute*, *Oracle Cloud Infrastructure - Object Storage* oder *Oracle Cloud Infrastructure - MySQL Database*[53].

Oracle Cloud Infrastructure gehört zu der Firma Oracle Corporation mit Sitz in Texas in den USA.

Alibaba Cloud

Alibaba Cloud wurde 2009 gegründet und ist das führende Unternehmen im Bereich Cloud-Computing in China. Bis heute betreibt Alibaba Cloud international 85 Rechenzentren in 28 Regionen. Alibaba Cloud hebt insbesondere die Flexibilität und Belastbarkeit der eigenen Plattform hervor, welche sich zum *Singles Day* am 11.11. eines jeden Jahres immer wieder unter Beweis stellt[1]. Alibaba Cloud entspricht den internationalen Standards für Informationssicherheit und erfüllt nach Möglichkeit viele regionale Anforderungen, zum Beispiel des C5-Zertifikat es BSI[4]. Das Produktportfolio von Alibaba Cloud beinhaltet alle Servicemodelle. Die Preisgestaltung orientiert sich dabei an den IaaS-Produkten. Die Kosten für ein PaaS-/SaaS-Produkt berechnen sich aus den gewählten IaaS-Produkten, welche beispielsweise einer dedizierten Datenbank zur Verfügung gestellt werden[12]. Alibaba Cloud gibt in den Service-Level-Agreements eine Verfügbarkeit von mindestens 99,9% an, wie zum Bei-

spiel bei dem PaaS-/SaaS-Produkt *AnalyticDB for PostgreSQL*[9]. Bei IaaS-Produkten, wie zum Beispiel *Object Storage Service*, garantiert Alibaba Cloud eine Verfügbarkeit von mindestens 99,99%. Zusätzlich gibt es zu den Standard-Produkten teilweise noch Optionen, welche eine Verfügbarkeit von mindestens 99,995% angeben[3]. Alibaba Cloud ist eine Tochterfirma der *Alibaba Group Holding Limited* mit Sitz in Hangzhou, China. Vertrieben wird Alibaba Cloud international durch die Niederlassung in Singapur.

Anbieter	IaaS	PaaS/SaaS	Verfügbarkeit	Datensicherheit	Datenschutz
Amazon Web Services	✓	✓	2-3	1-2	Nicht-EU
Microsoft Azure	✓	✓	2-3	1-2	Nicht-EU
Google Cloud Platform	✓	✓	1-3	1-2	Nicht-EU
OVHcloud	✓	✓	2-3	1-2	EU
Open Telekom Cloud	✓	✓	2	2-3	EU
Ionos	✓	✓	1-2	1-2	EU
Hetzner	✓	✗	2	1	EU
IBM Cloud	✓	✓	2-3	1-2	Nicht-EU
Oracle Cloud	✓	✓	2	1-2	Nicht-EU
Alibaba Cloud	✓	✓	2-3	1-2	Nicht-EU

Tabelle 3.6: Übersicht Cloud-Anbieter

3.4 On-Site Speicherung

Die On-Site Speicherung bezeichnet das Speichern von Daten am Standort eines Unternehmens, die gesamte Verantwortung für den Betrieb liegt bei dem Unternehmen.(s. Abbildung 2.1). Innerhalb dieser Arbeit wird die On-Site Speicherung nicht weitergehend betrachtet.

Für die zuverlässige Speicherung von Daten muss eine geeignete Infrastruktur, Server und ein Konzept zur Instandhaltung vorliegen.

Der zuverlässige Betrieb von Servern setzt eine geeignete Infrastruktur voraus, das beinhaltet insbesondere eine gegen Ausfälle geschützte Stromversorgung, zum Beispiel durch eine USV. Den Schutz vor Zugriffen durch unbefugte Personen, sowohl physisch als auch virtuell, und eine geeignete Netzwerkanbindung.

Zur Erhöhung der Datensicherheit und dem Vermeiden von Ausfällen sollten kritische Komponenten der Infrastruktur, sowie zentrale Server, redundant ausgelegt werden. Die Server müssen den Anforderungen entsprechend konfiguriert werden, das beinhaltet beispielsweise die Nutzung eines geeigneten Redundant Array of Independent Disks (RAID)-Verbundes. Des Weiteren ist eine Backup-Strategie zu einem entfernten Standort empfehlenswert, um dem Einfluss lokaler Ereignisse vorzubeugen.

Außerdem müssen alle Bestandteile durch das Unternehmen gewartet und betreut werden, das setzt geeignete Personal voraus. Je nach Art der gespeicherten Daten und der zur Verfügung gestellten Dienste ist eine automatisierte Überwachung, einschließlich einer Rufbereitschaft empfehlenswert.

4 Konzept

Das folgende Kapitel beschreibt zwei konkurrierende Konzepte. Jedes hat das Ziel, für einen MMC die am besten geeignete Kombination aus Cloud-Produkten zusammenzustellen. Die zur Verfügung stehenden Cloud-Produkte werden durch vom Benutzer gegebene Anforderungen auf deren Eignung geprüft. Für einen gegebenen MMC werden die Bestandteile und deren Zusammenhänge jeweils analysiert und kategorisiert.

Der MMC wird durch die Speicherung in der Public Cloud inhaltlich nicht verändert.

4.1 Kollaboratives Arbeiten

Kollaboratives Arbeiten bedeutet, dass mehrere Personen von verschiedenen Endgeräten zeitgleich auf sich überschneidende Teile eines MMC zugreifen können.

Die Dateien müssen dazu auf Ressourcen abgelegt werden, welche allen Personen einen Zugriff ermöglichen. Dafür bieten sich insbesondere dezentrale Ressourcen in Form von Cloud-Produkten der Public Cloud an. Bei umfangreichen Projekten mit vielen Nutzern ist es notwendig, die geteilten Ressourcen zu administrieren. Dies erfolgt zum Beispiel durch Berechtigungskonzepte und eine Versionsverwaltung. Dadurch wird sichergestellt, dass Änderungen an Teilen des MMC protokolliert und Schreibkonflikte vermieden werden.

Das Erstellen eines Berechtigungskonzeptes sowie die Untersuchung von Versionsverwaltungen sind nicht Teil dieser Arbeit. Bei Bedarf müssen diese separat untersucht und realisiert werden. Des Weiteren existieren am Markt bereits Konzepte, welche auf die Untersuchungen dieser Arbeit angewandt werden können.

4.2 Speichermethoden

Für Speicherung der einzelnen Bestandteile eines MMC existieren mehrere Möglichkeiten. Eine Möglichkeit ist die Speicherung von Modellen als Dateien in der Public Cloud. Darüber hinaus besteht die Option, bestimmte Modelle eines MMC in eine Datenbank zu überführen.

Datei Jedes Modell kann direkt als Datei in der Public Cloud gespeichert werden. Das Analysieren des Inhaltes einer Datei ist nicht notwendig, die Datei kann als Blackbox betrachtet werden. Zum Speichern von Dateien in der Public Cloud werden IaaS-Produkte verwendet, meistens in Form von Block- oder Objekt-Speichern. Die meisten Kostenmodelle von IaaS-Produkten sind schlank gehalten und orientieren sich an dem benötigten Speicherplatz und gegebenenfalls anfallenden Kosten für genutzten Netzwerkverkehr. Bei jedem Zugriff ist es

notwendig, die gesamte Datei aus der Public Cloud zum Nutzer zu übertragen. Bei den meisten Cloud-Anbietern sind die Kosten für den Netzwerkverkehr deutlich höher als der eigentliche Speicherplatz. Je nach Anzahl der Zugriffe auf ein Modell entsteht ein Großteil der Kosten durch das Übertragen der Dateien. Werden Änderungen an einer Datei vorgenommen, muss die gesamte Datei erneut hochgeladen werden.

Datenbank Ausgewählte Modelle können dem Nutzer mittels einer Datenbank in der Public Cloud zur Verfügung gestellt werden. Datenbanken zählen zu den PaaS-/SaaS-Produkten, das sind beispielsweise die Datenbanken MySQL- und PostgreSQL. Für das Speichern von Modellen in einer Datenbank ist eine Transformation dieser notwendig. Das setzt voraus, dass der Inhalt der Dateien interpretiert werden kann und der Aufbau für eine Transformation geeignet ist.

Durch das Überführen von Elementarmodellen in eine Datenbank kann der benötigte Speicherplatz von der ursprünglichen Größe der Datei abweichen. Für die folgenden Speicherstrategien wird vereinfacht davon ausgegangen, dass der benötigte Speicherplatz gleich bleibt.

In dieser Arbeit beschränkt sich die Betrachtung von Datenbank-Produkten auf die Linkmodelle (s. Abschnitt 3.2.3) und die Index-Datei (s. Abschnitt 3.2.1). Beide liegen als RDF-Dateien vor, welche auf dem XML-Format basieren und für die Transformation in eine relationale Datenbank geeignet sind. Nachfolgend werden alle Betrachtungen auf die Nutzung einer SQL-Datenbank bezogen.

Gegebenenfalls erfüllen auch einige Fachmodelle die Voraussetzungen für eine Speicherung in Datenbanken. Damit diese weiterhin verwendet werden können, muss jede Anwendung, welche dieses Modell verwendet, mit der Speicherung in einer Datenbank kompatibel sein. Dafür ist es erforderlich, sehr viele Tools verschiedener Hersteller näher zu untersuchen, was im Rahmen dieser Arbeit nicht möglich ist. Des Weiteren könnte das Konzept der losen Kopplung in Multimodellen eingeschränkt werden, weil einzelne Fachmodelle verändert werden.

Die Kostenmodelle von Datenbanken sind im Vergleich zu denen von IaaS-Produkten komplexer. Diese sind häufig Kombinationen aus den Kostenmodellen Festpreis und Pay-per-Use. Bei vielen Datenbank-Produkten ist es notwendig, minimale Ressourcen dauerhaft zu buchen, damit die Grundfunktionen der Datenbank zur Verfügung stehen. Die dauerhaft notwendigen minimalen Ressourcen können als Festpreis abgebildet werden. Darüber hinaus benötigte Ressourcen werden bei Bedarf in Anspruch genommen. Die zusätzlich benötigte Ressource unterscheidet sich je nach Cloud-Anbieter und lässt sich im Vorhinein nicht genauer abschätzen. Die reale Performance eines jeden Cloud-Produktes muss separat evaluiert werden und wird nicht in dieser Arbeit erfasst. Für den Vergleich von Cloud-Produkten werden die minimale Konfiguration oder die offiziellen minimalen Anforderungen an die Ressourcen einer Datenbank betrachtet.

Zusätzlich können die Anzahl der Anfragen an die Datenbank, der Speicherplatz und der Netzwerkverkehr berechnet werden. Der benötigte Netzwerkverkehr kann durch die Verwendung optimierter Abfragen deutlich reduziert werden. Anstelle des gesamten Modells werden nur die benötigten Daten übertragen.

Bei einer großen Anzahl von Zugriffen wird der höhere Festpreis durch die deutlich geringeren Kosten des Netzwerkverkehrs kompensiert (s. Tabelle 4.2). Des Weiteren können alle Linkmodelle gemeinsam in einer Datenbank gespeichert werden, was den Festpreis je Linkmodell senkt.

Im folgenden Beispiel werden die entstehenden Kosten durch die verschiedenen Speichermethoden eines einzelnen Linkmodells einander gegenübergestellt. Es werden verein-

facht zwei Zugriffsszenarien angenommen, eines mit 60.000 Zugriffen je Monat und eines mit 6.000.000 Zugriffen je Monat. Die zur Berechnung verwendeten Produktpreise sind in Tabelle 4.1 zusammengefasst.

Das Linkmodell besitzt eine Dateigröße von 150 kB und enthält 25 Links. Eine durchschnittliche Anfrage benötigt 5 Links. Das bedeutet vereinfacht, dass bei Verwendung einer Datenbank im Durchschnitt nur 20% des Netzwerkverkehrs entsteht.

Aus der Gegenüberstellung in Tabelle 4.2 geht hervor, dass bei geringen Zugriffen auf das Linkmodell die Speicherung als Datei deutlich kostengünstiger ist. Die hohen Fixkosten werden erst bei einer größeren Anzahl an Zugriffen im Monat kompensiert. Wird auf den MMC kaum zugegriffen, zum Beispiel wenn dieser archiviert wird, ist eine Speicherung des Linkmodells als Datei besser geeignet. Wird der MMC durch mehrere Parteien verwendet, ist eine Speicherung in Form einer Datenbank kostengünstiger.

Ein MMC enthält meistens mehrere Linkmodelle sowie eine Index-Datei. Werden mehrere Modelle in einer gemeinsamen Datenbank abgebildet, sinkt die Relevanz der Fixkosten gegenüber den Transferkosten. Des Weiteren können durch den Cloud-Anbieter die Zugriffe auf die Datenbank optimiert werden, womit schneller auf die Daten zugegriffen werden kann.

Für die anschließenden Speicherstrategien wird, wenn verfügbar, eine Datenbank zur Abbildung von der Index-Datei und Linkmodellen verwendet.

Cloud-Produkt	AWS S3	AWS Aurora
Fixkosten (Monat)	0,00€	50,40€
Speicherkosten (GB)	0,0245€	0,119€
Transaktionskosten (10.000)	0,0043€	0,0022€
Transferkosten (GB)	0,09€	0,09€

Tabelle 4.1: Produktkosten AWS S3 und AWS Aurora (Stand: November 2022)

Cloud-Produkt Zugriffe/Monat	AWS S3 (Datei)		AWS Aurora (Datenbank)	
	60.000	6.000.000	60.000	6.000.000
Fixkosten	0,00€	0,00€	50,40€	50,40€
Speicherkosten	0,0004ct	0,0004ct	0,0018ct	0,0018ct
Transferkosten	0,81€	81,00€	0,162€	16,20€
Transaktionskosten	0,03€	2,58€	0,01€	1,32€
Gesamtkosten	0,84€	83,58€	50,57€	67,92€

Tabelle 4.2: Gegenüberstellung Kosten je Speichermethode

4.3 Strategien zur kostenoptimierten Cloudspeicherung von Multimodellen

In den folgenden beiden Abschnitten wird jeweils eine Speicherstrategie erläutert. Jede verwendet eine Abstraktion der von Cloud-Anbietern genutzten Kostenmodelle. Auf dieser Abstraktion aufbauend werden die voraussichtlich anfallenden Kosten bei der Speicherung eines MMC in der Public Cloud ermittelt und optimiert.

4.3.1 Modellorientierte Kostenoptimierung

Die Modellorientierte Kostenoptimierung verfolgt die Strategie, für jedes Modell eines gegebenen MMC das ideale Cloud-Produkt aus dem Portfolio aller Cloud-Anbieter herauszusuchen. Die von Cloud-Anbietern genutzten Kostenmodelle werden auf eine Abstraktion des Kostenmodells Pay-per-Use abgebildet. Im Vergleich zu dem direkten Ansatz, den gesamten MMC auf einer Netzwerkfestplatte abzulegen, können bei der individuellen Betrachtung eines jeden Modells neben IaaS-Produkten auch PaaS-/SaaS-Produkte genutzt werden. Die Auswahl der Cloud-Produkte wird stark durch das zu erwartende Nutzungsverhalten sowie mögliche Vorschriften bezüglich der Datensicherheit beeinflusst. Es ist notwendig, vorher einige Parameter festzulegen, welche nicht aus den Modellen des MMC abgeleitet werden können.

Eingabeparameter

Zugriffsfrequenz Aus dem Aufbau und den Referenzen innerhalb eines MMC lassen sich Schlussfolgerungen zu der Verteilung der Zugriffe auf die Elementarmodelle ziehen. Das bedeutet, ein Fachmodell wird sehr wahrscheinlich abgerufen, wenn es besonders häufig in Linkmodellen referenziert wird.

Darüber hinaus ist eine Schätzung der Zugriffe auf ein MMC notwendig, welche als Eingabeparameter realisiert wird. Mit der Angabe der erwarteten Zugriffe auf einen MMC und der Zugriffswahrscheinlichkeit auf die Fachmodelle lassen sich Annahmen für einen durchschnittlichen Zugriff treffen. Durch diese Informationen ist es möglich, die anfallenden Kosten durch Zugriffe abzuschätzen. Diese setzen sich zum Beispiel aus Kosten für Abfragen und den entstehenden Netzwerkverkehr zusammen.

Skalierbarkeit Jedes Cloud-Produkt kann unterschiedlich gut skalieren. Je nach geplanter Nutzung des MMC muss berücksichtigt werden, dass zeitnah gegebenenfalls weiterer Speicherplatz benötigt wird. Die Anforderungen an die Skalierbarkeit werden auf einer Skala mit den Werten 0-2 angegeben. Sind großteils Lesezugriffe auf den MMC zu erwarten, kann ein Wert von 0 übergeben werden. Soll beispielsweise ein MMC archiviert werden, wird zeitnah kein zusätzlicher Speicherplatz benötigt. Bei einem Wert von 0 existiert keine Anforderung an die Skalierbarkeit und alle Cloud-Produkte werden ohne Einschränkungen in die Auswahl mit einbezogen. Ein Wert von 1 bedeutet, dass zeitnah mehr Speicherplatz benötigt wird. Zum Beispiel sind regelmäßige Änderungen an bestehenden Fachmodellen zu erwarten, vereinzelt können auch neue hinzukommen. Es werden nahezu alle Cloud-Produkte in die Auswahl einbezogen. Es kann notwendig sein, zu gegebener Zeit manuell zu einem Cloud-Produkt mit einer höheren Kapazität zu wechseln. Produkte mit einer fixen Kapazität werden bei einem Wert von 1 mit 60% ihrer eigentlichen Kapazität in der Auswahl berücksichtigt. Durch diese Einschränkung steht auf absehbare Zeit ausreichend Speicherplatz für Änderungen am MMC zur Verfügung. Die Grenze von 60% ermöglicht es auch Cloud-Produkte mit einer fixen Kapazität in die Auswahl einzubeziehen. Zudem ermöglicht die gewählte Grenze die Nutzung einer Warnstufe bei 80%. Dadurch sind 20% für ein Wachstum des MMC berücksichtigt. Der Wert 2 entspricht der höchsten Anforderung an die Skalierbarkeit von Cloud-Produkten. Eine automatische Skalierung des Speicherplatzes bei Bedarf wird unterstützt. Cloud-Produkte mit fester Kapazität werden bei dieser Anforderung nicht in die Auswahl einbezogen. Diese Anforderung kann gewählt werden, wenn der benötigte Speicherplatz des MMC zeitnah um ein Vielfaches wachsen wird. Zum Beispiel ist kurz nach der Erstellung eines MMC ein starkes Wachstum des benötigten Speicherplatzes zu erwarten, weil weitere Fachmodelle hinzukommen.

Verfügbarkeit Die Verfügbarkeit eines Cloud-Produktes gibt an, wie lange ein Cloud-Produkt im Monat höchstens nicht verfügbar sein darf. Jeder Cloud-Anbieter gibt in seinem Service-Level-Agreement eine Verfügbarkeit der Cloud-Produkte an. Die Werte der meisten Cloud-Anbieter liegen bei über 99%, dennoch sind die Unterschiede der möglichen Ausfallzeiten groß (s. Abschnitt 3.3.1).

Die Angaben können je nach Produktkategorie stark variieren, bei einem IaaS-Produkt wird häufig eine höhere Verfügbarkeit angeboten als bei einem PaaS-/SaaS-Produkt. Bei PaaS-/SaaS-Produkten übernimmt der Cloud-Anbieter zusätzlich die Verantwortung für den Betrieb der Software.

Je nach Nutzung des MMC ist eine besonders hohe Verfügbarkeit für den ungestörten Geschäftsablauf relevant. Diese Anforderung wird mittels einer Skala mit den Werten von 0 bis 3 realisiert. Ein Wert von 3 entspricht der höchsten Anforderung an die Verfügbarkeit von mindestens 99,99%. Die Daten werden bei zeitkritischen Aufgaben benötigt, zum Beispiel wenn Sicherheitskräfte diese benötigen. Der Wert 2 entspricht einer Verfügbarkeit von mindestens 99,9%. Diese Anforderung kann zum Beispiel für besonders große MMC genutzt werden. Der MMC wird von vielen Nutzern oder Automationen benötigt, und kleinere Ausfälle führen zu einer nicht vertretbaren Störung im Arbeitsablauf. Der Wert 1 entspricht einer Verfügbarkeit von mindestens 99%. Diese Anforderung eignet sich zum Beispiel für Multimodelle, an welchen aktiv gearbeitet wird, temporäre Störungen im Arbeitsablauf aber vertretbar sind. Ein Wert von 0 entspricht einer Verfügbarkeit von weniger als 99%. Es finden wenige Zugriffe auf das Multimodell statt, beziehungsweise sind diese nicht zeitkritisch, zum Beispiel bei Backups oder der Archivierung von Multimodellen.

Wird keine Anforderung zur Verfügbarkeit angegeben, so wird als Standard der Wert 0 angenommen. Es werden alle Cloud-Produkte berücksichtigt. Für die meisten Anwendungsfälle ist eine Verfügbarkeit von mindestens 99% bereits ausreichend. Die Anforderung an die Verfügbarkeit sollte mit Bedacht gewählt werden, da die Kosten bei einer besonders hohen Verfügbarkeit erheblich steigen.

Datenredundanz Die meisten Cloud-Anbieter treffen Vorkehrungen zur Sicherung der Kundendaten, zum Beispiel in Form von eines RAID-Verbundes oder der Spiegelung von Servern. Bei geschäftskritischen Daten ist eine separate und georedundante Strategie für Backups zu empfehlen, unabhängig von den bereits getroffenen Vorkehrungen eines Cloud-Anbieters.

Die Anforderung an die Datenredundanz eines Cloud-Produktes wird auf einer Skala von 0 bis 2 dargestellt. Ein Wert von 0 bedeutet, dass keine redundante Speicherung benötigt wird. Unter Umständen sind alle Daten auf einer einzigen Festplatte ohne weitere Sicherheitsvorkehrungen gespeichert. Von dieser Art der Speicherung sollte bei der Arbeit mit Geschäftsdaten dringend abgesehen werden! Der Verzicht von der redundanten Speicherung bietet sich lediglich für einzelne Entwicklerdaten an. Ein Wert von 1 bedeutet, dass am Standort des Rechenzentrums mindestens eine Kopie der Daten vorgehalten wird oder Sicherheitsvorkehrungen zum Erhalt der Daten getroffen wurden. Das kann zum Beispiel durch einen RAID-Verbund im gleichen Server oder eine Spiegelung auf einen anderen Server gewährleistet werden. Diese Art der Speicherung ist für die meisten Geschäftsdaten ausreichend. Der unwiederbringliche Verlust von Daten ist sehr unwahrscheinlich, kann aber nicht ausgeschlossen werden. Ein Wert von 2 wird verwendet, wenn alle Daten an mindestens zwei geografisch voneinander getrennten Standorten georedundant gespeichert werden. Der Verlust von Daten ist nahezu ausgeschlossen. Diese Art der Speicherung empfiehlt sich für geschäftskritische Daten. Zu beachten ist hierbei, dass deutlich höhere Kosten entstehen und gegebenenfalls eine manuelle Backup-Strategie kostengünstiger sein kann.

Wird keine Anforderung an die Datenredundanz angegeben, werden alle Cloud-Produkte in die Auswahl einbezogen.

Datenhoheit Die Datenhoheit ist ein geopolitischer, aber auch ein rechtlicher Aspekt. Je nach Hauptsitz des Cloud-Anbieters kann dessen nationales Recht angewandt werden. Sind unter den Modellen besonders sensible Daten, kann es sinnvoll sein, den geografischen und rechtlichen Raum bei der Suche geeigneter Cloud-Anbieter einzuschränken. Dabei werden beispielhaft die geopolitischen Räume EU und Nicht-EU verwendet. Der Eingabewert EU entspricht dem Gebiet der Europäischen Union. Innerhalb der Europäischen Union existiert eine nahezu einheitliche Regelung bezüglich des Datenschutzes. Der Wert Nicht-EU steht dabei für alle weiteren geopolitischen Räume wie Nordamerika oder China. Wird keine Anforderung bezüglich der Datenhoheit angegeben, werden alle Cloud-Anbieter in den Auswahlprozess mit einbezogen.

I. Einlesen von Cloud-Produkten

Die erste Phase, auch als *Initialisierungsphase* bezeichnet, beinhaltet das Einlesen der zur Verfügung stehenden Cloud-Produkte aus einer Konfigurationsdatei. Diese sollte auf einer relationalen Struktur aufbauen, zum Beispiel auf XML oder JSON basierende Dateiformate. Jeder Eintrag beschreibt genau ein Cloud-Produkt und enthält grundlegende Informationen über den Cloud-Anbieter sowie Details zu dem Cloud-Produkt. Dazu zählen zum Beispiel das Servicemodell, die Speicherkosten, Transferkosten, Transaktionskosten, sowie eine Bewertung der Skalierbarkeit, Verfügbarkeit und Redundanz (s. Listing 5). Das Kostenmodell des Cloud-Produktes wird dabei auf einen Preis je Einheit abgebildet. Bei einem Cloud-Produkt mit Festpreis wird dieser zum Beispiel auf Euro je GB umgerechnet.

```

1  [
2    "aws-s3-fra": {
3      "provider": "AWS",
4      "serviceType": "IaaS",
5      "location": "NON-EU",
6      "capacityCost": 0.0245,
7      "transferCost": 0.09,
8      "transactionCost": 0.00043,
9      "scalability": 2,
10     "availability": 2,
11     "redundancy": 1
12   },
13   "ionos-s3": {
14     "provider": "Ionos",
15     "serviceType": "IaaS",
16     "location": "EU",
17     "capacityCost": 0.015,
18     "transferCost": 0.03,
19     "transactionCost": 0.0,
20     "scalability": 2,
21     "availability": 1,
22     "redundancy": 2
23   }
24 ]

```

Listing 5: Ausschnitt der products.json für die Modellorientierte Kostenoptimierung

II. Analyse (Multi-)Modell

Die zweite Phase behandelt das Einlesen des Multimodells sowie die Bewertung eines jeden Elementarmodells. Das Multimodell liegt als MMC vor. Weitere Formen von Multimodellen sind ebenfalls möglich, werden aber in dieser Arbeit nicht näher betrachtet. Aufgrund der standardisierten Struktur des MMC steht von Anfang an eine Index-Datei zur Verfügung, welche das gezielte Heraussuchen der Elementarmodelle erlaubt.

Modellanalyse Jedes Elementarmodell wird einzeln bewertet. Je nach Typ des Elementarmodells kann gegebenenfalls der Inhalt weitergehend analysiert werden (s. Abbildung 4.1). Zunächst wird für jedes Elementarmodell die Dateigröße erfasst. Die Dateigröße eines Modells stellt den größten Kostenfaktor bei der Nutzung von Cloud-Produkten dar. Sie hat einen direkten Einfluss auf den benötigten Speicherplatz und den entstehenden Netzwerkverkehr. Im Anschluss wird geprüft, welchen Dateityp das zu untersuchende Elementarmodell besitzt. Sollte dieser ein Extensible Markup Language (XML)-basiertes Format sein, kann eine Datei gegebenenfalls in einer Datenbank abgespeichert werden. Handelt es sich zum Beispiel um ein Linkmodell im RDF-Format, kann dieses weitergehend analysiert werden. Handelt es sich um ein Fachmodell, wird der Inhalt nicht analysiert und das Dokument wird als Blackbox behandelt.

Nachdem alle Fachmodelle eingelesen wurden, werden die Linkmodelle weitergehend analysiert. Die enthaltenen Referenzen ermöglichen es, die Verteilung der Zugriffe über die Elementarmodelle abzuschätzen. Je häufiger ein Dokument in einem Linkmodell referenziert wird, desto höher ist die Wahrscheinlichkeit, dass auf dieses zugegriffen wird. Je mehr Referenzen ein Linkmodell enthält, desto höher ist die Wahrscheinlichkeit, dass auf dieses zugegriffen wird.

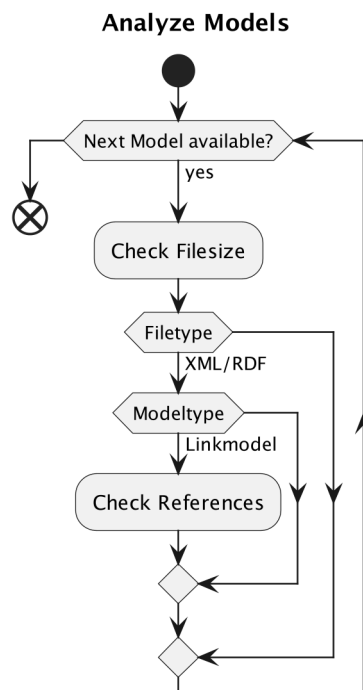


Abbildung 4.1: Aktivitätsdiagramm Analyse Modell

III. Auswahl Cloud-Produkte

Im Anschluss an das Einlesen der Cloud-Produkte und der Analyse aller Elementarmodelle folgt die Auswahl der Cloud-Produkte. Jedem einzelnen Elementarmodell wird das am besten geeignete Cloud-Produkt zugeordnet. Je nach Typ des Elementarmodells werden verschiedene Kategorien von Cloud-Produkten bevorzugt.

Datenhoheit Als Erstes wird die Anforderung an die Datenhoheit geprüft. Dafür wird der Cloud-Anbieter eines Cloud-Produktes betrachtet. Wurde als Eingabeparameter ein Wert für die Datenhoheit angegeben, werden alle Cloud-Anbieter, welche nicht dieser Eingabe entsprechen, einschließlich deren Cloud-Produkte, aussortiert. Diese Cloud-Produkte stehen im weiteren Auswahlprozess nicht mehr zur Verfügung. Tritt der Fall auf, dass anschließend keine Cloud-Produkte mehr zur Verfügung stehen, wird der Auswahlprozess abgebrochen. In diesem Fall sollte das Produktportfolio um Cloud-Produkte, welche dieser Anforderung entsprechen, erweitert werden. Wurde kein Eingabeparameter zur Datenhoheit angegeben, werden alle Cloud-Produkte im weiteren Auswahlprozess mit einbezogen.

Verfügbarkeit Als Nächstes werden die Cloud-Produkte bezüglich der Anforderung an die Verfügbarkeit geprüft. Es werden alle Cloud-Produkte aussortiert, welche einen geringeren Wert auf der Skala zur Verfügbarkeit besitzen. Alle Cloud-Produkte, welche die Anforderung zur Verfügbarkeit erfüllen oder übererfüllen, bleiben im Produktportfolio erhalten und werden im weiteren Auswahlprozess berücksichtigt. Tritt der Fall auf, dass keine Cloud-Produkte diesem Kriterium, beziehungsweise der Kombination mit den vorhergehenden Kriterien entsprechen, wird der Auswahlprozess abgebrochen. In diesem Fall sollten die Gründe für diese Anforderung geprüft werden. Anderenfalls muss das Produktportfolio angepasst beziehungsweise erweitert werden.

Datenredundanz Anschließend werden im Auswahlprozess die Cloud-Produkte bezüglich der Anforderung an die Datenredundanz geprüft. Es werden die Cloud-Produkte von dem Auswahlprozess ausgeschlossen, welche einen geringeren Wert auf der Skala besitzen. Cloud-Produkte, welche die Anforderung erfüllen oder übererfüllen, werden im weiteren Auswahlprozess berücksichtigt. Sollte nach der Prüfung der Anforderung zur Datenredundanz beziehungsweise der Kombination mit den vorhergehenden Anforderungen kein Cloud-Produkt mehr zur Verfügung stehen, wird der Auswahlprozess abgebrochen. In diesem Fall sollten die Gründe für die Anforderung an die Datenredundanz geprüft werden. Eine manuelle Realisierung einer Backupstrategie kann gegebenenfalls die Anforderung an die Datenredundanz ersetzen oder ergänzen. Anderenfalls muss das Produktportfolio um Cloud-Produkte erweitert werden, welche die gestellten Anforderungen erfüllen.

Skalierbarkeit Bei der Anforderung der Skalierbarkeit wird die zu erwartende Nutzung des MMC abgebildet, insbesondere ob sich dessen Komplexität zeitnah verändert (s. Seite 29). Es werden alle Cloud-Produkte aus dem Auswahlprozess entfernt, welche nicht die Anforderung die Skalierbarkeit erfüllen oder übererfüllen. Tritt der Fall ein, dass kein Cloud-Produkt dieser Anforderung beziehungsweise der Kombination mit den vorhergehenden Anforderungen entspricht, wird der Auswahlprozess abgebrochen. In diesem Fall sollte die Anforderung an die Skalierbarkeit geprüft werden. Alternativ muss das Produktportfolio hinsichtlich der benötigten Kriterien erweitert werden.

Im Anschluss an die Prüfung der Anforderungen wird jedes Elementarmodell einzeln zu dessen Bewertung herangezogen.

PaaS/SaaS-Produkte Für Linkmodelle oder die Index-Datei wird eine Speicherung in Form einer Datenbank bevorzugt (s. Abschnitt 4.2). Durch geeignete Abfragen auf diese Modelle kann die Nutzung der Ressourcen optimiert werden, zum Beispiel muss nicht die gesamte Datei eines Linkmodells heruntergeladen werden. Mit einem geeigneten Datenbankbefehl können zum Beispiel alle Einträge des Linkmodells zu einem Dokument abgerufen werden. Falls kein geeignetes PaaS/SaaS-Produkt zur Verfügung steht, wird alternativ auf IaaS-Produkte zurückgegriffen.

Zugriffsfrequenz Die Zugriffsfrequenz ist ebenfalls ein Eingabeparameter, bezieht sich jedoch nicht auf die Eigenschaft von Cloud-Produkten. Vielmehr ist die Zugriffsfrequenz eine Annahme für ein zu erwartendes Nutzungsverhalten des MMC. Zu beachten ist, dass sich die Anzahl der Zugriffe auf die Elementarmodelle eines MMC unterscheiden. Je nach Zweck des Zugriffs werden nur einige Elementarmodelle abgerufen. Die Zugriffsszenarien sind stark von dem jeweiligen MMC abhängig und lassen sich nicht verallgemeinern. Hinweise auf die Verteilung der Zugriffe lassen sich aus den Linkmodellen ableiten. Je mehr Referenzen für ein Elementarmodell existieren, desto höher ist die Wahrscheinlichkeit eines Zugriffs.

Nicht jedes Dokument ist in den Linkmodellen referenziert und nicht jeder Zugriff nutzt Linkmodelle und deren referenzierte Dokumente. Daher ist die grundlegende Annahme, dass ein Fachmodell bei mindestens jedem zweiten Zugriff auf das MMC abgerufen wird (s. Beispiel 1).

Beispiel 1: Es sollen die Gesamtkosten für die benötigten Ressourcen eines Bauabschnitts berechnet werden. Die Kosten sind je Ressource in der Liste enthalten. Alle notwendigen Informationen sind in dem Dokument enthalten, ein weiterer Zugriff auf ein Linkmodell oder andere Dokumente ist nicht notwendig.

Beispiel 2: Für die Lieferanten soll für die nächste Lieferung eine Liste der benötigten Ressourcen zusammengestellt werden. Die dafür benötigten Informationen befinden sich in dem Zeitplan der Projektphase und der Liste der Ressourcen. Der Zeitplan enthält die demnächst benötigten Ressourcen. Diese sind durch einen Link mit der entsprechenden Zeile in der Liste der Ressourcen verknüpft. Die verknüpfte Zeile enthält die Informationen zu einem Lieferanten, welche das Sortieren nach dem Lieferanten ermöglicht. In diesem Zugriffsszenario wird auf zwei Dokumente und ein Linkmodell zugegriffen.

Das Zusammenführen der beiden Annahmen führt zu folgender Berechnung für die Wahrscheinlichkeit p , dass ein Elementarmodell bei einem Zugriff aufgerufen wird:

$$P(m) = 0,5 + 0,5 \cdot \frac{ref(m)}{\max(ref(M))}, m \in M$$

$ref(m)$...Anzahl der Referenzen auf das Elementarmodell m

M ...Menge aller Elementarmodelle eines Multimodells

Darüber hinaus ist davon auszugehen, dass auf die Index-Datei deutlich häufiger zugegriffen wird als auf die Elementarmodelle. Für jeden Zugriff auf ein Elementarmodell ist zuvor ein Zugriff auf die Index-Datei notwendig. Sollten mehrere Elementarmodelle benötigt werden, finden nacheinander mehrere Zugriffe auf der Index-Datei statt. Diese können nicht gebündelt werden, weil benötigte Daten erst zum Zeitpunkt des konkreten Bedarfs abgerufen werden. Dies ist notwendig, damit zu jedem Zeitpunkt auf den aktuellen Daten gearbeitet wird und nur benötigte Ressourcen beansprucht werden. Die Anzahl der Zugriffe auf die

Index-Datei entspricht der Anzahl der durchschnittlich abgerufenen Elementarmodelle.

$$Z = \frac{\sum_{m=1}^{|M|} P(m)}{|M|} \cdot |M| = \sum_{m=1}^{|M|} P(m)$$

Z ...Anzahl der Zugriffe auf die Index-Datei

M ...Menge aller Elementarmodelle eines MMC

$P(m)$...Zugriffswahrscheinlichkeit des Elementarmodells m

Mit der Anzahl der zu erwartenden Zugriffe je Elementarmodell, lassen sich die entstehenden Kosten abschätzen. Diese Berechnung wird für jedes Elementarmodell durchgeführt, ausgewählt wird das Cloud-Produkt, welches die geringsten zu erwartenden Kosten enthält. Die Kosten berechnen sich folgendermaßen:

$$\begin{aligned} c(m) = & \text{size}(m) \cdot \text{capacityCost} \\ & + P(m) \cdot \text{Zugriffe} \cdot \text{transactionCost} \\ & + P(m) \cdot \text{Zugriffe} \cdot \text{transferCost} \cdot \text{size}(m), m \in M \\ c(m) \dots & \text{Zu erwartende Kosten durch das Elementarmodell } m \end{aligned}$$

Für Linkmodelle und die Index-Datei wird die Formel bei der Berechnung des genutzten Netzwerkverkehrs abgewandelt. Ein durchschnittlicher Zugriff lädt nicht die gesamte Datei herunter, sondern nur explizite Einträge. Daher wird die Berechnung der Kosten ergänzt:

$$\begin{aligned} \text{transfersize}(m) &= \frac{\text{size}(m)}{\text{ref}(m)} \\ c(m) = & \text{size}(m) \cdot \text{capacityCost} \\ & + P(m) \cdot \text{Zugriffe} \cdot \text{transactionCost} \\ & + P(m) \cdot \text{Zugriffe} \cdot \text{transferCost} \cdot \text{transfersize}(m), m \in M \\ \text{ref}(m) \dots & \text{Anzahl der Referenzen auf das Elementarmodell } m \\ c(m) \dots & \text{Zu erwartende Kosten durch das Elementarmodell } m \end{aligned}$$

Sobald für jedes Modell ein geeignetes Cloud-Produkt ausgewählt wurde, wird das Ergebnis in Form einer Datei ausgegeben. Darin ist jedes ausgewählte Cloud-Produkt mit den jeweiligen Elementarmodellen enthalten. Zum Beispiel wurde das Produkt *productA* zum abspeichern des Modells *modelA*, zu einem erwarteten Preis von 2,3172€ je Monat ausgewählt (s. Listing 6).

```

1  [
2    {
3      "productid": "productA",
4      "costs": 2.3172,
5      "models": [
6        {
7          "modelid": "modelA",
8          "filename": "modelA.ifc"
9        }
10   ]
11 }
12 }
```

Listing 6: Auszug der Ausgabe-Datei

4.3.2 Matrix-basierte Kostenoptimierung

Die Matrix-basierte Kostenoptimierung verfolgt den gleichen Ansatz wie die Modellorientierte Kostenoptimierung. Für einen gegebenen MMC wird unter Berücksichtigung der individuellen Eigenschaften eines jeden Elementarmodells das ideale Produktportfolio zusammengestellt. Die Besonderheit der Matrix-basierten Kostenoptimierung findet sich in der Berücksichtigung der verschiedenen Kostenmodelle von Cloud-Produkten wieder. Das beinhaltet insbesondere Kostenmodelle mit Festpreisen. Die Festpreise werden nicht auf einen Preis je GB umgerechnet, stattdessen werden die Fixkosten direkt mit den erwarteten Kosten der Pay-per-Use-Alternativen verglichen.

Eingabeparameter

Die Eingabeparameter der Matrix-basierten Kostenoptimierung entsprechen denen der Modellorientierten Kostenoptimierung. Eine Übersicht ist in Tabelle 4.3 zusammengestellt, für weitere Details siehe Seite 29.

Eingabeparameter	Wertebereich	Beschreibung
Zugriffsfrequenz	Zugriffe / Monate	Zu erwartenden Zugriffe je Monat
Skalierbarkeit	0-2	Flexibilität des Speicherplatzes
Verfügbarkeit	0-3	Verfügbarkeit über den Monat
Datenredundanz	0-2	Vorkehrungen gegen Datenverlust
Datenhoheit	EU / Nicht-EU	Sitz des CloudProviders innerhalb der EU

Tabelle 4.3: Übersicht der Eingabeparameter

I. Analyse Cloud-Produkte

Die erste Phase ist ähnlich der Modellorientierten Kostenoptimierung. Es wird eine Konfigurationsdatei mit allen zur Verfügung stehenden Cloud-Produkten eingelesen. Zusätzlich zur Modellorientierten Kostenoptimierung erhält jedes Cloud-Produkt zwei weitere Eigenschaften. Diese ermöglichen die Angabe eines Festpreises in Euro und der Speicherkapazität eines Produktes in GigaByte (s. Listing 7). Eigenschaften, die nicht auf das Cloud-Produkt zutreffen, werden mit einer Null angegeben.

Ein Cloud-Produkt, welches das Kostenmodell Pay-per-Use verwendet, besitzt 0€ an Fixkosten und 0GB an Speicherkapazität (s. Listing 7 Zeile 2-14). Besitzt ein Cloud-Produkt das Kostenmodell Festpreis, werden die Kosten für den Speicherplatz mit 0€/GB angegeben (s. Listing 7 Zeile 15-27).

Eine Kombination dieser beiden Beispiele zur Abbildung weiterer Kostenmodelle ist möglich. Zum Beispiel muss für den Betrieb einer Datenbank ein monatlicher Grundbetrag bezahlt werden, der benötigte Speicher wird zusätzlich nach Verbrauch berechnet. Es kann weitere Kombinationen von verwendeten Kostenmodellen geben, die sich nicht präzise abbilden lassen. Zum Beispiel, wenn für ein Produkt ein geringes Speicherkontingent kostenlos bereitgestellt und nur der Speicherplatz darüber hinaus berechnet wird.

```

1  [
2    "ionos-s3": {
3      "provider": "Ionos",
4      "serviceType": "IaaS",
5      "location": "EU",
6      "basePrice": 0.0,
7      "capacity": 0,
8      "capacityCost": 0.015,
9      "transferCostOut": 0.03,
10     "transactionCost": 0.0,
11     "scalability": 2,
12     "availability": 1,
13     "redundancy": 2
14   },
15   "hetzner-bx11": {
16     "provider": "Hetzner",
17     "serviceType": "IaaS",
18     "location": "EU",
19     "basePrice": 3.81,
20     "capacity": 1000,
21     "capacityCost": 0.0,
22     "transferCostOut": 0.0,
23     "transactionCost": 0.0,
24     "scalability": 1,
25     "availability": 2,
26     "redundancy": 1
27   }
28 ]

```

Listing 7: Ausschnitt der Datei products.json

II. Analyse Modelle

Die zweite Phase übernimmt die Analyse des MMC sowie der enthaltenen Elementarmodelle. Der Ablauf ist identisch zu der zweiten Phase der Modellorientierten Kostenoptimierung, siehe Seite 32.

III. Zuordnung von Cloud-Produkten und Modellen

Die dritte Phase beinhaltet das Zuordnen von Elementarmodellen und Cloud-Produkten. Dies lässt sich durch zwei disjunkten Mengen darstellen. Die erste Menge M entspricht allen Elementarmodellen ($m_1 \dots m_x$), einschließlich der Index-Datei.

$$m_1 \dots m_x \in M, x \in \mathbb{N}$$

Die Zweite Menge P beinhaltet alle Cloud-Produkte ($p_1 \dots p_y$), welche aus der Produktkonfiguration eingelesen wurden

$$p_1 \dots p_y \in P, y \in \mathbb{N}$$

Zunächst wird die Menge aller Cloud-Produkte P bezüglich der Anforderungen aus den Eingabeparametern geprüft. Das geschieht identisch zu der Modellorientierten Kostenoptimierung. Cloud-Produkte, welche nicht den Anforderungen entsprechen, werden aus der Menge P entfernt.

Anschließend wird eine Verknüpfung von jedem Elementarmodell hin zu jedem kompatiblen Cloud-Produkt erstellt. Dabei werden nach Möglichkeit für XML-basierte Dateien PaaS-/SaaS-Produkte eingesetzt, ähnlich der Modellbasierten Kostenoptimierung.

Die Verknüpfung von Modellen und Cloud-Produkten lässt sich als Graph mit ungewichteten Kanten darstellen (s. Abbildung 4.2).

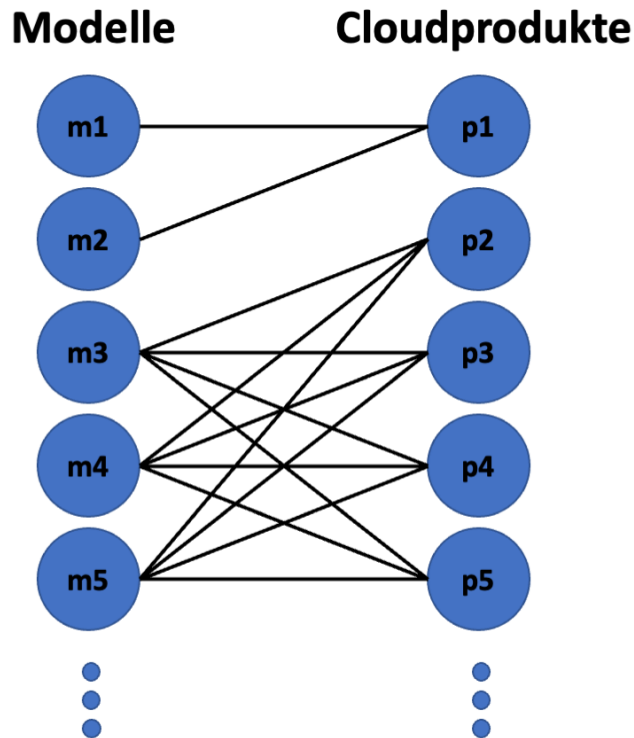


Abbildung 4.2: Relation Elementarmodelle und zulässige Cloud-Produkte

Die vereinfachte Darstellung der Verknüpfung von Elementarmodellen und Cloud-Produkten durch einen Graphen mit ungewichteten Kanten ist nur bedingt geeignet. Jede Kante, die eine Verknüpfung eines Elementarmodells mit einem Cloud-Produkt darstellt, verursacht gegebenenfalls Kosten, insbesondere bei Cloud-Produkten mit dem Kostenmodell Pay-per-Use. Deshalb erhält jede Verknüpfung ein Gewicht, welches den zu erwartenden Kosten für diese Kombination aus Elementarmodell und Cloud-Produkt entspricht.

Matrixdarstellung Ein Graph mit gewichteten Kanten wird in dieser Darstellung schnell unübersichtlich, deshalb wird dieser in eine Matrix-Darstellung überführt. Die Spalten der Matrix repräsentieren alle Cloud-Produkte, die Zeilen sind alle Elementarmodelle des MMC. Ist ein Elementarmodell mit einem Cloud-Produkt nicht kompatibel, dann bleibt die Zelle leer. Sind Elementarmodell und Cloud-Produkt kompatibel zueinander, werden die durch das Elementarmodell entstehenden Kosten berechnet und in der Zelle eingetragen (s. Abbildung 4.3).

Zugriffsfrequenz Um die entstehenden Kosten eines Elementarmodells bei der Nutzung eines Cloud-Produktes zu berechnen, wird die Anzahl der Zugriffe benötigt. Die Annahmen für die Zugriffsverteilung innerhalb des MMC entsprechen denen aus der Modellorientierten Kostenoptimierung (s. Seite 34).

Die Kosten, welche direkt durch ein Elementarmodell entstehen, werden identisch zu der Modellorientierten Kostenoptimierung berechnet (s. Seite 35).

Für Linkmodelle und die Index-Datei wird die Formel für die Berechnung des entstehenden Netzwerkverkehrs, identisch zu der Modellorientierten Kostenoptimierung abgewandelt (s. Seite 35).

	p1	p2	p3	p4	p5	...
m1	0,52€	-	-	-	-	...
m2	0,25€	-	-	-	-	...
m3	-	0,00€	0,00€	0,00€	0,22€	...
m4	-	0,00€	0,00€	0,00€	0,17€	...
m5	-	0,00€	0,00€	0,00€	7,15€	...
...	...	...	...	...	...	...

Abbildung 4.3: Matrixdarstellung - Modelle zu Cloud-Produkten

Die Matrix enthält die durch die Elementarmodelle entstehenden Kosten. Zusätzlich verursachen einige Cloud-Produkte monatliche Fixkosten. Damit diese mit einbezogen werden, wird eine zusätzliche Zeile oberhalb der Modelle hinzugefügt. Diese Zeile enthält die monatlichen Fixkosten eines Cloud-Produktes (s. Abbildung 4.4).

IV. Auswahl Festpreis-Produkten

Die vierte Phase behandelt den Vergleich von Cloud-Produkten mit Festpreisen mit denen mit dem Kostenmodell Pay-per-Use.

Dafür werden im ersten Schritt alle Cloud-Produkte, welche einen Festpreis besitzen, zum Vergleich herangezogen.

Kapazitätsprüfung Besitzt ein Cloud-Produkt eine begrenzte Kapazität, muss geprüft werden, ob alle geeigneten Elementarmodelle in einer Instanz von dem Cloud-Produkt untergebracht werden können. Die zu verwendende Kapazität kann von der tatsächlichen Kapazität abweichen, das hängt von der Anforderung an die Skalierbarkeit ab. Wurde ein Wert von 1 bei der Anforderung an die Skalierbarkeit gewählt, werden nur 60% der verfügbaren Kapazität verwendet (s. Seite 29). Nachfolgend wird die Prüfung des Cloud-Produktes mit der zu verwendenden Kapazität durchgeführt. Zunächst werden alle mit dem Cloud-Produkt abbildbaren Elementarmodelle abgerufen und deren Dateigröße aufsummiert. Ist in dem Cloud-Produkt ausreichend Speicherplatz verfügbar, ist kein weiteres Eingreifen notwendig. Ist nicht genügend Speicherplatz verfügbar, kann nur ein Teil der Elementarmodelle abgebildet werden. Die abzubildenden Elementarmodelle werden anhand der Dateigröße absteigend ausgewählt. Es werden so viele große Elementarmodelle wie möglich in dem Cloud-Produkt abgebildet.

	p1	p2	p3	p4	p5	...
Fix	0,00€	3,81€	12,79€	24,75€	0,00€	...
m1	0,52€	-	-	-	-	...
m2	0,25€	-	-	-	-	...
m3	-	0,00€	0,00€	0,00€	0,22€	...
m4	-	0,00€	0,00€	0,00€	0,17€	...
m5	-	0,00€	0,00€	0,00€	7,15€	...
...	...	...	...	...	...	...

Abbildung 4.4: Matrixdarstellung - Modelle zu Cloud-Produkten mit Festpreis

Kostenvergleichsrechnung Bei der Kostenvergleichsrechnung wird geprüft, ob sich bei der Nutzung des Cloud-Produktes mit Festpreis ein Kostenvorteil gegenüber der Nutzung verschiedener Cloud-Produkte mit Pay-per-Use Kostenmodell ergibt. Dafür werden alle Elementarmodelle, welche in dem Cloud-Produkt mit Festpreis abgebildet werden können, zum Vergleich herangezogen.

Zum Beispiel wird das Produkt *p2* mit einem Festpreis ausgewählt. Die abbildbaren Modelle *m3*, *m4*, *m5* werden zum Vergleich herangezogen (s. Abbildung 4.5).



	p1	p2	p3	p4	p5	...
Fix	0,00€	3,81€	12,79€	24,75€	0,00€	...
m1	0,52€	-	-	-	-	...
m2	0,25€	-	-	-	-	...
m3	-	0,00€	0,00€	0,00€	0,22€	...
m4	-	0,00€	0,00€	0,00€	0,17€	...
m5	-	0,00€	0,00€	0,00€	7,15€	...
...	...	...	...	...	...	...

Abbildung 4.5: Matrixdarstellung - Modelle, welche sich mit dem Cloud-Produkt p2 abbilden lassen

Für jedes Elementarmodell wird das alternative Cloud-Produkt mit dem günstigsten zu erwartenden Preis ausgewählt. Kann ein Modell in keinem anderen Cloud-Produkt abgebildet werden, wird das Cloud-Produkt sofort ausgewählt. Die Vergleichssumme berechnet sich aus den Einzelkosten der alternativen Cloud-Produkte und wird den Kosten des Cloud-Produktes mit Festpreis gegenübergestellt. Die Ersparnis wird folgendermaßen berechnet:

$$E(P_1) = \sum_{n=1}^{|P_1|} cost(m_n) - (basePrice(P_1) + \sum_{n=1}^{|P_1|} cost(P_1, m_n))$$

$E(P_1)$...Kostensersparnis bei Verwendung des Festpreis-Produkts P_1

P_1 ...Menge der mit dem Cloud-Produkt P_1 abbildbaren Modelle

$cost(m_n)$...Geringste Kosten für Modell m_n bei Verwendung eines Pay-per-Use-Produkts

$cost(P_1, m_n)$...Kosten für Modell m_n bei Verwendung des Cloud-Produkts P_1

Zum Beispiel wird dies für die Modelle $m3$, $m4$ und $m5$ durchgeführt. Anschließend wird die Ersparnis im Vergleich zu den Cloud-Produkten mit Festpreis ausgerechnet. Das Produkt $p2$ besitzt die höchste Ersparnis und wird ausgewählt (s. Abbildung 4.6).

Ist die Ersparnis negativ, weil die alternativen Cloud-Produkte kostengünstiger sind, wird das Cloud-Produkt nicht weiter in Betracht gezogen. Die Spalte mit dem Cloud-Produkt in der Matrix wird eliminiert (s. Abbildung 4.7).

Wurden alle Cloud-Produkte mit Festpreis geprüft, wird das Cloud-Produkt mit der höchsten Ersparnis ausgewählt. Alle in dem gewählten Cloud-Produkt abgebildeten Elementarmodelle werden entfernt, die entsprechende Zeile in der Matrix wird eliminiert (s. Abbildung 4.7).

Der gesamte Vorgang zur Prüfung von Cloud-Produkten mit Festpreis wird wiederholt, bis alle Cloud-Produkte mit Festpreis eliminiert wurden.

		+3,73€	-5,25€	-17,21€			
		p1	p2	p3	p4	p5	...
	Fix	0,00€	3,81€	12,79€	24,75€	0,00€	...
	m1	0,52€	-	-	-	-	...
	m2	0,25€	-	-	-	-	...
➡	m3	-	0,00€	0,00€	0,00€	0,22€	...
➡	m4	-	0,00€	0,00€	0,00€	0,17€	...
➡	m5	-	0,00€	0,00€	0,00€	7,15€	...
	...	...	...	...	...	...	...

Abbildung 4.6: Matrixdarstellung - Festpreis-Produkte nach Kostenvergleichsrechnung

V. Auswahl von Pay-per-Use-Produkten

In der fünften Phase der Matrix-basierten Kostenoptimierung werden Elementarmodelle behandelt, welche in der vorherigen Phase nicht abgebildet werden konnten. Sie werden dem günstigsten Cloud-Produkt mit Pay-per-Use Kostenmodell zugeordnet. Dafür wird in der Matrix nacheinander jedes Elementarmodell selektiert. Es wird die Zelle ausgewählt, welche die geringsten Kosten enthält (s. Abbildung 4.7). Die Spalten der Cloud-Produkte mit Festpreis sind in der vierten Phase bereits eliminiert worden. Das Elementarmodell wird dem gewählten Cloud-Produkt zugeordnet und dessen Zeile anschließend in der Matrix eliminiert.

Dieser Vorgang wird wiederholt, bis alle Elementarmodelle einem Cloud-Produkt zugewiesen sind.

Die ausgewählten Cloud-Produkte werden als Liste in einer JSON-Datei ausgegeben. Jedem Cloud-Produkt werden weitere Informationen hinzugefügt, beispielsweise die abgebildeten Elementarmodelle, Gesamtkosten, genutzter Speicherplatz und Transferkosten (s. Listing 8).

	p1				p5	...
Fix	0,00€				0,00€	...
m1	0,52€				-	...
m2	0,25€				-	...
...	...				...	...

Abbildung 4.7: Matrixdarstellung - Auswahl von Pay-per-Use-Produkten

```

1  [
2    {
3      "productid": "productA",
4      "costs": 2.3172,
5      "usedspace": 0.002454,
6      "capacitycosts": 0.0,
7      "transactioncosts": 0.0,
8      "transfercosts": 2.3152,
9      "models": [
10     {
11       "modelid": "modelA",
12       "filename": "modelA.ifc"
13     }
14   ]
15 }
16 }
```

Listing 8: Auszug der Ausgabe-Datei

4.4 Fazit

Sowohl die Modellorientierte Kostenoptimierung als auch die Matrix-basierte Kostenoptimierung erlauben es, für einen gegebenen MMC geeignete Cloud-Produkte auszuwählen. Bei beiden Konzepten wird über Eingabeparameter die Eingabe grundlegender Anforderungen an Cloud-Produkte ermöglicht. Die Modellorientierte Kostenoptimierung nutzt die Normierung auf Preise je GB beziehungsweise Zugriffe, um den Vergleich zwischen mehreren Cloud-Produkten vereinfacht zu betrachten. Das führt unter Umständen zu einer verzerrten Darstellung der Kosten in der ausgegebenen Lösung, weil einige Cloud-Produkte nicht je GB erworben werden können, sondern ein Budget von Speicherkapazität pauschal erworben werden muss. Die Matrix-basierte Kostenoptimierung kann sich insbesondere in zwei Punkten gegenüber der Modellorientierten Kostenoptimierung behaupten, in der detailgetreuen Abbildung von Cloud-Produkten mit verschiedenen Kostenmodellen und in der Ausgabe präziserer Prognosen von Kosten. Durch das Einbeziehen monatlicher Festpreise sowie einer möglichen maximalen Kapazität je Cloud-Produkt können Kostenmodelle wie Festpreise exakt angegeben werden. Auch Variationen von Kostenmodellen lassen sich besser annähern. Die Zusammenstellung von Cloud-Produkten für einen gegebenen MMC findet mit geringeren Abweichungen zu den realen Kosten statt, so wird das Cloud-Produkt mit den geringsten Kosten zuverlässiger ausgewählt.

Aus diesen Gründen ist die Matrixbasierte Kostenoptimierung deutlich präziser und näher an der Realität. Daher wird den folgenden Kapiteln das Konzept der Matrix-basierten Kostenoptimierung zugrunde gelegt.

	Vorteile	Nachteile
Modellorientierte Kostenoptimierung	einfacher und direkter Vergleich von Cloud-Produkten	Nur Kostenmodell Pay-per-Use abbildbar
Matrix-basierte Kostenoptimierung	Kostenmodell Festpreis und Speicherkapazität abbildbar	Aufwendiges Auswahlverfahren

Tabelle 4.4: Gegenüberstellung Modellorientierte und Matrix-basierte Kostenoptimierung

5 Implementierung

Das folgende Kapitel beschreibt den Prozess zur Implementierung der Matrix-basierten Kostenoptimierung. Dabei werden insbesondere die einzelnen Phasen des Konzeptes sowie Entscheidungen zum Code-Design und die Lösung auftretender Probleme betrachtet. Eine Übersicht zum Ablauf der Implementierung findet sich in Abbildung 5.1.

Wahl der Programmiersprache Die Implementierung der Matrix-basierten Kostenoptimierung wird in der Programmiersprache Java realisiert. Java zählt mit einem Marktanteil von 17,03% weltweit zu den beliebtesten Programmiersprachen im Jahr 2022 und ist in der Industrie aufgrund der Flexibilität und Stabilität weit verbreitet[28][48, Kap. 1.2.6].

Java ist eine objektorientierte Programmiersprache und eignet sich deshalb hervorragend zur Abbildung von Elementarmodellen und Cloud-Produkten innerhalb der Matrix-basierten Kostenoptimierung[48, Kap. 1.2]. Eine weitere Besonderheit ist der Prozess der Kompilierung, welcher ein Plattformunabhängiges Ausführen von Java-Programmen ermöglicht. Java wird nicht für eine spezielle Zielarchitektur gebaut, sondern als *Bytecode* für virtuelle Maschinen[48, Kap. 1.2.1]. Eine *Java Virtual Machine (JVM)* interpretiert den bereits kompilierten *Bytecode* und führt ihn anschließend in einer kontrollierten Umgebung aus. Für nahezu alle aktuellen Prozessorarchitekturen und Betriebssysteme sind Implementierungen einer *Java Virtual Machine* verfügbar[48, Kap. 1.2.3]. Zur Programmierung von Java-Anwendungen stehen aktuelle Entwicklungsumgebungen zur Verfügung. Zum Beispiel *Jetbrains IntelliJ IDEA* oder *Visual Studio Code*.

Des Weiteren wird Java als Programmiersprache für das Framework ICDD der *Fakultät Bauingenieurwesen, Technische Universität Dresden* eingesetzt, welches MMC erstellt und analysiert.

5 Implementierung

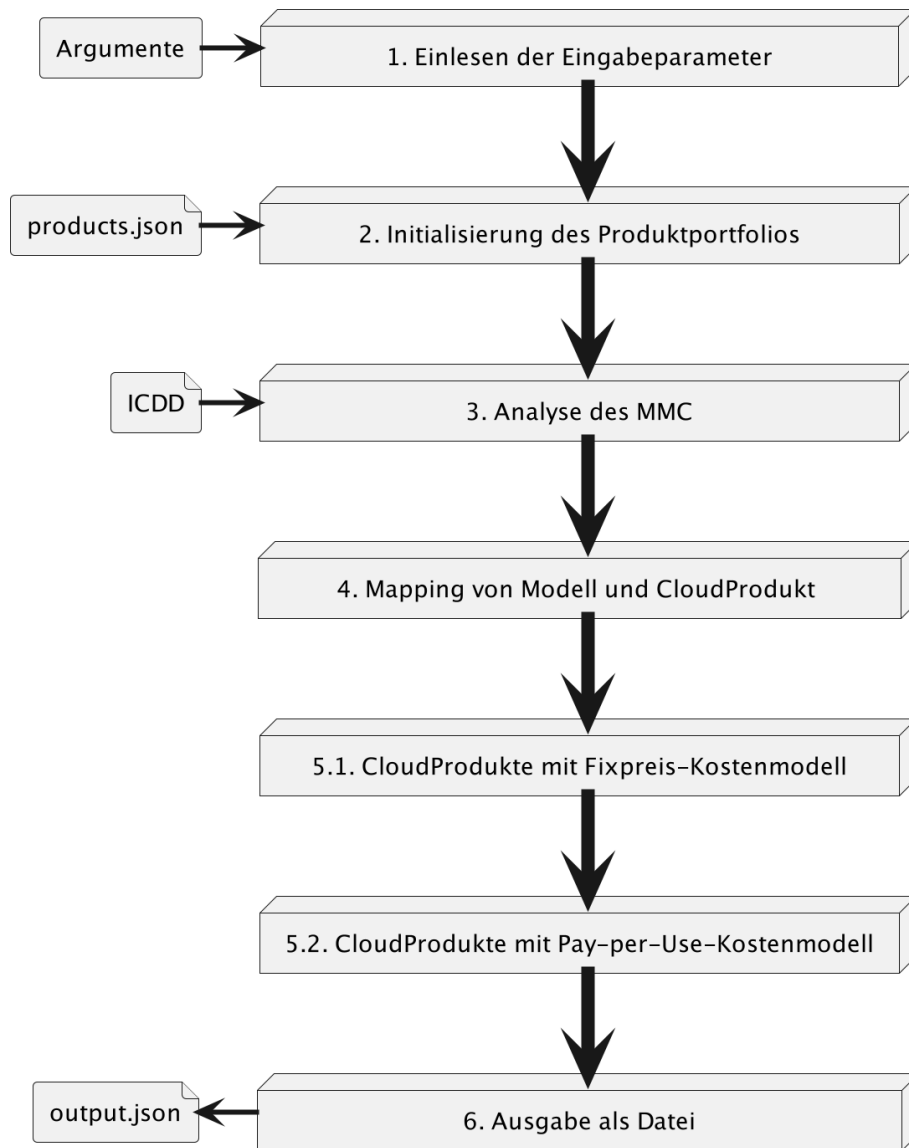


Abbildung 5.1: Ablauf der Matrix-basierten Kostenoptimierung

5.1 Umsetzung der Eingabeparameter

Die Werte der Eingabeparameter sind bei nahezu jeder Ausführung verschieden. Für ein einfaches Ändern der Eingabeparameter werden sie als Argumente beim Ausführen der Anwendung angegeben. Ein Argument setzt sich dabei immer aus einem Kriterium und dessen Wert zusammen (s. Tabelle 5.1). Ein Aufruf in der Konsole mit einigen Argumenten sieht zum Beispiel folgendermaßen aus:

```
java -jar mm-cloud-mapper.jar frequency=20000 scalability=2 availability=2 redundancy=1 region=eu
```

Eingabeparameter	Wertebereich	Konsolen-Argument
Zugriffsfrequenz	Zugriffe / Monate	frequency=20000
Skalierbarkeit	0-2	scalability=2
Verfügbarkeit	0-3	availability=2
Datenredundanz	0-2	redundancy=1
Datenhoheit	EU / Nicht-EU	region=eu

Tabelle 5.1: Eingabeparameter als Argument

Damit die Werte der Eingabeparameter während der Laufzeit des Programms zur Verfügung stehen, werden alle Argumente ausgelesen und dem jeweiligen Eingabeparameter zugewiesen. Wird für einen Eingabeparameter kein Argument angegeben, bleibt der Standardwert erhalten (s. Seite 36). Im Anschluss an das Auslesen der Argumente wird der Wertebereich der Eingabewerte geprüft. Ist der Wert eines Eingabeparameters unzulässig, so wird dieser auf den nächst zulässigen Wert gesetzt (s. Listing 9).

Der Standardwert für die Eingabeparameter entspricht dem Wert mit den geringsten Einschränkungen. Wird ein Eingabeparameter nicht angegeben, werden alle Produkte in diesem Kriterium zugelassen.

```

1  if (in_freq == 0) {
2      System.err.println("frequency is set to 0!");
3  } else {
4      System.out.println("frequency: " + in_freq);
5  }
6  if(in_scalability < 0) {
7      in_scalability = 0;
8  } else if (in_scalability > 2) {
9      in_scalability = 2;
10 }
11 if(in_availability < 0) {
12     in_availability = 0;
13 } else if (in_availability > 3) {
14     in_availability = 3;
15 }
16 if(in_redundancy < 0) {
17     in_redundancy = 0;
18 } else if (in_redundancy > 2) {
19     in_redundancy = 2;
20 }

```

Listing 9: Prüfung der Eingabeparameter

5.2 Initialisierung des Produktportfolios

Die zur Auswahl stehenden Produkte werden über die Datei *products.json* zur Verfügung gestellt. Diese befindet sich in demselben Verzeichnis wie die Anwendung und enthält einen Array von JSON-Objekten. Jedes JSON-Objekt beschreibt ein Cloud-Produkt inklusive der benötigten Eigenschaften der Matrix-basierten Kostenoptimierung (s. Seite 36).

Jedes Cloud-Produkt wird mittels der Bibliothek *gson* in die Klassen *CloudProvider* und *CloudProduct* überführt (siehe Abbildung 5.2). Um die Redundanz von Daten beim Cloud-Anbieter zu vermeiden, werden alle Cloud-Produkte einem Cloud-Anbieter zugeordnet. Die gewählte Struktur führt dazu, dass Cloud-Produkte nicht direkt über deren ID abgefragt werden können. Stattdessen ist es zusätzlich notwendig, bei der Suche nach einem Cloud-Produkt über alle Cloud-Anbieter zu iterieren.

Bevor ein Cloud-Produkt und dessen Cloud-Anbieter der *HashMap* von verfügbaren Cloud-Produkten hinzugefügt wird, werden diese entsprechend der Eingabeparameter geprüft (s. Listing 11 Zeile 1-10). Sollte das Cloud-Produkt oder der Cloud-Anbieter ein Kriterium nicht erfüllen, wird der Schleifendurchlauf für das Cloud-Produkt an der Stelle beendet.

Erfüllen das Cloud-Produkt und dessen Cloud-Anbieter die Kriterien der Eingabeparameter, wird geprüft, ob ein Objekt des Cloud-Anbieters bereits existiert. Falls nicht, wird eine neue Instanz der Klasse *CloudProvider* angelegt (s. Listing 11 Zeile 12-22). Anschließend wird eine Instanz der Klasse *CloudProduct* für das Cloud-Produkt erstellt und der Map von Produkten des Cloud-Anbieters hinzugefügt (s. Listing 11 Zeile 24-28).

Zum Ende der ersten Phase steht eine Liste mit allen zulässigen Cloud-Anbietern einschließlich deren Cloud-Produkten zur Verfügung.

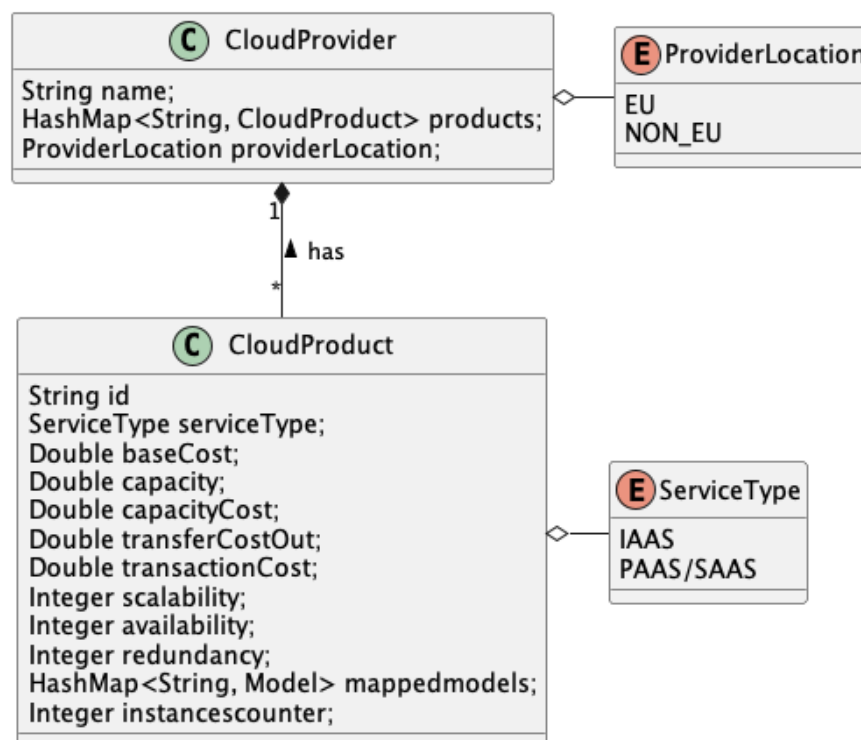


Abbildung 5.2: Klassendiagramm Cloud-Produkte

```

1 Gson gson = new Gson();
2 Reader reader = Files.newBufferedReader(productsFile.toPath());
3 List<Map<String, ?>> list = gson.fromJson(reader, List.class);
4 reader.close();

```

Listing 10: Einlesen der products.json

```

1 if (in_region.equals(ProviderLocation.EU) && !getLocation(location)
2     .equals(ProviderLocation.EU)) {
3     continue;
4 }
5 if(in_scalability > scalability || in_availability > availability
6     || in_redundancy > redundancy) {
7     continue;
8 }
9
10 CloudProvider cloudProvider;
11 if(!providers.containsKey(provider)) {
12     cloudProvider = new CloudProvider(provider, getLocation(location));
13     providers.put(provider, cloudProvider);
14 } else {
15     cloudProvider = providers.get(provider);
16 }
17 if(cloudProvider.getCloudProduct(id) != null) {
18     System.out.println("cloudProduct " + id + " has a double entry");
19     continue;
20 }
21
22 CloudProduct product = new CloudProduct(id, getServiceType(serviceType), baseCost,
23     capacity, capacityCost, transferCostOut, transactionCost, (int) scalability,
24     (int) availability, (int) redundancy);
25 cloudProvider.addProduct(product);

```

Listing 11: Prüfung und Erstellung eines Cloud-Produktes

5.3 Analyse des Multimodell-Containers

Das Ziel der MMC-Analyse ist, für jedes Modell eine Instanz der Klasse *Model* zu erhalten (s. Abbildung 5.3). Diese repräsentiert Dokumente, Linkmodelle und die Index-Datei. Für eine vollständige Bewertung ist es notwendig, alle Eigenschaften der in der Klasse *Model* benötigten Daten zur Verfügung zu stellen (s. Abbildung 5.3).

5.3.1 ICDD-Framework

Das ICDD Framework vereinfacht die Erstellung und Analyse eines MMC. Entwickelt wurde es von dem *Institut für Bauinformatik, Fakultät Bauingenieurwesen, Technische Universität Dresden*.

Unterstützt wird die Erstellung eines MMC mittels externer Ontologien. Damit ist ein Domänen-übergreifender Einsatz des ICDD-Frameworks möglich.

Das Einlesen eines MMC und die Ausgabe der enthaltenen Dokumente ist mit dem ICDD-Framework in wenigen Zeilen möglich (s. Listing 12). Auch die Linkmodelle werden beim Einlesen des MMC analysiert und können abgefragt werden.

Ebenfalls ist das Erstellen eines MMC möglich. Dafür wird zuerst ein leeres Container-Objekt im ICDD-Framework erstellt. Anschließend können Dokumente hinzugefügt und Verlinkungen erstellt werden. Das ICDD-Archiv wird zum Schluss durch das ICDD-Framework der Norm entsprechend erstellt und abgespeichert.

Das ICDD-Framework erlaubt aufgrund dessen Architektur keine direkten Anpassungen einzelner Methoden oder Klassen. Daher besteht eine direkte Abhängigkeit der im Code fest implementierten Ontologien einschließlich der externen Ressourcen, auf welche referenziert wird. Die in der ersten Version des ICDD-Frameworks verwendeten Ontologien enthalten Referenzen auf externe Ressourcen, welche nicht mehr zur Verfügung stehen. Dadurch ist das ICDD-Framework zur Analyse existierender MMC nicht zu verwenden. Die Nutzung externer Ontologien wird nur bei der Erstellung von MMC unterstützt. Bei einer neueren Version des ICDD-Frameworks wurde die URL der externen Ressourcen ersetzt. Die Abhängigkeit von externen Ressourcen ist unverändert.

Bei der Verwendung des ICDD-Frameworks zur Analyse bestehender MMC ist ein Zugriff auf die erstellten Java-Objekte möglich. Ein direkter Zugriff auf die entpackten Dateien des MMC wird nicht unterstützt. Für die Dokumente des MMC enthält die Java-Klasse (*ICDDDocument*) eine Methode zum Abrufen der Modell-Datei. Dadurch ist es möglich, die Dateigröße des jeweiligen Dokuments zu bestimmen. Für Linkmodelle existiert keine vergleichbare Methode, um Zugriff auf die Datei zu erhalten. Das Problem betrifft ebenfalls die Index-Datei. Ein Zugriff auf die Datei zum Abrufen der Dateigröße ist nicht möglich.

Die alleinige Verwendung des ICDD-Frameworks zum Abrufen aller benötigten Eigenschaften für die Matrix-basierte Kostenoptimierung ist nicht ausreichend. Für die Ermittlung der noch unbekannten Dateigrößen ist es notwendig, den MMC ein weiteres Mal zu entpacken.

```

1 File icddfile = new File({Path to container});
2 Container container = Container.readICDDContainer(icddfile);
3 String containername = container.getName();
4 String containerdescription = read.getContainerDescription().getDescription();
5 for (Iterator<ICDDDocument> iter = container.getContainerDescription().listDocuments();
6     iter.hasNext();) {
7     ICDDDocument document = iter.next();
8     System.out.println(document.getName());
9 }

```

Listing 12: Einlesen eines MMC mit dem ICDD-Framework

Das ICDD-Framework eignet sich insbesondere durch die Unterstützung externer Ontologien für die Erstellung neuer MMC. Für die Analyse bestehender MMC ist es aufgrund der Abhängigkeit von externen Ressourcen und dem Fehlen der Möglichkeit diese zu ersetzen ungeeignet.

5.3.2 XML-Parsing

Der MMC liegt als ICDD-Archiv vor. Für eine weitere Analyse des Inhaltes muss dieser zunächst entpackt werden. Das ICDD-Archiv wird dafür mit der Bibliothek *Zip4j* in ein tempo-

räres Verzeichnis entpackt. Die entpackten Dateien werden am Ende der Analyse des MMC wieder gelöscht. Sind mehrere MMC in dem Ordner *model* gegeben, dann werden diese nacheinander analysiert und bei der Auswahl der Cloud-Produkte gemeinsam betrachtet.

Analyse der Index-Datei Die Analyse des MMC beginnt mit der Index-Datei, sie beschreibt den Inhalt des MMC. Dafür wird die Eigenschaft genutzt, dass die *RDF*-Datei auf dem XML-Format basiert. Mit der *JSON*-Bibliothek wird die Index-Datei eingelesen und automatisch in ein *JSONObject* überführt. Auf dem erstellten *JSONObject* können anschließend Abfragen ausgeführt werden. Aus dem *JSONObject* wird ein *JSONArray* ausgelesen, welches eine Liste aller *RDF*-Einträge darstellt. Eine solche Abfrage auf Basis des XML-Formats ist nur möglich, weil der Aufbau der Index-Datei in der Ontologie definiert ist. Jeder Eintrag der Liste kann anhand der Namespaces unter dem Attribut *rdf:type* als Dokument ('*#Document*') oder Linkmodell ('*#Linkset*') klassifiziert werden (s. Listing 13 Zeile 4 & 14). Einträge darüber hinaus werden nicht weiter analysiert, zum Beispiel ist der Eintrag der Index-Datei einschließlich der Beschreibung des MMC für den Algorithmus nicht relevant. Für die Einträge der Dokumente und Linkmodelle werden die Eigenschaften abgefragt, welche für die Matrix-basierte Kostenoptimierung benötigt werden, zum Beispiel Modell-Identifizier, Dateiname, Dateityp und Dateigröße (s. Abbildung 5.3). Das Abfragen der Dateigröße ist direkt möglich, weil alle Dateien in dem temporären Verzeichnis vorliegen.

Des Weiteren werden alle Linkmodelle in einer Liste zwischengespeichert, damit im Anschluss die Referenzen analysiert werden können (s. Listing 13 Zeile 24).

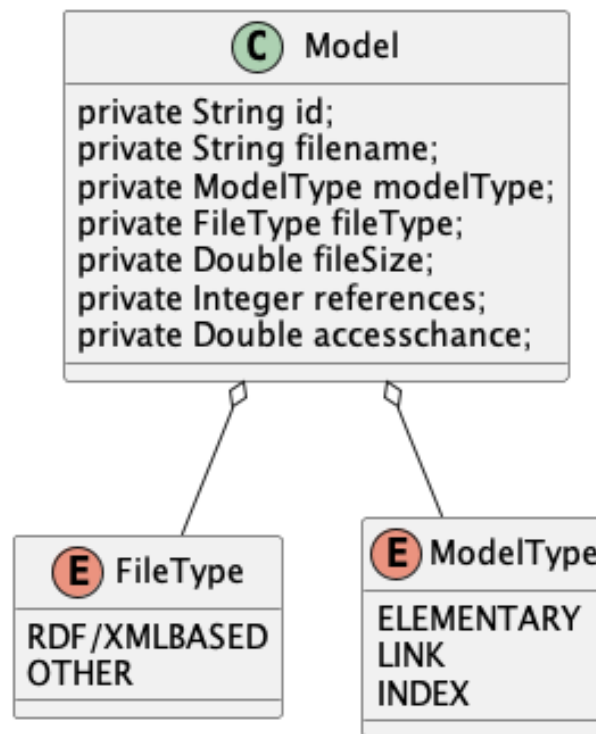


Abbildung 5.3: Klassendiagramm Multimodel

```

1  for (Object o : content) {
2      JSONObject map = (JSONObject) o;
3
4      if(map.get("rdf:type").toString().contains("#Document")) {
5          String id = map.getString("rdf:about");
6          String filename = map.getString("ct:filename");
7          FileType filetype = getFileType(map.getString("ct:filetype"));
8          File documentfile = getElementaryModelFileByName(file, filename);
9          double filesize = 0.0;
10         if (documentfile != null) {
11             filesize = ((double) documentfile.length()) / (1024*1024*1024);
12         }
13         models.put(id, new Model(id, filename, ModelType.ELEMENTARY, filetype, filesize));
14     } else if (map.get("rdf:type").toString().contains("#Linkset")) {
15         String id = map.getString("rdf:about");
16         String filename = map.getString("ct:filename");
17         FileType filetype = FileType.RDF_XMLBASED;
18         File linkfile = getLinksetFileByName(file, filename);
19         double filesize = 0.0;
20         if (linkfile != null) {
21             filesize = ((double) linkfile.length()) / (1024*1024*1024);
22         }
23         models.put(id, new Model(id, filename, ModelType.LINK, filetype, filesize));
24         linkfiles.put(id, linkfile);
25     }
26 }

```

Listing 13: Analyse der Index-Datei im XML-Format

Analyse der Linkmodelle Die Analyse der Linkmodelle kann erst durchgeführt werden, wenn die Analyse der Index-Datei abgeschlossen ist. Nur in dieser Reihenfolge ist sichergestellt, dass alle in einem Linkmodell referenzierten Dokumente bereits bekannt sind. Bei der Analyse der Linkmodelle wird ebenfalls die Eigenschaft der RDF-Dateien genutzt, dass diese auf dem XML-Format basieren. Für jedes Linkmodell wird ein *JSONObject* erzeugt, dessen Einträge als *JSONArray* abgebildet werden. Für jeden Eintrag wird geprüft, ob er den Key *ls:hasDocument* enthält (s. Listing 14 Zeile 4). Enthält ein Eintrag diesen Key, kann der Identifier des referenzierten Dokuments abgefragt werden. Für das referenzierte Dokument und das Linkmodell wird der Referenzen-Zähler um eins erhöht (s. Listing 14 Zeile 5-11).

Das Zählen der Referenzen je Dokument und Linkmodell ermöglicht es, die Verteilung der Zugriffe auf die einzelnen Modelle abzuschätzen. Die Wahrscheinlichkeit eines Zugriffs wird entsprechend der Matrix-basierten Kostenoptimierung errechnet (s. Seite 38). Dafür wird in einer ersten Schleife das Maximum der Referenzen je Modell ermittelt. Anschließend wird in einer weiteren Schleife die individuelle Wahrscheinlichkeit errechnet.

```

1  for (Object o : content) {
2      JSONObject map = (JSONObject) o;
3
4      if(map.has("ls:hasDocument")) {
5          String refid = map.getJSONObject("ls:hasDocument").getString("rdf:resource");
6          if(!models.containsKey(refid)) {
7              System.out.println("Referenced Document not found: " + refid);
8              continue;
9          }
10         models.get(refid).incReferences();
11         models.get(linkid).incReferences();
12     }
13 }

```

Listing 14: Analyse der Linkmodelle im XML-Format

5.4 Mapping von Modell und Cloud-Produkt

Für die Realisierung der Matrix aus der Matrix-basierten Kostenoptimierung wird eine Beihilfsklasse erstellt. Diese enthält eine zweidimensionale *HashMap* (s. Abbildung 5.4). Die äußere *HashMap* stellt die Zeilen der Matrix dar, die Modelle des MMC. Ein Eintrag besteht aus dem Modell-Identifizier (m1, m2, ...) und einer inneren *HashMap*. Die innere *HashMap* bildet die kompatiblen Produkte des Modells ab. Ein Eintrag besteht aus dem Produkt-Identifizier (p1, p2, ...) und den errechneten Kosten für die Abbildung des Modells mit diesem Cloud-Produkt.

Für das Erstellen der Matrix wird mittels einer Schleife jedes Modell nacheinander betrachtet. Zuerst wird eine neue Zeile für das Modell angelegt. Handelt es sich bei dem betrachteten Modell um ein Linkmodell oder die Index-Datei, werden zuerst alle Cloud-Produkte des Typs PaaS/SaaS geprüft, weil diese in einer Datenbank dargestellt werden können (s. Listing 15 Zeile 3-12). Wird kein kompatibles PaaS/SaaS-Produkt gefunden, werden auch IaaS-Produkte für das Modell herangezogen (s. Listing 15 Zeile 14-16).

Anschließend folgt die Prüfung der Cloud-Produkte vom Typ IaaS für alle Dokumente. Die geschätzten Kosten werden als Eintrag unter dem Key des Produkt-Identifiziers abgelegt (s. Listing 15 Zeile 19-26).

Die gefüllte Matrix enthält je Modell des MMC alle kompatiblen Cloud-Produkte und die entstehenden Kosten (s. Abbildung 5.4).

Äußere HashMap

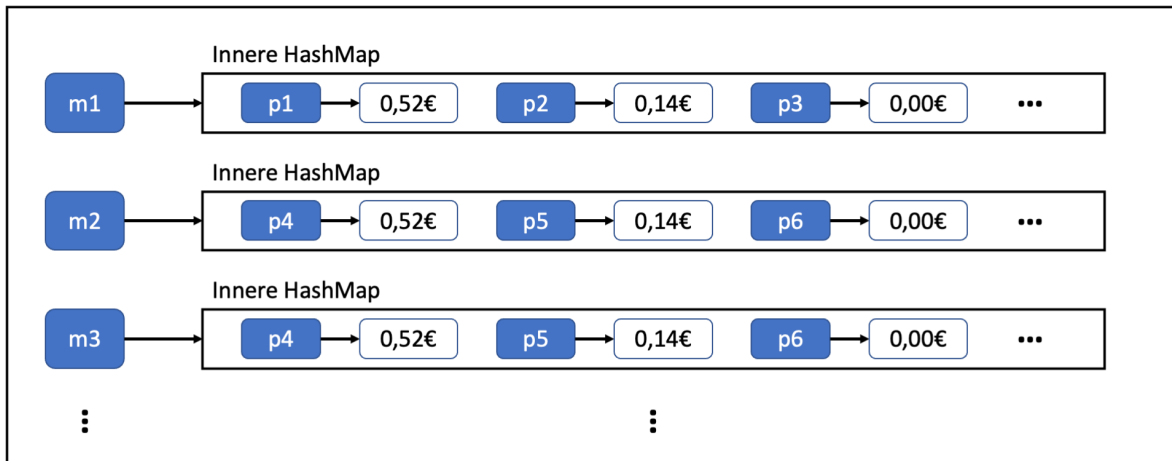


Abbildung 5.4: Aufbau Matrix aus HashMaps

```

1  matrix.createModelRow(model.getId());
2
3  if (!model.getModelType().equals(ModelType.ELEMENTARY)) {
4      for (CloudProvider provider : providers.values()) {
5          for (CloudProduct product : provider.getProducts().values()) {
6              if (product.getServiceType().equals(ServiceType.PAAS_SAAS)) {
7                  double costs = getCostForModelProduct(models, model, product, in_freq);
8                  matrix.putCosts(model.getId(), product.getId(), costs);
9              }
10         }
11     }
12 }
13
14 if(matrix.getModelRow(model.getId()).values().size() > 0) {
15 }
16
17 for (CloudProvider provider : providers.values()) {
18     for (CloudProduct product : provider.getProducts().values()) {
19         if (product.getServiceType().equals(ServiceType.IAAS)) {
20             double costs = getCostForModelProduct(models, model, product, in_freq);
21             matrix.putCosts(model.getId(), product.getId(), costs);
22         }
23     }
24 }

```

Listing 15: Zuweisen und Kostenschätzung von Modellen und Produkten

5.5 Cloud-Produkte mit Festpreis-Kostenmodell

Für die Auswahl der Cloud-Produkte mit Festpreis wird zunächst eine Liste der verfügbaren Produkte erstellt. Die Liste enthält alle Cloud-Produkte, welche in dieser Phase zur Auswahl stehen und noch nicht entfernt wurden. Die Auswahl geeigneter Cloud-Produkte wird wiederholt, bis die Liste der verfügbaren Cloud-Produkte leer ist.

Je Durchgang werden alle verfügbaren Cloud-Produkte nacheinander auf deren Rentabilität geprüft. Dafür werden alle Modelle in der Matrix ausgewählt, welche einen Eintrag zu dem aktuell betrachteten Cloud-Produkt besitzen. Die Kosten der einzelnen Modelle werden addiert und bilden mit dem Festpreis die Kosten des Cloud-Produktes. Außerdem werden je Modell auch die geringsten Kosten für ein Cloud-Produkt ohne Festpreis herausgesucht. Die Summe der Kosten für die Nutzung der alternativen Produkte bilden die Vergleichssumme. Sollte der summierte Speicherplatz aller Modelle die Kapazität des zu untersuchenden Cloud-Produktes übersteigen, so werden nur die größten Modelle in die Berechnung einbezogen.

Liegen die Kosten des Cloud-Produktes über den Kosten der Vergleichssumme, dann ist der Einsatz des Cloud-Produktes mit Festpreis nicht rentabel. Das Cloud-Produkt wird aus der Liste der zur Verfügung stehenden Cloud-Produkte mit Festpreis und aus der Matrix entfernt.

Das Cloud-Produkt mit Festpreis, das die höchste Rentabilität besitzt, wird ausgewählt. Die abgebildeten Modelle werden dem Cloud-Produkt hinzugefügt und die Zeilen der Matrix entfernt (s. Abbildung 5.5).

5.6 Cloud-Produkte mit Pay-per-Use-Kostenmodell

Die in der Matrix verbleibenden Modelle konnten keinem Produkt mit Festpreis zugewiesen werden. Für jedes verbleibende Modell wird in dessen Zeile der Eintrag mit den geringsten Kosten ausgewählt (s. Abbildung 5.4). Das Modell wird dem gewählten Cloud-Produkt hinzugefügt und anschließend aus der Matrix entfernt.

Kann ein Modell keinem Cloud-Produkt zugewiesen werden, wird dies im Log vermerkt und übersprungen.

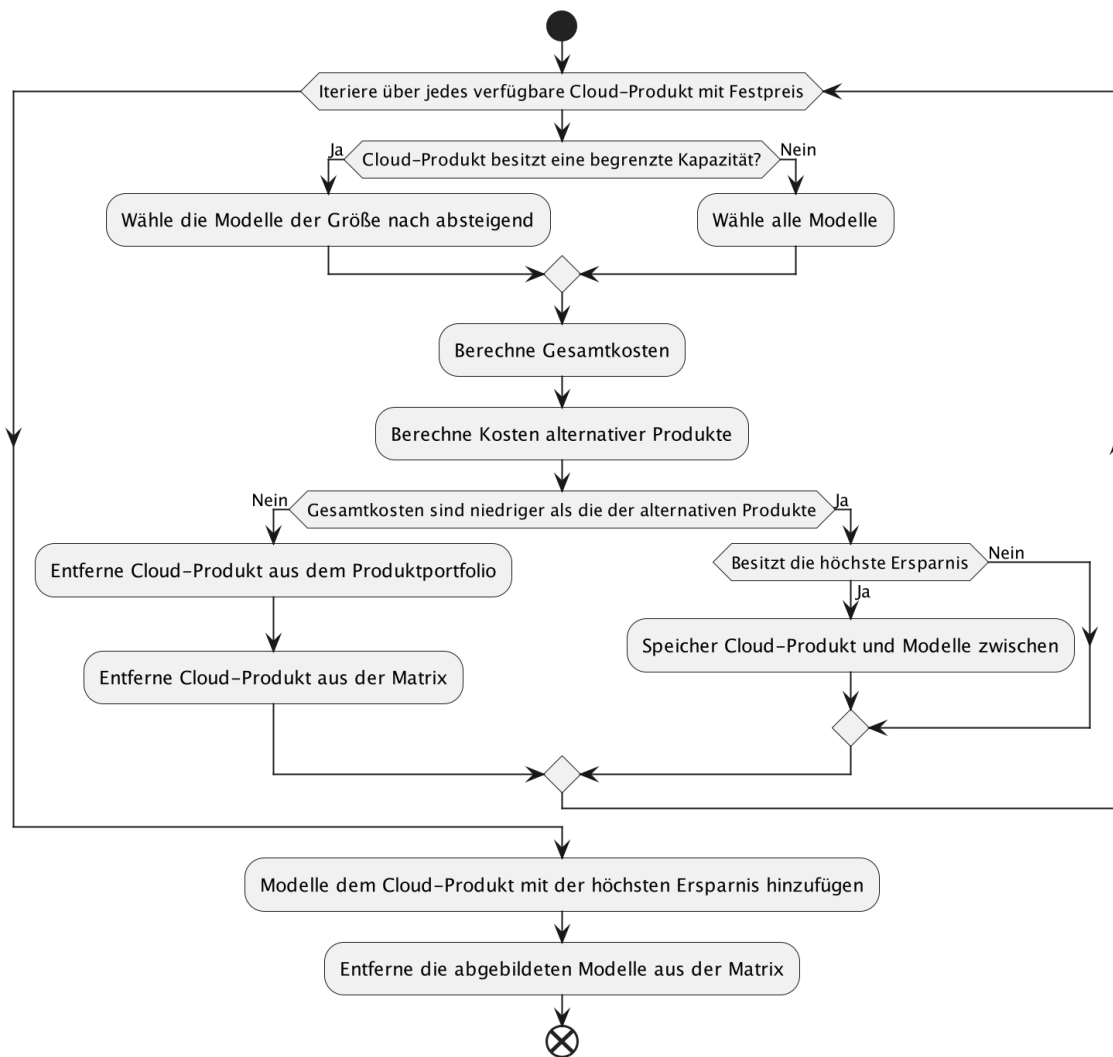


Abbildung 5.5: Auswahl eines Cloud-Produktes mit Festpreis

5.7 Ausgabe als Datei

Nach der Auswahl der Produkte sind alle Modelle einem Produkt zugewiesen worden.

Die ausgewählten Cloud-Produkte und deren zugewiesenen Modelle werden in der Datei *output.json* ausgegeben. Ist einem Cloud-Produkt mindestens ein Modell zugeordnet worden, so wird ein neues *JSONObject* für das Cloud-Produkt erstellt, zum Beispiel das Cloud-Produkt *ionos-s3* (s. Listing 16 Zeile 3). Das *JSONObject* enthält die errechneten Teilkosten für ein Cloud-Produkt (s. Listing 16 Zeile 5,7-9) und eine Liste aller darin abgebildeten Modelle, zum Beispiel die Datei *document.ifc* (s. Listing 16 Zeile 13). Alle für die Cloud-Produkte erstellten *JSONObject* werden in einem *JSONArray* zusammengefasst. Das gesamte *JSONArray* wird anschließend mit der Bibliothek *gson* in die Datei *output.json* geschrieben (s. Listing 16).

```
1  [  
2  {  
3    "productid": "ionos-s3",  
4    "instances": 1,  
5    "costs": 2.3172,  
6    "usedspace": 0.000408,  
7    "capacitycosts": 0.000006,  
8    "transactioncosts": 0.0,  
9    "transfercosts": 2.3171,  
10   "models": [  
11     {  
12       "modelid": "http://tu-dresden.de/isigg/container#{documentID}",  
13       "filename": "document.ifc"  
14     },  
15     {}  
16   ]  
17 },  
18 {}  
19 ]
```

Listing 16: Ausschnitt der Datei output.json

6 Evaluation

Im folgenden Kapitel wird zunächst die Implementierung der Matrix-basierten Kostenoptimierung an zwei Fallbeispielen evaluiert. Das Ergebnis wird dem der Nutzung eines einzelnen Cloud-Anbieters gegenübergestellt. Anschließend wird die Matrix-basierte Kostenoptimierung evaluiert und der Einfluss verschiedener Limitierungen betrachtet.

6.1 Evaluation der Implementierung

Die in Kapitel 5 realisierte Implementierung der Matrix-basierten Kostenoptimierung wird mithilfe des MMC *StationNightClub* validiert. Es werden 2 Szenarien mit verschiedenen Anforderungen untersucht (s. Tabelle 6.2 & 6.4). Der zu analysierende MMC enthält 7 Fachmodelle, 6 Linkmodelle und eine Index-Datei (s. Tabelle 6.1).

MMC StationNightClub		
	Dateiname	Dateigröße
Index	• index.rdf	12KB
Fachmodelle	• Station_MaterialData.xml • pedestrian_testcase1.owl • Station Nightclub v11-Revit 2017-DV.ifc • Station_SimPara.xml • Station_Sprinkler.xml • Station_Firesource.xml • UseCase2_SimPlan_with_furniture.0003.rvt	8KB 27KB 377KB 6KB 11KB 5KB 41,6MB
Linkmodelle	• IFC_DoorState_Link.rdf • IFC_Sprinkler_Link.rdf • IFC_Material_Link.rdf • IFC_SimPara_Link.rdf • IFC_FireSource.rdf • IFC_Pedestrian_Link.rdf	17KB 12KB 40KB 2KB 5KB 39KB

Tabelle 6.1: Inhalt des MMC StationNightClub

Szenario A - Modernisierung

Basierend auf dem bestehenden MMC soll die Modernisierung einer Immobilie geplant werden. Aufgrund des umfangreichen Vorhabens werden dem MMC viele neue Fachmodelle hinzugefügt, weshalb die gewählten Cloud-Produkte eine automatische Skalierung des Speicherplatzes unterstützen müssen. Verschiedene Teams, bestehend aus mehreren Personen, arbeiten zur gleichen Zeit an dem MMC. Es werden insgesamt zirka 2.000.000 Zugriffe im Monat erwartet. Damit die Arbeit der einzelnen Teams nicht negativ beeinflusst wird, muss die Verfügbarkeit der Daten mindestens 99,9% betragen. In Tabelle 6.2 sind die daraus resultierenden Anforderungen zusammengefasst dargestellt.

Das Programm kann über einen Befehl ausgeführt werden, die Anforderungen werden als Argumente angegeben:

```
java -jar mm-cloud-mapper.jar frequency=2000000 scalability=2 availability=2 redundancy=1
```

Nach erfolgreicher Ausführung des Programms wird die Datei *output.json* erstellt, sie enthält die ausgewählten Cloud-Produkte inklusive der geschätzten Kosten und abgebildeten Dateien (s. Tabelle 6.3).

Das empfohlene Produktportfolio besteht aus 3 Cloud-Produkten (s. Tabelle 6.3). Das erste Cloud-Produkt, *Azure Cosmos*, ist eine Datenbank für die Linkmodelle und die Index-Datei. Durch den Einsatz einer Datenbank wird die Nutzung von SQL-Abfragen ermöglicht, welche das Beschleunigen von Zugriffszeiten und das Reduzieren des Netzwerkverkehrs auf die benötigten Daten ermöglicht (s. Abschnitt 4.2). Das Cloud-Produkt *Azure Cosmos* unterscheidet sich in dem verwendeten Kostenmodell deutlich von den Alternativen anderer Anbieter. Insbesondere wird für das Zurverfügungstellen der Datenbank kein monatlicher Fixpreis verlangt, stattdessen wird ausschließlich die Nutzung berechnet.

Wird eine Datenbank für mehrere MMC genutzt oder es werden deutlich mehr Zugriffe erwartet, dann ist ein Produkt mit einem Fixpreis gegebenenfalls kostengünstiger.

Für die Fachmodelle des MMC werden die Cloud-Produkte *Azure Blob* und *AWS S3* genutzt. Beide sind Objekt-Speicher und eignen sich für eine einfache Skalierung des benötigten Speicherplatzes. Die beiden Cloud-Produkte besitzen das Kostenmodell Pay-per-Use und unterscheiden sich nur minimal in der Gestaltung von Transaktions- und Transferkosten. Je kleiner die Datei eines Modells ist, desto größer ist der Anteil der Transaktionskosten je Zugriff. Bei kleinen Dateien wird das Cloud-Produkt *AWS S3* verwendet, weil die Transaktionskosten geringer sind. Für große Dateien wird das Produkt *Azure Blob* verwendet, da die Transferkosten geringer sind. Die Kombination beider Cloud-Produkte ist möglich, weil bei beiden keine Fixkosten berechnet werden.

Setzt ein Unternehmen stattdessen zum Beispiel ausschließlich Produkte von AWS ein, kommen die beiden Produkte *AWS Aurora* und *AWS S3* zum Einsatz. Die durch die Datenbank entstehenden Kosten belaufen sich auf 58,1471 € im Monat, zirka das Fünffache des vom Algorithmus ausgewählten Cloud-Produktes *Azure Cosmos*. Für die Speicherung der Fachmodelle des MMC durch das Cloud-Produkt *AWS S3* fallen zirka 3561,976 € im Monat an, 8% mehr als in der vom Algorithmus vorgeschlagenen Kombination mit *Azure Blob*.

Zugriffsfrequenz	Skalierbarkeit	Verfügbarkeit	Datenredundanz	Datenhoheit
2.000.000	2	2	1	-

Tabelle 6.2: Szenario A - Übersicht der Eingabeparameter

Produkt	Kosten (je Monat)	Dateien
Azure Cosmos	9,5292€	• index.rdf • IFC_DoorState_Link.rdf • IFC_Sprinkler_Link.rdf • IFC_Material_Link.rdf • IFC_SimPara_Link.rdf • IFC_FireSource_Link.rdf • IFC_Pedestrian_Link.rdf
Azure Blob	3307,8177€	• pedestrian_testcase1.owl • StationNightclubv11-Revit2017-DV.ifc • UseCase2_SimPlan_with_furniture.0003.rvt
AWS S3	5,8344€	• Station_MaterialData.xml • Station_SimPara.xml • Station_Sprinkler.xml • Station_Firesource.xml
Summe	3323,1813€	

Tabelle 6.3: Szenario A - Ergebnis

Szenario B - Archivierung

In Szenario B sind die aktiven Arbeiten am MMC abgeschlossen, die Daten werden für eine spätere Verwendung verwahrt. Es wird ausschließlich zur Rechenschaftslegung gegenüber dem Eigentümer auf den MMC zugegriffen, es werden zirka 8.000 Zugriffe im Monat erwartet. Der benötigte Speicherplatz des MMC wird sich in absehbarer Zeit nicht ändern, weshalb keine Anforderungen an die Skalierbarkeit bestehen. Die Zugriffe sind nicht zeitkritisch, weshalb keine hohen Anforderungen an die Verfügbarkeit bestehen. Aus diesem Grund ist ebenfalls keine georedundante Speicherung notwendig. In der Tabelle 6.4 sind die daraus resultierenden Anforderungen zusammengefasst dargestellt.

Das Programm kann über einen Befehl in einem Terminal ausgeführt werden, die Anforderungen werden als Argumente übergeben:

```
java -jar mm-cloud-mapper.jar frequency=8000 scalability=0 availability=0 redundancy=1
```

Nach erfolgreicher Ausführung des Programms wird das Ergebnis in der Datei *output.json* ausgegeben. Für das Szenario B werden 2 Cloud-Produkte ausgewählt (s. Tabelle 6.5).

Als Cloud-Produkt für eine Datenbank wird *Azure Cosmos* genutzt. Auf der Datenbank finden nur sehr wenige Abfragen statt, weshalb ein monatlicher Fixpreis vermieden wird. Für die Fachmodelle wird das Cloud-Produkt *Hetzner BX11* genutzt, welches einen Fixpreis besitzt. In der SLA wird eine etwas geringere Verfügbarkeit garantiert als bei anderen Cloud-Anbietern, dafür sind die Gesamtkosten geringer als bei der Nutzung von beispielsweise Objekt-Speichern.

Im Zeitraum der Archivierung wird auf dem MMC nicht aktiv gearbeitet, weshalb die geringere Verfügbarkeit vernachlässigt werden kann.

Verwendet ein Unternehmen zum Beispiel ausschließlich Produkte von AWS, werden die Produkte *AWS Aurora* und *AWS S3* genutzt. Die durch die Datenbank *AWS Aurora* entstehenden Kosten belaufen sich auf zirka 50,431 € je Monat, ein Vielfaches des vorgeschlagenen Cloud-Produktes *Azure Cosmos*. Die durch die Speicherung der Fachmodelle in dem Cloud-Produkt *AWS S3* entstehenden Kosten belaufen sich auf zirka 14,2489 € im Monat, etwa

vielmehr so viel wie bei der Nutzung des Cloud-Produktes *Hetzner BX11*.

Zugriffsfrequenz	Skalierbarkeit	Verfügbarkeit	Datenredundanz	Datenhoheit
8.000	0	0	1	-

Tabelle 6.4: Szenario B - Übersicht der Eingabeparameter

Produkt	Kosten (je Monat)	Dateien
Azure Cosmos	0,0038€	• index.rdf • IFC_DoorState_Link.rdf • IFC_Sprinkler_Link.rdf • IFC_Material_Link.rdf • IFC_SimPara_Link.rdf • IFC_FireSource_Link.rdf • IFC_Pedestrian_Link.rdf
Hetzner BX11	3,81€	• pedestrian_testcase1.owl • StationNightclubv11-Revit2017-DV.ifc • UseCase2_SimPlan_with_furniture.0003.rvt • Station_MaterialData.xml • Station_SimPara.xml • Station_Sprinkler.xml • Station_Firesource.xml
Summe	3,8138€	

Tabelle 6.5: Szenario B - Ergebnis

Fazit In beiden Szenarien werden aus dem zur Verfügung stehenden Produktportfolio geeignete Cloud-Produkte ausgewählt. Die Implementierung der Matrix-basierten Kostenoptimierung errechnet auf der Basis der getroffenen Annahmen die zu erwartenden monatlichen Kosten. Außerdem können mehrere MMC in einem Durchlauf analysiert werden, um die Kosten gegebenenfalls zu reduzieren. Bereits bei dem gegebenen kleinen MMC ist zu erkennen, dass Kombinationen verschiedener Cloud-Produkte einen Kostenvorteil erbringen. Insbesondere bei einer höheren Anzahl an Zugriffen, wie in Szenario A, werden minimale Unterschiede in den Kostenmodellen berücksichtigt und gegebenenfalls kombiniert.

Durch den Aufbau der Implementierung ist es möglich, dass einzelne Phasen der Matrix-basierten Kostenoptimierung ergänzt oder ausgetauscht werden können. Daher ist eine Anpassung an zukünftige Anforderungen möglich, zum Beispiel für die Berücksichtigung von realen Nutzungswerten des jeweiligen MMC.

Die Implementierung der Matrix-basierten Kostenoptimierung analysiert gegebene MMC und trifft auf Basis von gegebenen Annahmen Aussagen über die zu erwartenden Kosten. Die Kombination von Cloud-Produkten basiert auf einem individuellen Produktportfolio, welches über die Datei *products.json* angepasst werden kann. Durch die Ausgabe der ausgewählten Cloud-Produkte als JSON-Datei und das Ausführen des Programms durch einen Befehl kann die Implementierung in automatisierte Arbeitsabläufe eingebunden werden.

6.2 Evaluation der Speicherstrategie

Die Matrix-basierte Kostenoptimierung ermöglicht die präzise Abbildung der beiden Kostenmodelle Fixpreis und Pay-per-Use. Durch den zweistufigen Auswahlprozess wird jede Kombination von Cloud-Produkten mit dem Kostenmodell Pay-per-Use denen mit Fixpreis gegenübergestellt. Dadurch werden die möglichen Abweichungen der geschätzten von den realen Kosten durch unterschiedliche Kostenmodelle minimiert. Die Abbildung der Kostenmodelle in der Matrix-basierten Kostenoptimierung sind eine Annäherung an die realen Kostenmodelle von Cloud-Anbietern. Diese werden vereinfacht abgebildet auf die Daten Fixpreis, Kapazität, Speicherkosten, Transaktionskosten und Transferkosten. Das ermöglicht den Vergleich verschiedener Cloud-Produkte.

Viele Cloud-Anbieter gewähren einen Rabatt bei der umfangreichen Nutzung von Cloud-Produkten. Diese Preisstufen sind in der Matrix-basierten Kostenoptimierung nicht berücksichtigt, weil die Größe der Stufen in den meisten Fällen für die Nutzung bei einem einzelnen MMC keinen Einfluss hat. Zum Beispiel bietet AWS einen Preisnachlass ab 10 TB Netzwerkverkehr, welcher erst bei mehreren durchschnittlichen MMC erreicht werden könnte. Das Berücksichtigen von Preisstufen führt zu einer zusätzlichen Komplexität, welche nicht durch den zweistufigen Auswahlprozess zu realisieren ist.

Teilweise stellen Cloud-Anbieter auch kostenfreie Kontingente je Cloud-Produkt bereit. Bei kleinen MMC können diese Kontingente die realen Kosten deutlich reduzieren, aber nur, wenn es sich um den ersten MMC des Unternehmens handelt. Kostenfreie Kontingente gelten bei einem Cloud-Anbieter nur einmalig je Account, weshalb bereits genutzte Cloud-Produkte mit in die Auswahl einbezogen werden müssen. Das Einbeziehen bestehender Cloud-Produkte ist nicht Bestandteil der Matrix-basierten Kostenoptimierung.

Die Performance verschiedener Cloud-Produkte, welche den gleichen Dienst anbieten, wird in der Matrix-basierten Kostenoptimierung nicht berücksichtigt. Das betrifft zum Beispiel die Latenz, Netzwerkbandbreite, Lese- und Schreibgeschwindigkeit. Darüber hinaus müssen für das Kriterium der Performance weitere Annahmen bezüglich der Art des Zugriffs und der Architektur der zugreifenden Software getroffen werden. Die meisten Cloud-Produkte verfügen über eine ausreichende Performance für durchschnittliche Nutzungsszenarien eines MMC. Ist eine hohe Performance besonders relevant, zum Beispiel eine geringe Latenz oder schnelle Lese- und Schreibgeschwindigkeiten für große Teams, lassen sich diese Anforderungen über die Datenhoheit und die Skalierbarkeit annähern. Zum Beispiel durch die Einschränkung der Datenhoheit auf die Europäische Union kann die Latenz für europäische Unternehmen deutlich reduziert werden. Cloud-Produkte, welche die höchste Anforderung an die Skalierbarkeit erfüllen, sind meistens große dezentrale Systeme, welche mit einer hohen Beanspruchung der Ressourcen umgehen können.

Der Inhalt und Aufbau eines MMC stellt keine Anhaltspunkte dazu bereit.

Alle Annahmen sind auf dem aktuellen Stand der Forschung getroffen und orientieren sich ausschließlich an den Eigenschaften des MMC. Ein Validieren und gegebenenfalls Anpassen dieser Annahmen in der Praxis steht noch aus.

7 Zusammenfassung und Ausblick

In dem folgenden Kapitel wird die Arbeit kurz zusammengefasst und ein Ausblick für künftige Themen gegeben.

7.1 Zusammenfassung der Diplomarbeit

Das Ziel der vorliegenden Arbeit ist, für gegebene Multimodell-Container eine Speicherstrategie basierend auf Cloud-Produkten aus der Public Cloud zu entwickeln (s. Abschnitt 1.2). Die Motivation des Ansatzes besteht darin, die Multimodelle in einer Cloud-Umgebung zu speichern und ein kollaboratives Arbeiten an diesen zu ermöglichen.

Die Nutzung von Multimodellen ermöglicht einen Informationsgewinn durch die Verknüpfung verschiedener Fachmodelle. Aufgrund der losen Kopplung von Fachmodellen durch separate Linkmodelle können die Fachmodelle ohne Anpassungen der dazugehörigen Software weiterverwendet werden (s. Abschnitt 2.2).

Das kollaborative Arbeiten über mehrere Teams und verschiedene Standorte hinweg ist bei umfangreichen Projekten unerlässlich, weshalb Alternativen zu der lokalen Speicherung, beispielsweise in der Form eines ICDD untersucht werden. Eine Option ist die Nutzung von Produkten der Public Cloud, mit welchen die Arbeit auf einem gemeinsamen Datenbestand ermöglicht wird. Für den Vergleich von Cloud-Produkten verschiedener Anbieter werden die jeweiligen Kostenmodelle auf ein einheitliches Produktschema der Matrix-basierten Kostenoptimierung abgebildet.

Durch die Nutzung weiterer Cloud-Produkte wie zum Beispiel Datenbanken kann die Nutzung von Multimodellen, insbesondere in der Cloud deutlich optimiert werden. Dafür werden die Index-Datei und Linkmodelle von dem auf Dateien basierenden RDF-Format in eine relationale Datenbank überführt. Anschließend wird mittels geeigneter Abfragen ausschließlich auf die benötigten Daten zugegriffen, was zur Optimierung der genutzten Ressourcen beiträgt. (s. Abschnitt 4.2)

Basierend auf den Eingabeparametern wird aus den zur Verfügung stehenden Cloud-Produkten, ein Portfolio der den Anforderungen entsprechenden Cloud-Produkten zusammengestellt.

Durch die Matrix-basierte Kostenoptimierung wird je Multimodell ein individuelles Portfolio an Cloud-Produkten zusammengestellt. Dafür werden alle in einem Multimodell enthaltenen Elementarmodelle jeweils analysiert, einschließlich der aus den Linkmodellen hervorgehenden Verknüpfungen von Elementarmodellen. Auf Basis dieser Analyse wird eine

Verteilung der zu erwartenden Zugriffe auf die Elementarmodelle eines Multimodells erstellt.

Die zu erwartenden Zugriffe je Elementarmodell ermöglichen gemeinsam mit weiteren Informationen, wie zum Beispiel der Dateigröße und -format, eine präzise Schätzung der benötigten Ressourcen.

Durch den zweistufigen Auswahlprozess der Matrix-basierten Kostenoptimierung wird für das analysierte Multimodell ein optimiertes Produktportfolio zusammengestellt. Insbesondere werden Cloud-Produkte mit monatlichen Festpreisen den realen Kosten von anderen Cloud-Produkten gegenübergestellt. Dafür werden in der ersten Phase des Auswahlprozesses Cloud-Produkte mit Festpreisen in deren Nutzung optimiert und den Alternativen gegenübergestellt. Ist die Nutzung eines Cloud-Produktes nicht rentabler als dessen Alternativen, wird dieses aus dem Portfolio der zur Verfügung stehenden Cloud-Produkte entfernt. Im Anschluss werden in der zweiten Phase des Auswahlprozesses alle Cloud-Produkte mit dem Kostenmodell Pay-per-Use berücksichtigt und entsprechend der Anforderungen und zu erwartenden Kosten ausgewählt. (s. Abschnitt 4.3.2)

Zur Überprüfung der Speicherstrategie, der Matrix-basierten Kostenoptimierung wurde in Kapitel 5 eine prototypische Implementierung in der Programmiersprache Java realisiert. Die vollständige Funktionsweise der Speicherstrategie konnte an dem Multimodell *StationNightClub* in Kapitel 6 demonstriert werden.

7.2 Ausblick und zukünftige Themen

Die im Rahmen dieser Arbeit entstandene Speicherstrategie der Matrix-basierten Kostenoptimierung zeigt, dass auf der Basis eines analysierten MMC ein geeignetes Produktportfolio aus Cloud-Produkten mit verschiedenen Kostenmodellen zusammengesetzt werden kann. Zusätzlich kann die Speicherstrategie weitergehend durch das Einbeziehen weiterer realer und externer Faktoren optimiert werden.

Externe Dokumente in Multimodellen

Das Konzept der Multimodelle ermöglicht neben internen Dokumenten, welche sich zum Beispiel in einem ICDD-Archiv befinden, die Einbindung externer Dokumente. Der Speicherort wird mittels einer URL angegeben. Externe Dokumente werden bei der Matrix-basierten Kostenoptimierung nicht betrachtet. Gegebenenfalls ist die Überführung externer Dokumente in gemeinsam genutzte Cloud-Produkte hinsichtlich der zu erwartenden Kosten hilfreich. Auch die Analyse von externen Dokumenten muss näher untersucht werden, zum Beispiel durch das Herunterladen dieser Dokumente.

Berücksichtigung bestehender Cloud-Produkte

In der Matrix-basierten Kostenoptimierung werden bereits genutzte Cloud-Produkte nicht berücksichtigt. Für den langfristigen Einsatz in der Praxis ist es von Relevanz, bereits zum Teil genutzte Cloud-Produkte mit Festpreis im Auswahlprozess zu berücksichtigen.

Versionsverwaltung

Eine zentrale Versionsverwaltung ist für Projekte, an welchen mehrere Teams beteiligt sind, notwendig. Dadurch lassen sich Änderungen genehmigen und zurückverfolgen, außerdem

ist es möglich einen Stand des Multimodells zur externen Begutachtung festzulegen. Ebenfalls kann eine Ressourcen-Verwaltung mit einbezogen werden, welche mögliche Ressourcen-Konflikte, zum Beispiel einen zeitgleichen Schreibzugriff löst. Die Versionsverwaltung muss für die Speicherung über verschiedene Cloud-Produkte hinweg ausgelegt sein.

Berücksichtigung realer Zugriffsverteilungen

Die reale Verteilung der Zugriffe auf die einzelnen Modelle weicht gegebenenfalls von den allgemeinen Annahmen in Abschnitt 4.3.2 ab. Je nach konkretem Kostenmodell und Nutzungsszenario können die realen Kosten in wenigen Fällen deutlich von den geschätzten Kosten abweichen.

Die auftretenden Abweichungen und die Möglichkeit, diese in den Algorithmus mit einzubeziehen, müssen separat untersucht werden.

Graphen-basierte Kostenoptimierung

Die Graphen-basierte Kostenoptimierung ist eine alternative Speicherstrategie, deren Motivation dem *Dijkstra-Algorithmus* entspringt. Gegenüber der Matrix-basierten Kostenoptimierung werden alle möglichen Kombinationen von Cloud-Produkten und Modellen bewertet. Insbesondere bei komplexen Multimodellen oder dem Einbeziehen weiterer Metadaten kann der Auswahlprozess von Cloud-Produkten mit Festpreis gegebenenfalls kein optimales Produktportfolio zusammenstellen.

Die Graphen-basierte Kostenoptimierung verfolgt den Ansatz, dass ein Baum von der Wurzel ausgehend aufgespannt wird. Jeder Knoten repräsentiert eine Menge an bereits abgebildeten Modellen. Die Kanten repräsentieren ein gewähltes Cloud-Produkt, der Pfad zu einem Blatt bildet das gewählte Produktportfolio. Iterativ werden dabei von der Wurzel ausgehend verschiedene Cloud-Produkte und Modelle abgebildet. Ein Knoten ist Teil der Randknotenmenge, wenn nicht alle verfügbaren Cloud-Produkte auf diesen angewandt wurden. Je Iterationsschritt wird der Knoten der Randknotenmenge, welcher bisher die geringsten Gesamtkosten beinhaltet, mit einem weiteren Cloud-Produkt kombiniert. Der Algorithmus wird beendet, sobald ein Knoten alle Modelle abbildet und die Gesamtkosten geringer sind als die Gesamtkosten aller Knoten in der Randknotenmenge.

Die theoretische Untersuchung und Beweisführung der Graphen-basierten Kostenführung sowie die Gegenüberstellung der Matrix-basierten Kostenoptimierung müssen separat untersucht werden.

Abkürzungsverzeichnis

ACU Aurora Capacity Unit

AWS Amazon Web Services

AZ Availability Zone

BIM Building Information Modeling

BSI Bundesamt für Sicherheit in der Informationstechnik

DSGVO Datenschutz-Grundverordnung

GCP Google Cloud Platform

IaaS Infrastructure-as-a-Service

ICDD Information Container for Data Drop

JSON JavaScript Object Notation

MMC Multimodell-Container

PaaS Platform-as-a-Service

RAID Redundant Array of Independent Disks

RDF Resource Description Framework

SaaS Software-as-a-Service

SLA Service-Level-Agreement

SQL Structured Query Language

URL Uniform Resource Locator

USV Unterbrechungsfreie Stromversorgung

XML Extensible Markup Language

Abbildungsverzeichnis

2.1	Übersicht Service-Modelle (Nach [68])	6
3.1	Dateistruktur eines MMC	12
4.1	Aktivitätsdiagramm Analyse Modell	32
4.2	Relation Elementarmodelle und zulässige Cloud-Produkte	38
4.3	Matrixdarstellung - Modelle zu Cloud-Produkten	39
4.4	Matrixdarstellung - Modelle zu Cloud-Produkten mit Festpreis	40
4.5	Matrixdarstellung - Modelle, welche sich mit dem Cloud-Produkt p2 abbilden lassen	41
4.6	Matrixdarstellung - Festpreis-Produkte nach Kostenvergleichsrechnung	42
4.7	Matrixdarstellung - Auswahl von Pay-per-Use-Produkten	43
5.1	Ablauf der Matrix-basierten Kostenoptimierung	46
5.2	Klassendiagramm Cloud-Produkte	48
5.3	Klassendiagramm Multimodel	51
5.4	Aufbau Matrix aus HashMaps	54
5.5	Auswahl eines Cloud-Produktes mit Festpreis	56

Tabellenverzeichnis

3.1	Abbildung eines Linkmodells in einer relationalen Datenbank	15
3.2	Verfügbarkeit je Monat (30 Tage)	18
3.3	Auszug - AWS Cloud-Produkte	19
3.4	Auszug - Azure Cloud-Produkte	20
3.5	Auszug - GCP Cloud-Produkte	22
3.6	Übersicht Cloud-Anbieter	25
4.1	Produktkosten AWS S3 und AWS Aurora (Stand: November 2022)	28
4.2	Gegenüberstellung Kosten je Speichermethode	28
4.3	Übersicht der Eingabeparameter	36
4.4	Gegenüberstellung Modellorientierte und Matrix-basierte Kostenoptimierung	44
5.1	Eingabeparameter als Argument	47
6.1	Inhalt des MMC StationNightClub	58
6.2	Szenario A - Übersicht der Eingabeparameter	59
6.3	Szenario A - Ergebnis	60
6.4	Szenario B - Übersicht der Eingabeparameter	61
6.5	Szenario B - Ergebnis	61

Quellcodeverzeichnis

1	Aufbau der Index-Datei	13
2	Dokument- und Linkset-Eintrag in der Index-Datei	14
3	Link eines Linkmodells	15
4	SQL-Abfrage zur Auflistung aller verlinkten Dokumente	15
5	Ausschnitt der products.json für die Modellorientierte Kostenoptimierung . .	31
6	Auszug der Ausgabe-Datei	35
7	Ausschnitt der Datei products.json	37
8	Auszug der Ausgabe-Datei	43
9	Prüfung der Eingabeparameter	47
10	Einlesen der products.json	49
11	Prüfung und Erstellung eines Cloud-Produktes	49
12	Einlesen eines MMC mit dem ICDD-Framework	50
13	Analyse der Index-Datei im XML-Format	52
14	Analyse der Linkmodelle im XML-Format	53
15	Zuweisen und Kostenschätzung von Modellen und Produkten	54
16	Ausschnitt der Datei output.json	57

Literatur

- [1] *About Alibaba Cloud: The Pulse of Digitalization*. <https://www.alibabacloud.com/de/about>. Accessed: 2022-11-12.
- [2] *AGB - Hetzner Online GmbH*. <https://www.hetzner.com/de/legal/terms-and-conditions>. Accessed: 2022-11-09.
- [3] *Alibaba Cloud International Website Object Storage Service Service Level Agreement*. <https://www.alibabacloud.com/help/en/legal/latest/object-storage-service-service-level-agreement?spm=a2c63.p38356.0.0.120d79525PeJ1x>. Accessed: 2022-11-12.
- [4] *Alibaba Cloud Security & Privacy Compliance Center for Cloud Computing*. <https://www.alibabacloud.com/de/trust-center/compliance>. Accessed: 2022-11-12.
- [5] *Allgemeine Geschäftsbedingungen (AGB) der IONOS Cloud - IONOS AGB*. <https://www.ionos.de/terms-gtc/terms-enterprise-cloud/enterprise-terms/>. Accessed: 2022-11-12.
- [6] *Amazon Aurora Service Level Agreement*. <https://aws.amazon.com/de/s3/sla/>. Accessed: 2022-11-07.
- [7] *Amazon Compute Service Level Agreement*. https://d1.awsstatic.com/legal/AmazonComputeServiceLevelAgreement/Amazon%20Compute%20Service%20Level%20Agreement_German_2020-07-22.pdf. Accessed: 2022-11-07.
- [8] *Amazon S3 SLA*. <https://aws.amazon.com/de/s3/sla/>. Accessed: 2022-11-07.
- [9] *AnalyticDB for PostgreSQL Service Level Agreement*. <https://www.alibabacloud.com/help/en/legal/latest/analyticdb-for-postgresql-service-level-agreement>. Accessed: 2022-11-12.
- [10] *Azure-Sicherheit | Microsoft Azure*. <https://azure.microsoft.com/de-de/explore/security/>. Accessed: 2022-11-08.
- [11] *BESONDERE VERTRAGSBEDINGUNGEN OVH PUBLIC CLOUD*. https://storage.gra.cloud.ovh.net/v1/AUTH_325716a587c64897acbef9a4a4726e38/contracts/8a43c63-Conditions_particulieres_OVH_Stack-DE-15.0.pdf. Accessed: 2022-11-09.
- [12] *Billable items*. <https://www.alibabacloud.com/help/en/polardb-for-mysql/latest/billable-items>. Accessed: 2022-11-12.
- [13] Matthias Book, Volker Gruhn und Rüdiger Striemer. *Erfolgreiche agile Projekte*. 2016. ISBN: 9783662533291.
- [14] Greg Boss, Padma Malladi und Dennis Quan. *CLOUD COMPUTING*. 2007. URL: www.ibm.com/developerworks/websphere/zones/hipods/.

- [15] *BSI - Kriterienkatalog C5*. https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Empfehlungen-nach-Angri-ffszielen/Cloud-Computing/Kriterienkatalog-C5/kriterienkatalog-c5_nod-e.html. Accessed: 2022-11-19.
- [16] *BSI C5:2020 | Google Cloud*. <https://cloud.google.com/security/compliance/bsi-c5>. Accessed: 2022-11-08.
- [17] *C5 Standard - Amazon Web Services (AWS)*. <https://aws.amazon.com/de/compliance/bsi-c5/>. Accessed: 2022-11-07.
- [18] Tien-Hsiang Chuang, Bo-Cing Lee und I-Chen Wu. *Applying Cloud Computing Technology to BIM visualization and manipulation*. 2011.
- [19] *Cloud Compliance | Oracle*. <https://www.oracle.com/corporate/cloud-compliance/>. Accessed: 2022-11-09.
- [20] *Cloud Computing - Marktanteile der führenden Unternehmen 2022 | Statista*. <https://de.statista.com/statistik/daten/studie/150979/umfrage/marktanteile-der-fuehrenden-unternehmen-im-bereich-cloud-computing/>. Accessed: 2022-11-07.
- [21] *Cloud Object Storage - IBM Cloud*. <https://cloud.ibm.com/objectstorage/create#pricing>. Accessed: 2022-11-11.
- [22] *Cloud SQL Service Level Agreement (SLA) | Google Cloud*. <https://cloud.google.com/sql/sla>. Accessed: 2022-11-08.
- [23] *Cloud Storage Service Level Agreement (SLA) | Google Cloud*. <https://cloud.google.com/storage/sla>. Accessed: 2022-11-08.
- [24] *Compute Engine Service Level Agreement (SLA) | Google Cloud*. <https://cloud.google.com/compute/sla>. Accessed: 2022-11-08.
- [25] *Daten und Sicherheit – Rechenzentren – Google*. <https://www.google.com/about/datacenters/data-security/>. Accessed: 2022-11-08.
- [26] *Datensouveränität: Definition, Herausforderungen und Lösungen | OVHcloud*. <https://www.ovhcloud.com/de/about-us/data-sovereignty/>. Accessed: 2022-11-09.
- [27] *De-Risking Cloud Consumption: IBM Cloud for Financial Services | eWEEK*. <https://www.eweek.com/cloud/de-risking-cloud-consumption-ibm-cloud-for-financial-services/>. Accessed: 2022-11-11.
- [28] *Die beliebtesten Programmiersprachen weltweit laut PYPL-Index im November 2022*. <https://de.statista.com/statistik/daten/studie/678732/umfrage/beliebteste-programmiersprachen-weltweit-laut-pypl-index/>. Accessed: 2022-11-23. Nov. 2022.
- [29] *Die Open Telekom Cloud ist live! | Deutsche Telekom*. <https://www.telekom.com/de/medien/medieninformationen/detail/die-open-telekom-cloud-ist-live--351686>. Accessed: 2022-11-12.
- [30] *DSGVO: Cloud Datenschutz & Compliance - Open Telekom Cloud*. <https://open-telekom-cloud.com/de/sicherheit/datenschutz-compliance>. Accessed: 2022-11-12.
- [31] *EU Data Boundary for the Microsoft Cloud: A progress report - EU Policy Blog*. <https://blogs.microsoft.com/eupolicy/2021/12/16/eu-data-boundary-for-the-microsoft-cloud-a-progress-report/>. Accessed: 2022-11-09.
- [32] *Everything in its write place: Cloud storage abstraction with Object Store - Dropbox*. <https://dropbox.tech/infrastructure/abstracting-cloud-storage-backends-with-object-store>. Accessed: 2022-12-02.

- [33] Sebastian Fuchs. *Erschließung domänenübergreifender Informationsräume mit Multimodellen*. ISBN: 9783867804516.
- [34] Sebastian Fuchs und Raimar J. Scherer. *Multimodels - Instant nD-modeling using original data*. 2016. DOI: 10.1016/j.autcon.2016.11.013.
- [35] Sebastian Fuchs, Mathias Kadolsky und Raimar J. Scherer. „Formal description of a generic multi-model“. In: Methodology for overall process to develop, implement and use model views filtering on Schema Level, Class Level, Object Level. 2011, S. 205–210. ISBN: 9780769544106. DOI: 10.1109/WETICE.2011.34.
- [36] *Georedundanz - Open Telekom Cloud*. <https://open-telekom-cloud.com/de/loesungen/use-cases/disaster-recovery/georedundanz>. Accessed: 2022-11-15.
- [37] *Google App Engine Blog: Introducing Google App Engine + our new blog*. <https://googleeappengine.blogspot.com/2008/04/introducing-google-app-engine-our-new.html>. Accessed: 2022-11-08.
- [38] *Google Infrastructure Security Design Overview*. https://cloud.google.com/docs/security/infrastructure/design/resources/google_infrastructure_whitepaper_fa.pdf/. Accessed: 2022-11-08.
- [39] Fuchs Hilbert Schülbe. *Context-dependent information space for construction information processes*. 2019. DOI: 10.1088/1755-1315/323/1/012111.
- [40] *History of Microsoft Azure*. <https://techcommunity.microsoft.com/t5/educator-developer-blog/the-history-of-microsoft-azure/ba-p/3574204>. Accessed: 2022-11-08.
- [41] *IaaS Public Cloud Services Market Grew in 2018 | Gartner*. <https://www.gartner.com/en/newsroom/press-releases/2019-07-29-gartner-says-worldwide-iaas-public-cloud-services-market-grew-31point3-percent-in-2018>. Accessed: 2022-11-11.
- [42] *IBM and SoftLayer: What a difference a year makes | IT PRO*. <https://www.itpro.co.uk/cloud/23266/ibm-and-softlayer-what-a-difference-a-year-makes>. Accessed: 2022-11-11.
- [43] *IBM Cloud Compliance Programs | IBM*. <https://www.ibm.com/cloud/compliance>. Accessed: 2022-11-11.
- [44] *IBM Cloud Global Data Centers | IBM*. <https://www.ibm.com/cloud/data-centers>. Accessed: 2022-11-11.
- [45] *IBM Terms*. <https://www.ibm.com/support/customer/csol/terms?id=i126-9268&lc=de#>. Accessed: 2022-11-11.
- [46] *Impressum - Open Telekom Cloud*. <https://open-telekom-cloud.com/de/impressum>. Accessed: 2022-11-12.
- [47] *IONOS Cloud Documentation - Products*. <https://docs.ionos.com/cloud/>. Accessed: 2022-11-12.
- [48] *Java ist auch eine Insel*. 15. Ausgabe. Rheinwerk computing, 2020. ISBN: 9783836277372.
- [49] *Open Telekom Cloud - Leistungsbeschreibung*. https://open-telekom-cloud.com/_Resources/Persistent/0/4/1/f/041fe599ef3cbace66ec992eae404708e79b005d/open-telekom-cloud-leistungsbeschreibung-tdg.pdf. Accessed: 2022-11-12.
- [50] *Oracle Cloud Infrastructure Platform Overview*. <https://www.oracle.com/a/ocom/docs/cloud/oracle-cloud-infrastructure-platform-overview-wp.pdf>. Accessed: 2022-11-09.

- [51] *OVHcloud Corporate – Geschichte und Fakten* | OVHcloud. <https://corporate.ovhcloud.com/de/company/history/>. Accessed: 2022-11-09.
- [52] David A. Patterson, Garth Gibson und Randy H. Katz. *A Case for Redundant Arrays of Inexpensive Disks (RAID)*. 1988. DOI: 10.1145/50202.50214.
- [53] *PolicyOracle PaaS and IaaS Public Cloud Services Pillar Documentation*. <https://www.oracle.com/assets/paas-iaas-pub-cld-srvs-pillar-4021422.pdf>. Accessed: 2022-11-09.
- [54] *Produkte und Dienste* | Google Cloud. <https://cloud.google.com/products>. Accessed: 2022-11-08.
- [55] *Rechenzentren mit Standortvorteil - Open Telekom Cloud*. <https://open-telekom-cloud.com/de/sicherheit/rechenzentren>. Accessed: 2022-11-11.
- [56] *Rechenzentren und Anbindung - Hetzner Docs*. <https://docs.hetzner.com/de/general/others/data-centers-and-connection>. Accessed: 2022-11-09.
- [57] *Service Level Agreement - IONOS AGB*. <https://www.ionos.de/terms-gtc/terms-enterprise-cloud/enterprise-agreement/>. Accessed: 2022-11-12.
- [58] Bundesamt für Sicherheit in der Informationstechnik (BSI). *Cloud Computing Compliance Criteria Catalogue – C5:2020*. Okt. 2020.
- [59] *Sicherheit und Compliance* | OVHcloud. <https://corporate.ovhcloud.com/de/trusted-cloud/security-certifications/>. Accessed: 2022-11-09.
- [60] *SLA für Azure SQL-Datenbank* | Microsoft Azure. https://azure.microsoft.com/de-de/support/legal/sla/azure-sql-database/v1_8/. Accessed: 2022-11-08.
- [61] *SLA für Speicherkonten* | Microsoft Azure. https://azure.microsoft.com/de-de/support/legal/sla/storage/v1_5/. Accessed: 2022-11-08.
- [62] *SLA für Virtuelle Computer* | Microsoft Azure. https://azure.microsoft.com/de-de/support/legal/sla/virtual-machines/v1_9/. Accessed: 2022-11-08.
- [63] H. Stachowiak. *Allgemeine Modelltheorie*. 1973. ISBN: 9783867804516.
- [64] *The importance of securing data in the cloud* | IBM. <https://www.ibm.com/cloud/smarterpapers/securing-data-in-the-cloud/>. Accessed: 2022-11-11.
- [65] *Über uns - Das Unternehmen im Überblick* | IONOS. https://www.ionos.de/unternehmen?linkId=ft.nav.company&_gl=1*1svvxvt*_ga*MTc4MTE3MDkzOC4xNjY1NDc2MDE4*_ga_29DKNVLRWX*MTY2ODI0DE5My4xLjEuMTY2ODI0DI3OC42MC4wLjA. Accessed: 2022-11-12.
- [66] *Über uns - Hetzner Online GmbH*. <https://www.hetzner.com/de/unternehmen/ueber-uns>. Accessed: 2022-11-09.
- [67] *Über uns* | OVHcloud. <https://www.ovhcloud.com/de/about-us/>. Accessed: 2022-11-09.
- [68] *Vergleich von IaaS, PaaS und SaaS*. <https://www.redhat.com/de/topics/cloud-computing/iaas-vs-paas-vs-saas>. Accessed: 2023-01-03.
- [69] Gottfried Vossen, Till Haselmann und Thomas Hoeren. *Cloud-Computing für Unternehmen*. 2012. ISBN: 9783898648080.
- [70] *Was ist AWS? Sicheres Cloud Computing mit Amazon Web Services (AWS)*. <https://aws.amazon.com/de/what-is-aws/>. Accessed: 2022-11-07.
- [71] *Zertifizierung - Hetzner Online GmbH*. <https://www.hetzner.com/de/unternehmen/zertifizierung/>. Accessed: 2022-11-09.