



FOM Hochschule für Ökonomie & Management

Hochschulzentrum Frankfurt am Main

Master Thesis

im Studiengang IT-Management

zur Erlangung des Grades eines

Master of Science (M.Sc.)

über das Thema

**Architektonischer Entwurf und Modellierung einer Cloud Native Plattform für
den Einsatz von containerisierten Microservices und Anwendungen**

von

Dominik Otte

Betreuer : Dr. phil. Patrick Hedfeld

Matrikelnummer : 585039

Abgabedatum : 29. August 2023

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Abkürzungsverzeichnis	VI
Symbolverzeichnis	VII
1 Einleitung	1
1.1 Ein Medienecho	1
1.2 Problemstellung und Zielsetzung der Thesis	2
1.3 Abgrenzung	4
1.4 Methodik und Vorgehensweise	5
2 Softwarearchitektur	6
2.1 Begriffsdefinition	6
2.2 Architekturmodelle	6
2.2.1 3-Tier Architektur	6
2.2.2 Monolith	6
2.2.3 Microservices	6
2.3 Container	6
2.4 Automation und Orchestration	6
3 Cloud Computing	7
3.1 Begriffsdefinition	7
3.2 Grundlagen der Cloud-Technologie	7
3.3 Cloud Native Plattform	7
3.4 Kubernetes	7
4 Application Observability	8
4.1 Begriffsdefinition	8
4.2 Grundbausteine der Application Observability	8
4.2.1 Protokolle und Log-Dateien	8
4.2.2 Metriken und Performance Daten	8
4.2.3 Tracing	8
4.3 Herausforderungen in Cloud Native Umgebungen	8

5	Methodische Vorgehensweise	9
5.1	Auswahl Prototyping-Ansatz(Begründung Wahl der Methodik)	9
5.2	Identifizierung der Kernanforderungen an eine Cloud Native Plattform . . .	9
5.3	Planung und Design	9
6	Theoretischer Entwurf eines Modells	10
6.1	Ausprägungsmerkmale des Modells	10
6.2	Datenflussdiagramm	10
7	Prototypentwicklung aus dem theoretischen Modell	11
7.1	Aufbau der Laborumgebung	11
7.2	Iterative Prototyping Schleife (Auswahl der Tools)	11
7.3	Validierung und Test	11
8	Ergebnisse und Diskussion	12
8.1	Zusammenfassung der Ergebnisse	12
8.2	Handlungsempfehlung bei der Implementierung	12
9	Kritische Betrachtung	13
9.1	Limitation der angewandten Methodik	13
9.2	Limitation der Ergebnisse	13
9.3	Ausblick für künftige Forschungsarbeiten	13
10	Fazit und Ausblick	14
	Anhang	15
	Literaturverzeichnis	16

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

Symbolverzeichnis

1 Einleitung

Die fortschreitende Digitalisierung und die rasante Entwicklung von Technologien haben die Art und Weise, wie Softwareanwendungen entwickelt und bereitgestellt werden, grundlegend verändert. In der heutigen Arbeitswelt ist Agilität, Skalierbarkeit und Effizienz entscheidend für den Erfolg von Unternehmen. Aus diesem Grund haben sich neben zeitgemäßen Softwarearchitekturen und Technologien auch innovative Arbeitsmethoden und Vorgehensmodelle als grundlegende Elemente zeitgemäßer Softwareentwicklung und Bereitstellung etabliert.

Die Weltwirtschaft steht vor einer Ära, in der die nahtlose Bereitstellung von Diensten und Anwendungen unabhängig von räumlichen Einschränkungen von höchster Priorität ist. Die jüngsten Ereignisse, wie die globale COVID-19-Pandemie, haben die Notwendigkeit unterstrichen, dass Unternehmen agil und flexibel auf sich ändernde Marktbedingungen reagieren können. Laut einem Bericht von McKinsey hat die Pandemie die digitale Transformation beschleunigt und die Nachfrage nach Cloud-basierten Lösungen verstärkt, um remote zu arbeiten, Geschäftsprozesse umzugestalten und Kunden digital zu erreichen¹.

In dieser dynamischen Umgebung ist es von entscheidender Bedeutung, die Methoden und Ansätze zu verstehen, die bei der Gestaltung und Modellierung von Cloud Native Plattformen für containerisierte Anwendungen angewendet werden können. Die vorliegende Thesis widmet sich genau diesem Thema und beabsichtigt, einen tiefgreifenden Einblick in den architektonischen Entwurf und die Modellierung solcher Plattformen zu bieten. Indem sie aktuelle Nachrichten, Trends und bewährte Praktiken berücksichtigt, strebt die Thesis danach, einen Beitrag zur Weiterentwicklung dieser bedeutenden Technologie zu leisten und Unternehmen bei der Realisierung ihrer digitalen Visionen zu unterstützen.

1.1 Ein Medienecho

Der weitreichende und stetig wachsende Themenbereich der Cloud-Technologien hat über die letzten Jahre hinweg eine kontinuierliche Präsenz in einer Vielzahl von Medienkanälen aufrechterhalten, darunter Fachzeitschriften, Online-Blogs Podcasts sowie verschiedenste andere Formen von Medienplattformen und -Formaten. So behandelt der Artikel von David Linthicum eine skeptische Sicht auf eine von Gartner veröffentlichte Studie, in welcher prognostiziert wird dass bis 2025 mehr als 95% von Anwendungs-Workloads in einer Cloud

¹ Vgl. *McKinsey*, 2020, o.S.

Nativen Plattform laufen werden. Grundlegend möchte der Autor hervorheben, dass Unternehmen bei neuen Technologien neben Chancen auch die Risiken berücksichtigt sollen². So werden im Artikel drei Aspekte betont:

- Vendor Lock-in: Anwendungen die gezielt für eine bestimmte Cloud Plattformen entwickelt wurden, lassen sich schwieriger auf andere Plattformen übertragen. Die eingeschränkte Portabilität steht somit zum Teil im Widerspruch dessen was Cloud Native Anwendungen definiert.
- Skill Gap: Unternehmen ohne Erfahrung stehen vor Herausforderungen, die zusätzliche Schulungen oder Ressourcen erfordern, was zu schlecht konzipierten oder übermäßig komplexen Anwendungen führen kann. Dies wiederum könnte die Effizienz beeinträchtigen und möglicherweise die gesamte Umsetzung gefährden.
- unkontrollierter Kostenanstieg: Die nutzungsabhängige Preisgestaltung kann zu unvorhergesehenen Mehrkosten führen, wenn Anwendungen plötzlich stark frequentiert werden.

Pokemon Go, ein Spiel welches 2016 für Android und iOS Geräte erschien also für Smartphones und Tablets, setzt ebenfalls auf die Cloud Technologie von Google. Im News Blog von Google wird dargelegt wie es den Entwicklern möglich war mithilfe der bereitgestellten Cloud Technologie Live-Events im Spiel mit einem Transaktionsvolumen von 400.000 bis fast zu einer Millionen Transaktionen pro Sekunde umzugehen. Weiterhin wird ausgeführt, dass im Backend der Infrastruktur Services flexibel, nach Bedarf skalieren. Täglich werden 5-10 Terabyte an Daten im Rahmen Datenanalysen verarbeitet. Außerdem wird hervorgehoben, dass die Stabilität und Gesundheit durch umfassendes Logging, Monitoring und umfangreiche Dashboards sichergestellt wird³.

1.2 Problemstellung und Zielsetzung der Thesis

Die rasante Entwicklung von Cloud-Technologien und die wachsende Nachfrage nach agilen und skalierbaren Anwendungen haben zu einer Revolution in der Art und Weise geführt, wie Unternehmen ihre Software entwickeln, bereitstellen und betreiben. Die traditionelle monolithische Architektur wird zunehmend von einer auf Microservices basierenden Architektur abgelöst, die es ermöglicht, Anwendungen in kleinere, eigenständige Komponenten zu zerlegen. Diese Komponenten, auch als containerisierte Microservices

² Vgl. Linthicum, D., 2023, o.S.

³ Vgl. Priyanka Vergadia, J. P., 2021, o.S.

bezeichnet, können unabhängig voneinander entwickelt, bereitgestellt und skaliert werden. Die Cloud bietet eine ideale Umgebung für den Einsatz solcher Microservices, da sie elastische Ressourcen, automatische Skalierung und flexible Bereitstellungsmöglichkeiten bietet.

Der Einsatz von containerisierten Microservices in der Cloud bringt jedoch auch neue Herausforderungen mit sich. Eine entscheidende Frage ist die Auswahl geeigneter Technologien und Tools, um die Microservices effizient zu verwalten und zu orchestrieren. Container-Orchestrierungssysteme wie Kubernetes haben sich als Standard etabliert, bieten zudem eine Vielzahl von Möglichkeiten und Konfigurationen, die sorgfältig abgewogen werden müssen. Die Gewährleistung von Skalierbarkeit ist ein weiterer wichtiger Aspekt. Eine erfolgreiche Cloud-Native-Plattform muss in der Lage sein, die Anwendungen dynamisch zu skalieren, um eine hohe Auslastung zu bewältigen und gleichzeitig Ressourcenverschwendung zu vermeiden.

Darüber hinaus dürfen die Sicherheitsaspekte nicht vernachlässigt werden. Eine Cloud-Native-Plattform muss Mechanismen zur Isolation und Absicherung der einzelnen Microservices bereitstellen, um die Vertraulichkeit, Integrität und Verfügbarkeit der Daten zu gewährleisten. Ebenso spielt die effiziente Ressourcennutzung eine zentrale Rolle, da eine effektive Verwaltung und Auslastung der Ressourcen in der Cloud entscheidend für die Wirtschaftlichkeit und Leistungsfähigkeit der Plattform ist.

Vor diesem Hintergrund wird deutlich, dass der architektonische Entwurf und die Modellierung einer geeigneten Cloud-Native-Plattform von entscheidender Bedeutung sind. Eine fundierte und gut durchdachte Architektur legt den Grundstein für den erfolgreichen Einsatz von containerisierten Microservices und Anwendungen in der Cloud. Durch die Schaffung einer skalierbaren, sicheren und effizienten Plattform können Unternehmen die Vorteile der Cloud voll ausschöpfen und ihre Anwendungen mit Agilität und Skalierbarkeit entwickeln und betreiben.

Das Ziel dieser Arbeit ist es, einen umfassenden architektonischen Entwurf und eine beispielhafte Modellierung einer Cloud-Native-Plattform für den erfolgreichen Einsatz von containerisierten Microservices und Anwendungen zu entwickeln. Dabei sollen nicht nur die Anforderungen an Skalierbarkeit und effiziente Ressourcennutzung berücksichtigt werden, sondern auch weitere wichtige Aspekte, die für den reibungslosen Betrieb der Plattform von Bedeutung sind.

Ein zentraler Fokus liegt auf der Auswahl geeigneter Technologien, die eine nahtlose Integration und Interaktion der Microservices ermöglichen. Hierbei werden verschiedene

Aspekte berücksichtigt, wie die Containerisierungstechnologie (z. B. Docker), die Orchestrierung (z. B. Kubernetes), das Service-Discovery-Management und die Konfigurationsverwaltung. Die Auswahl der richtigen Technologien spielt eine entscheidende Rolle bei der Gewährleistung von Skalierbarkeit, Flexibilität und Wartbarkeit der Plattform.

Des Weiteren wird die Integration von Monitoring-, Logging- und Tracing-Funktionalitäten in den architektonischen Entwurf und die Modellierung einbezogen. Dies ist von großer Bedeutung, um eine umfassende Überwachung und Analyse der Microservices und deren Kommunikation zu ermöglichen. Durch die Implementierung dieser Funktionen können Performance-Probleme, Engpässe und Fehler frühzeitig erkannt und behoben werden.

Darüber hinaus wird in der Arbeit auch der Sicherheitsaspekt berücksichtigt. Die Architektur und Modellierung der Cloud-Native-Plattform müssen Mechanismen zur Sicherung der Datenintegrität, zum Schutz vor unbefugtem Zugriff und zur Abwehr potenzieller Bedrohungen umfassen. Hierbei werden verschiedene Sicherheitsmaßnahmen wie Authentifizierung, Autorisierung und Verschlüsselung in den Entwurf integriert.

Durch den erarbeiteten architektonischen Entwurf und die Modellierung einer Cloud-Native-Plattform wird eine solide Grundlage geschaffen, auf der Unternehmen aufbauen können, um die Vorteile von containerisierten Microservices und Anwendungen in der Cloud optimal zu nutzen und ihre IT-Infrastruktur effektiv zu modernisieren. Die entwickelte Plattform ermöglicht eine effiziente Ressourcennutzung, Skalierbarkeit und Flexibilität und unterstützt Unternehmen dabei, ihre Anwendungen agil zu entwickeln, zu betreiben und auf zukünftige Anforderungen anzupassen.

[Zusammenfassung Problemstellung, Forschungsfrage, Zielsetzung der Thesis]

1.3 Abgrenzung

Die Festlegung der Abgrenzung der vorliegenden Thesis ist von entscheidender Bedeutung, um den Umfang der Arbeit klar zu definieren und den Fokus gezielt auf die themenspezifischen Aspekte des architektonischen Entwurfs und der Modellierung einer Cloud-Native-Plattform für containerisierte Microservices und Anwendungen zu lenken. Durch die klare Definition des Umfangs werden die Grenzen der Untersuchung im weitreichenden Cloud-Native Themenkomplex abgesteckt und die Aufmerksamkeit auf die relevanten Themenbereiche gelenkt, die im Kontext dieser Arbeit von Interesse sind.

- **Technologische Ausrichtung:** Die Hausarbeit konzentriert sich auf Cloud-Native-Technologien und -Ansätze, insbesondere auf die Verwendung von Containern und

Microservices. Andere Ansätze, wie beispielsweise virtuelle Maschinen oder herkömmliche monolithische Architekturen, werden nicht im Detail behandelt.

- **Plattformfokus:** Die Arbeit legt den Schwerpunkt auf die Entwicklung einer Cloud-Native-Plattform, die den spezifischen Anforderungen von containerisierten Microservices und Anwendungen gerecht wird. Dabei werden Aspekte wie Architekturdesign, Auswahl geeigneter Technologien und Integration von Monitoring-, Logging- und Tracing-Funktionalitäten berücksichtigt. Die Implementierung und konkrete Umsetzung der Plattform in einer bestimmten Cloud-Umgebung oder mit spezifischen Tools wird jedoch nicht im Detail behandelt.
- **Anwendungsbereich:** Die Hausarbeit fokussiert sich auf den generellen architektonischen Entwurf und die Modellierung einer Cloud-Native-Plattform für containerisierte Microservices und Anwendungen. Es werden keine spezifischen Anwendungsdomänen oder Industrien betrachtet. Die vorgeschlagene Architektur und Modellierung sollten jedoch auf verschiedene Anwendungsfälle und Branchen anwendbar sein.
- **Zeitliche Betrachtung:** Die Hausarbeit bezieht sich auf den aktuellen Stand der Technologie und Best Practices zum Zeitpunkt der Erstellung der Arbeit. Zukünftige Entwicklungen oder Trends im Bereich der Cloud-Native-Architektur und Containerisierung können nicht berücksichtigt werden.

[exkludiert sind Managed Services wie (AWS, GKE, etc.) Zusammenfassung der Abgrenzung]

1.4 Methodik und Vorgehensweise

Literaturrecherche, Modellentwicklung auf Basis der Literatur, Anforderungsanalyse, Prototyp Entwicklung in Form einer praxis Implementierung

2 Softwarearchitektur

2.1 Begriffsdefinition

2.2 Architekturmodelle

2.2.1 3-Tier Architektur

2.2.2 Monolith

2.2.3 Microservices

2.3 Container

2.4 Automation und Orchestration

3 Cloud Computing

3.1 Begriffsdefinition

3.2 Grundlagen der Cloud-Technologie

3.3 Cloud Native Plattform

3.4 Kubernetes

4 Application Observability

4.1 Begriffsdefinition

4.2 Grundbausteine der Application Observability

4.2.1 Protokolle und Log-Dateien

4.2.2 Metriken und Performance Daten

4.2.3 Tracing

4.3 Herausforderungen in Cloud Native Umgebungen

5 Methodische Vorgehensweise

5.1 Auswahl Prototyping-Ansatz(Begründung Wahl der Methodik)

5.2 Identifizierung der Kernanforderungen an eine Cloud Native Plattform

5.3 Planung und Design

6 Theoretischer Entwurf eines Modells

6.1 Ausprägungsmerkmale des Modells

6.2 Datenflussdiagramm

7 Prototypentwicklung aus dem theoretischen Modell

7.1 Aufbau der Laborumgebung

7.2 Iterative Prototyping Schleife (Auswahl der Tools)

7.3 Validierung und Test

8 Ergebnisse und Diskussion

8.1 Zusammenfassung der Ergebnisse

8.2 Handlungsempfehlung bei der Implementierung

9 Kritische Betrachtung

9.1 Limitation der angewandten Methodik

9.2 Limitation der Ergebnisse

9.3 Ausblick für künftige Forschungsarbeiten

10 Fazit und Ausblick

Anhang

Literaturverzeichnis

Linthicum, David (2023): Drei Nachteile: Was gegen Cloud-Native spricht - computerwoche.de, o. O., 2023-05-24, URL: <https://www.computerwoche.de/a/was-gegen-cloud-native-spricht,3613754> [Zugriff: 2023-08-29]

McKinsey (2020): COVID-19 digital transformation & technology | McKinsey, o. O., 2020, URL: <https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/how-covid-19-has-pushed-companies-over-the-technology-tipping-point-and-transformed-business-forever/> [Zugriff: 2023-08-22]

Priyanka Vergadia, James Prompanya (2021): How Pokémon GO scales to millions of requests? | Google Cloud Blog, o. O., 2021-10-27, URL: <https://cloud.google.com/blog/topics/developers-practitioners/how-pok%C3%A9mon-go-scales-millions-requests?hl=en> [Zugriff: 2023-08-29]

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass die vorliegende Arbeit von mir selbstständig und ohne unerlaubte Hilfe angefertigt worden ist, insbesondere dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen sind, durch Zitate als solche gekennzeichnet habe. Ich versichere auch, dass die von mir eingereichte schriftliche Version mit der digitalen Version übereinstimmt. Weiterhin erkläre ich, dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde/Prüfungsstelle vorgelegen hat. Ich erkläre mich damit einverstanden, dass die Arbeit der Öffentlichkeit zugänglich gemacht wird. Ich erkläre mich damit einverstanden, dass die Digitalversion dieser Arbeit zwecks Plagiatsprüfung auf die Server externer Anbieter hochgeladen werden darf. Die Plagiatsprüfung stellt keine Zurverfügungstellung für die Öffentlichkeit dar.

Mörfelden-Walldorf, 29.8.2023

(Ort, Datum)

A handwritten signature in black ink, consisting of a large, stylized 'D' followed by a series of loops and a long horizontal stroke extending to the right.

(Dominik Otte)