



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 : 6. September 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	4
2 Softwarearchitektur	5
2.1 Begriffsdefinition	5
2.2 Architekturmodelle	5
2.2.1 3-Tier Architektur	5
2.2.2 Monolith	5
2.2.3 Microservices	5
2.3 Container	5
2.4 Automation und Orchestration	5
3 Cloud Computing	6
3.1 Begriffsdefinition	6
3.2 Grundlagen der Cloud-Technologie	6
3.3 Cloud Native Plattform	6
3.4 Kubernetes	6
4 Application Observability	7
4.1 Begriffsdefinition	7
4.2 Grundbausteine der Application Observability	7
4.2.1 Protokolle und Log-Dateien	7
4.2.2 Metriken und Performance Daten	7
4.2.3 Tracing	7
4.3 Herausforderungen in Cloud Native Umgebungen	7

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

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

CPU	Central Processing Unit
PaaS	Platform-as-a-Service
RAM	Random-Access Memory
SaaS	Software-as-a-Service

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 aus dem HomeOffice heraus zu arbeiten, Geschäftsprozesse umzugestalten und Kunden digital zu erreichen¹.

In einer dynamischen Cloud Umgebung ist es von signifikanter Relevanz, 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 Marktaktuellen Technologie zu leisten und Leser 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-Native Plattform laufen werden. Grundlegend möchte der Autor hervorheben, dass

¹ Vgl. *McKinsey*, 2020, o.S.

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 traditionelle monolithische Anwendungsarchitektur wird zunehmend von einer auf Microservices⁴ basierenden Architektur abgelöst, die es ermöglicht, Anwendungen in kleinere, eigenständige Komponenten zu zerlegen. Diese als containerisierte Microservices bezeichneten Software Komponenten, können unabhängig voneinander entwickelt, bereitgestellt und skaliert werden. Cloudlösungen werben damit, eine ideale Umgebung für den Einsatz solcher Microservices zu bieten, da sie unter anderem Ressourcen elastische bereitstellen und automatisch skalieren können⁵.

² Vgl. Linthicum, D., 2023, o.S.

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

⁴ Vgl. Wolff, E., 2018, S.4.

⁵ Vgl. Henneberger, M., 2016, S.8-19.

Der Einsatz von containerisierten Microservices in der Cloud bringt jedoch auch neue Herausforderungen mit sich. Wie das Wort Microservice bereits indiziert, liegt die Vermutung nahe, dass in einer Anwendungslandschaft zahlreiche weitere Microservices existieren, die als Konglomerat einen ganzheitlichen Service bereitstellen. Für den Betrieb im Produktiven Umfeld bedeutet dementsprechend, dass der Monitoring aufwand für Microservices steigt. Eine entscheidende Frage ist die Auswahl geeigneter Technologien und Tools, um die Microservices effizient zu verwalten und orchestrieren⁶. Container-Orchestrierungssysteme wie Kubernetes haben sich als Industriestandard etabliert und bieten zudem eine Vielzahl an Konfigurationsmöglichkeiten. Die Gewährleistung von Skalierbarkeit ist ein weiterer essenzieller Aspekt im Cloud Kontext. Um dynamische Workloads effizient zu bewältigen und Ressourcenverschwendung zu vermeiden, muss eine Cloud-Native Plattform in der Lage sein, Anwendungen flexibel zu skalieren. Eine effektive Verwaltung und Auslastung der Ressourcen in der Cloud sind entscheidend für die Wirtschaftlichkeit und Leistungsfähigkeit der Plattform⁷.

Die Thesis beschäftigt sich mit der Forschungsfrage: „Wie kann ein architektonischer Entwurf für eine Cloud-Native Plattform aussehen um den neuen Herausforderungen in einer Cloud-Native Umgebung zu begegnen“? Hauptaugenmerk liegt im Vergleich zu klassischen Architekturen nicht auf der Hardware Infrastruktur und der Ressourcen Ausstattung wie Central Processing Unit (CPU), Random-Access Memory (RAM) und Speicherplatz, sondern Aspekte im Sinne der Application Observability, die für den reibungslosen Betrieb der Plattform und den darauf ansässigen Anwendungen von Belang sind. Weiterhin betrachtet die Thesis eine eigenständig administrierte, on-Premise betriebene Plattform und keinen Dienstleistungsbezug von einem Service Provider in Form einer Software-as-a-Service (SaaS) oder Platform-as-a-Service (PaaS) Lösung.

Das vorrangige Ziel dieser Forschungsarbeit besteht darin, eine eingehende Untersuchung des Themenfelds Application Observability im Kontext von Cloud-Native Anwendungen durchzuführen, wobei besonderes Augenmerk auf die Identifizierung und Analyse relevanter Aspekte gelegt wird, die für den effizienten Betrieb von Anwendungen relevant sind. Darüber hinaus beabsichtigt diese Arbeit, die gewonnenen Erkenntnisse und Einsichten in ein umfassendes Architekturmodell zu überführen, das als Leitfaden und Rahmenwerk für die Implementierung und das Management von Application Observability in Cloud-Native Umgebungen dienen soll.

⁶ Vgl. *Wolff, E.*, 2018, S.12-16.

⁷ Vgl. *Kubernetes*, 2022, o.S.

1.3 Abgrenzung

Die Herausforderungen im Bereich der Application Observability von Cloud-Native Anwendungen sind vielfältig. Angesichts dieser Komplexität ist es unabdingbar, eine klare Abgrenzung und Fokussierung für die vorliegende Arbeit festzulegen. Die Themen, die im Rahmen dieser Thesis im Kontext der Application Observability untersucht werden sind⁸:

- **Monitoring:** Die Erfassung von Metriken von Hardware nahen Ressourcen wie CPU und RAM. Aber auch anwendungsspezifische Metriken wie den Status von Diensten und Echtzeitinformationen der Anwendung. Die Thesis wird sich darauf konzentrieren, wie Monitoring-Lösungen in Cloud-Native Umgebungen implementiert werden können, um einen umfassenden Einblick in die Gesundheit der Anwendungen zu gewährleisten.
- **Tracing:** Tracing von Anfragen oder Transaktionen durch verschiedene Microservices in Cloud-Native Anwendungen wird betrieben um Engpässe und Latenzprobleme zu identifizieren. Diese Arbeit wird die Konzepte und Tools zur Tracing-Implementierung in Cloud-Native Umgebungen beleuchten und zeigen, wie sie dazu beitragen, die Leistung und Zuverlässigkeit der Anwendungen zu verbessern.
- **Logging:** Das Sammeln und Analysieren von Logs ist ein wesentlicher Bestandteil der Application Observability. Logs liefern Einblicke in Fehler, Ausnahmen und Aktivitäten in Anwendungen. Die Thesis wird sich mit bewährten Praktiken für das Log-Management in Cloud-Native Anwendungen befassen, einschließlich der Strukturierung von Logdaten und deren Integration in zentrale Log-Aggregationsplattformen.
- **Telemetriedaten:** Telemetriedaten umfassen eine Vielzahl von Informationen, die zur Verbesserung der Anwendungsleistung beitragen können. Dies schließt zum Beispiel Netzwerkkommunikation oder, Datenbankzugriffe ein. Die Arbeit wird die Erfassung von Telemetriedaten in Form der Netzwerkkommunikation in Cloud-Native Umgebungen behandeln und visualisieren.

Des weiteren werden Dienstleistungen großer Cloud Plattformanbieter nicht betrachtet.

1.4 Methodik und Vorgehensweise

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

⁸ Vgl. *Pourmajidi, W. et al.*, 2023, S.5.

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

Begriffsdefinition:⁹

3.4 Kubernetes

⁹ Vgl. *Kratzke, N.*, 2021, S. 33-34.

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

Henneberger, Matthias (2016): Von „Cloud Enabling“ zu „Cloud Native“: Wie Cloud Computing die Unternehmens-IT verändert, in: Wirtschaftsinformatik & Management, 8 (2016), S. 8–19

Kratzke, Nane (2021): Cloud-native Computing: Software Engineering von Diensten und Applikationen für die Cloud, o. O.: Carl Hanser Verlag GmbH Co KG, 2021

Pourmajidi, William, Zhang, Lei, Steinbacher, John, Erwin, Tony, Miranskyy, Andriy (2023): A Reference Architecture for Observability and Compliance of Cloud Native Applications, o. O., 2023, arXiv: 2302.11617 [cs.SE]

Wolff, Eberhard (2018): Das Microservices-Praxisbuch: Grundlagen, Konzepte und Rezepte, o. O.: dpunkt. verlag, 2018

Internetquellen

Kubernetes (2022): Was ist Kubernetes? | Kubernetes, <<https://kubernetes.io/de/docs/concepts/overview/what-is-kubernetes/>> (2022) [Zugriff: 2023-09-03]

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

McKinsey (2020): COVID-19 digital transformation & technology | McKinsey, <<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/>> (2020) [Zugriff: 2023-08-22]

Priyanka Vergadia, James Prompanya (2021): How Pokémon GO scales to millions of requests? | Google Cloud Blog, <<https://cloud.google.com/blog/topics/developers-practitioners/how-pok%C3%A9mon-go-scales-millions-requests?hl=en>> (2021-10-27) [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, 6.9.2023

(Ort, Datum)

A handwritten signature in black ink, consisting of stylized, flowing letters that appear to be 'D. Otte'.

(Dominik Otte)