

Exposé

**Welchen Vorteil bietet der Einsatz von Microservices
in Kombination mit REST-APIs im Vergleich zu Monolithen?**

Zur Veranstaltung Programmieren
im Studiengang Informatik

vorgelegt dem Fachbereich Informatik und Wirtschaftsinformatik / Wirtschaftswissenschaften
der Provadis School of International Management and Technology
von

Hanna Förster
D971
BIN-T24 Gruppe 2
hannafoerster25@gmail.com

Erstgutachter: Prof. Dr. Grebner

Zweitgutachter: -

Ende der Bearbeitungsfrist: 31.01.2025

Inhalt

1	Thema und Themenrelevanz	2
2	Leitfrage der Arbeit, mögliche Antworten und mögliche Hypothesen	2
3	Kurzbeschreibung des Praxisfall	3
4	Methoden.....	3
5	Benötigte theoretische Grundlagen/ zentrale Literaturquellen	4
6	Gliederungsentwurf.....	5
	Literaturverzeichnis	6
	KI-Verzeichnis	8

1 Thema und Themenrelevanz

Die Monolithische Architektur, also ein eng gekoppelter Code, beziehungsweise ein Code, bei dem komplexe, kombinierte Aufgaben bearbeitet werden, welcher z.B. funktionale und nicht funktionale Anwendungsanforderungen erfüllt, wurde und wird in der Informatik immer noch verwendet.¹

Modulare Architekturen sind entstanden, um die Komplexität der monolithischen Architektur zu erleichtern und besteht aus lose gekoppelten Modulen. Beispiele für eine modulare Architektur sind Funktionen und Services.²

Microservices, eine Art der modularen Architektur, hingegen sind Programme mit nur einer (der kleinstmöglichen) Funktion.³ Außerdem erfüllen sie in der Regel nur eine einzige Aufgabe. Um Microservices richtig verwenden zu können müssen diese in Kombination mit APIs (Application Programm Interface), also Schnittstellen zwischen den einzelnen Services, verwendet werden. Ein Beispiel einer API ist das “Representational State Transfer” (REST).

REST APIs bezeichnen verschiedene Regeln und Richtlinien, die festlegen, wie eine Web-API aufgebaut sein soll.”⁴

Durch die Fähigkeit monolithische Architekturprobleme, wie begrenzte Skalierbarkeit, schwere Wartbarkeit und technologische Abhängigkeit, zu adressieren, wird die Microservice Architektur immer populärer. Dennoch ist die Migration von einer monolithischen Architektur zu einer microservice Architektur komplex. Daher stellt sich die Frage, ob es wirklich sinnvoll ist, zu migrieren.⁵

2 Leitfrage der Arbeit, mögliche Antworten und mögliche Hypothesen

Leitfrage:

“Welche Vorteile bietet der Einsatz von Microservices in Kombination mit Rest-APIs gegenüber Monolithen?”

Hypothesen:

¹ Vgl. Yousif 2016

² Vgl. Yousif 2016

³ Vgl. Yousif 2016

⁴ RESTful APIs (o.D.)

⁵ Vgl. Saucedo, Rodríguez, Rocha, Santos 2024

- Microservices in Kombination mit Rest-APIs bieten weitaus mehr Vorteile gegenüber Monolithen
- Microservices bieten kaum oder gar keine Vorteile gegenüber Monolithen
- Microservices bieten mehr Nachteile als Vorteile gegenüber Monolithen

3 Kurzbeschreibung des Praxisfall

Zur Planung des Netzausbaus der Deutschen Telekom AG (DTAG) werden verschiedene Applikationen sowie Tools angeboten, die diesen vereinfachen.

Ein Programm, Plural TNP (Plural), bietet so eine Erleichterung. Mit diesem “Planungssystem” werden Kabelwege, Geräte und Übertragungswege geplant und dokumentiert. Plural wurde mit dem System CaGen von IBM⁶ entwickelt und besitzt eine monolithische Architektur.

Dieses Programm ist veraltet, kompliziert zu bedienen und verwendet ein System, welches die Entwicklung eines Pseudocodes, der dann zu C Code generiert wird, ermöglicht. Dadurch entstehen Herausforderungen, wie das, durch den demographischen Wandel, verlassen der Kollegen, welche das System verstehen, sowie, das CA Gen kaum noch verwendet wird. Dadurch wurde entschieden dieses veraltete Programm von einem neuen System abgelöst.

Dieses ablösende System, unter dem Namen Plandok, wird mit der modernen Programmiersprache Java mit dem Quarkus Framework⁷ entwickelt und besitzt eine Microservice-orientierte Architektur, welche in Verbindung mit REST-APIs verwendet wird. Dadurch bietet Plandok zusätzlich zu der Planung und Dokumentation ein übersichtliches Graphical User Interface (GUI) und eine erleichterte Erweiterung des Programms.

4 Methoden

Zur Beantwortung der Leitfrage wird die wissenschaftliche Methode der konzeptionell-deduktiven Analyse in Kombination mit dem Prototyping angewandt.^{8 9}

Zunächst werden wesentliche Merkmale zu Monolithen sowie Microservices recherchiert, um die Herausarbeitung verschiedener, für die Evaluation wichtiger Kriterien einzugrenzen.

⁶ Vgl. CA Gen 2023

⁷ Quarkus documentation

⁸ Vgl. Wilde, Hess 2006

⁹ Vgl. Houde, Hill 1997

Im weiteren Verlauf wird die praktische Umsetzung eines Prototyps, hierbei der Microservice in Kombination mit REST-APIs welches mit Java programmiert wird und auf dem Quarkus Framework entsteht, umgesetzt.

Außerdem wird für die Evaluation eine beispielhafte Verwendung des Monolithen angewandt, hier eine beispielhafte Verwendung von Plural.

Im Anschluss wird auf den erarbeiteten Grundlagen, eine Kriterien geleitete, konzeptionell-deduktive Analyse durchgeführt und eine Zusammenfassung der Evaluation verfasst.

Die verwendete Methode soll zu einem klar verständlichen und geeigneten Vergleich, welcher anhand des Implementierten REST-Services und der herangezogenen monolithischen Funktion stattfindet, beitragen.

5 Benötigte theoretische Grundlagen/ zentrale Literaturquellen

Die Methode des Prototyping sowie die Methode der deduktiven Analyse sind hinlänglich untersucht und etablierte Herangehensweisen in der Informatik.^{10 11}

Die Monolithische, sowie die Microservice Architektur sind weit verbreitete und viel verwendete Architekturen in der Informatik.¹²

Für die Modellierung des Prototyps wird Java 17 verwendet. Außerdem wird das Quarkus Framework unter Beachtung der Dokumentation verwendet.¹³ Weitergehend wird eine PostgreSQL Datenbank verwendet, hierbei unter der Beachtung der PostgreSQL Dokumentation.¹⁴

Die Funktionsweise von Rest-APIs ist weitgehend dokumentiert, dies unterstützt in der Implementierung.¹⁵

Der Monolith wurde unter Verwendung einer Oracle Datenbank und der Programmiersprache CaGen entwickelt.^{16 17}

¹⁰ Vgl. Carr, Verner 1997

¹¹ Vgl. Wilde, Hess 2006

¹² Vgl. Kushtagi 2024

¹³ Quarkus documentation

¹⁴ PostgreSQL documentation

¹⁵ Understanding Rest APIs

¹⁶ Oracle documentation

¹⁷ CaGen documentation

6 Gliederungsentwurf

1. Einleitung

1.1 Einführung in das Thema

1.1.1 Problemstellung

1.1.2 Forschungsfrage und Hypothesen

1.1.3 Ziel der Arbeit

1.2 Relevanz des Projekts für den Betrieb

1.3 Methodik und Vorgehensweise

2. Grundlagen

2.1 Architekturen (Monolith und Microservices)

2.2 Software-Intermediär (REST-APIs)

2.3 Frameworks (Quarkus)

3. Bearbeitung beider Architekturen (Prototyping)

3.1 Recherche

3.2 Microservices

3.2.1 Implementierung

3.2.1.1 Planung

3.2.1.2 Umsetzung

3.3 Monolithen

3.3.1 beispielhafte Verwendung

4. konzeptionell-deduktive Analyse

5. Zusammenfassung und Ausblick

6. Literaturverzeichnis

7. Anhang

Literaturverzeichnis

Again, M. C. S. G. (2024, 1. September). Understanding REST API: A Comprehensive guide - Make Computer Science Great Again - Medium. Medium. <https://medium.com/@MakeComputerScienceGreatAgain/understanding-rest-api-a-comprehensive-guide-52fc10f6c9ed>

GeeksforGeeks. (2025, 3. Januar). Monolithic vs. Microservices Architecture. GeeksforGeeks. <https://www.geeksforgeeks.org/monolithic-vs-microservices-architecture/>

Kushtagi, R. (2024, 24. November). System Design Secrets: Monolith vs. Microservices Explained. Medium. <https://medium.com/@roopa.kushtagi/system-design-trade-offs-monolithic-vs-microservices-architecture-1e14a9fe9e99>

@Harsh. (2024, 15. November). Understanding Monolithic Architecture and the Shift to Microservices. Medium. <https://harsh05.medium.com/understanding-monolithic-architecture-and-the-shift-to-microservices-080a1768273c>

Al-Debagy, O., & Martinek, P. (2018, November). A comparative review of microservices and monolithic architectures. In 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI) (pp. 000149-000154). IEEE. <https://ieeexplore.ieee.org/abstract/document/8928192/>

H, J. (2024, 11. Juni). REST APIs vs Microservices: Key differences. Dreamfactory. <https://blog.dreamfactory.com/restful-api-and-microservices-the-differences-and-how-they-work-together>

IEEE Xplore Full-Text PDF: (o. D.). <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10160171>

Seobility. (o. D.). Was ist eine REST API? Definition und Erklärung - Seobility Wiki. <https://www.seobility.net/de/wiki/REST-API>

Houde, S., & Hill, C. (1997). What do prototypes prototype?. In Handbook of human-computer interaction (pp. 367-381). North-Holland.

https://courses.cs.washington.edu/courses/cse440/08au/readings_files/Houde-Prototypes.pdf

Wilde, T., & Hess, T. (2006). Methodenspektrum der Wirtschaftsinformatik: Überblick und Portfoliobildung (No. 2/2006). Arbeitsbericht.

<https://www.econstor.eu/bitstream/10419/60077/1/71972564X.pdf>

Carr, M., & Verner, J. (1997). Prototyping and software development approaches. Department of Information Systems, City University of Hong Kong, Hong Kong, 319-338.

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=0b05add730e04843e234937a070f24b19efaadc3>

Quarkus documentation content types. (n.d.). Quarkus. <https://quarkus.io/guides/doc-concept>

PostgreSQL: documentation. (n.d.). The PostgreSQL Global Development Group. <https://www.postgresql.org/docs/>

Database documentation - Database documentation. (2024, October 22). Oracle Help Center. <https://docs.oracle.com/en/database/>

Inc, C. (2023, May 4). CA gen. https://techdocs.broadcom.com/us/en/ca-miscellaneous/legacy_bookshelves_and_pdfs/bookshelves_and_pdfs/bookshelves/ca-gen.html

Yousif, M. (2016). Microservices. IEEE Cloud Computing, 3(5), 4-5. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7742262>

Was ist eine REST API? – Einführung in RESTful APIs. (o. D.). <https://www.redhat.com/de/topics/api/what-is-a-rest-api>

Saucedo, A. M., Rodríguez, G., Rocha, F. G. & Santos, R. P. D. (2024). Migration of monolithic systems to microservices: A systematic mapping study. Information And Software Technology, 107590. <https://doi.org/10.1016/j.infsof.2024.107590>

Inc, C. (2023, May 4). CA gen. https://techdocs.broadcom.com/us/en/ca-miscellaneous/legacy_bookshelves_and_pdfs/bookshelves_and_pdfs/bookshelves/ca-gen.html

KI-Verzeichnis