

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: 10.01.2025

Inhalt

1	Forschungsfrage und Themenrelevanz.....	1
2	Praxisfall und Zielsetzung.....	1
3	Methodisches Vorgehen bei der Untersuchung	1
4	Durchführung der Untersuchung	2
5	Theoretische Grundlagen und vorläufige Literaturquellen	3
6	Gliederungsentwurf	3
7	Literaturverzeichnis	3
8	KI-Verzeichnis	4

1 Forschungsfrage und Themenrelevanz

Die Migration von Monolithen auf Microservices wird zu einem immer beliebteren Thema. Aber was macht Microservices und die entsprechend verwendeten APIs so begehrt und welche Vorteile bietet die Migration? Mit der Frage: "Welche Vorteile bietet der Einsatz von Microservices in Kombination mit Rest-APIs gegenüber Monolithen", wird herausgestellt, was mögliche Gründe sind, warum man von einer monolithischen Architektur zu einer Microservice Architektur wechselt und ob es nicht doch mehr Vorteile gibt, wenn man bei der monolithischen Architektur bleibt.

2 Praxisfall und Zielsetzung

In dem Betriebseinsatz wird ein veraltetes Programm, was den Namen "Plural TNP" (wird im folgenden als Plural geschrieben) hat, von Grund auf erneuert. Plural ist ein "Planningtool" zur Dokumentation von Kabelwegen, Geräten, Anschlüssen und Ü-Wegen (Übertragungswege), dass mit der Programmiersprache CaGen (von Broadcom) geschrieben wurde und eine monolithische Architektur verwendet. Plural, also ein das monolithische Programm, wird hierbei in eine Service-orientierte Architektur transformiert und unter dem neuen Namen "Plandok" als neues System abgelöst. Dieses kümmert sich genau wie Plural um die Dokumentation verschiedener Komponenten, verwendet aber eine Microservice-Architektur. Plandok soll im Gegensatz zu Plural übersichtlicher und einfacher zu bedienen sein. Außerdem soll es für zukünftige Generationen einfacher werden Plandok erweitern zu können, da man von CaGen (von IBM) zu Java mit Quarkus Framework wechselt. Am Ende der Arbeit sollen sich Gründe für den Wechsel und weitere Vorteile einer Microservice-Architektur gegenüber einer monolithischen Architektur herausstellen.

3 Methodisches Vorgehen bei der Untersuchung

Um die Zielsetzung einhalten zu können müssen zuerst Kriterien, welche den Vergleich der beiden Architekturen leiten werden, definiert werden. Ein Beispiel hierfür ist Modularität.¹ Dazu wird zuerst nach möglichst wichtigen Kriterien recherchiert, damit man das möglichst beste Ergebnis erzielt. Um die Kriterien auf die Architekturen anwenden zu können, müssen

¹ <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10160171>

auch hier Informationen über die jeweilige Architektur herausgesucht werden. Dazu werden beide für Architekturen praktische Beispiele verwendet, wobei der Monolith schon existiert und der Microservice als Programmierprojekt geschrieben wird. Dass der Monolith nicht auch implementiert wird, liegt daran, dass die Zeit, die es dafür benötigt, nicht ausreichen wird. Zum Schluss werden die einzelne, nach den Kriterien sortierte Informationen, beider Architekturen verglichen.

4 Durchführung der Untersuchung

Zunächst wird eine Recherche zu Monolithen und zu Microservices durchgeführt. Dadurch folgt eine Herausarbeitung der verschiedenen Kriterien. Im weiteren Verlauf wird die praktische Umsetzung ausgeführt, also die Implementation des Microservices und die beispielhafte Verwendung des Monoliths. Im Anschluss wird auf Grundlagen der Kriterien eine Analyse der beiden Architekturen durchgeführt und zum Schluss wird eine Zusammenfassung sowie Evaluierung verfasst.

Projekt aufwand WAB [18 Tage]	Dez	Jan	Feb	Mär	Stundenarbeit	Tage pro Aufgabe
Thema finden (fachlich) und formulieren	5				5	1
Kriterien für den Vergleich heraussuchen			5		5	1
Programmierprojekt umsetzen		10	10		20	4
Recherche zu monolithischer Architektur		5	5		10	2
Literatur Recherche	2	5	3		10	1
Exposé schreiben	5	10			15	3
Wissenschaftlich ausgearbeiteten Bericht schreiben			10	15	25	5
Verbesserungsarbeit/Formulierung/...				5	5	1
Summe	15	30	30	20	95	18

5 Vorläufige Literaturquellen

<https://medium.com/@MakeComputerScienceGreatAgain/understanding-rest-api-a-comprehensive-guide-52fc10f6c9ed>

<https://www.geeksforgeeks.org/monolithic-vs-microservices-architecture>

<https://medium.com/@roopa.kushtagi/system-design-trade-offs-monolithic-vs-microservices-architecture-1e14a9fe9e99>

<https://harsh05.medium.com/understanding-monolithic-architecture-and-the-shift-to-microservices-080a1768273c>

<https://ieeexplore.ieee.org/abstract/document/8928192/>

<https://blog.dreamfactory.com/restful-api-and-microservices-the-differences-and-how-they-work-together>

<https://blog.dreamfactory.com/restful-api-and-microservices-the-differences-and-how-they-work-together>

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10160171>

<https://www.seobility.net/de/wiki/REST-API>

6 Gliederungsentwurf

1. Einleitung

1. Einführung in das Thema
 1. Problemstellung
 2. Forschungsfrage und Hypothesen
 3. Ziel der Arbeit
2. Relevanz des Projekts für den Betrieb
3. Methodik und Vorgehensweise

2. Grundlagen

1. Architekturen (Monolith und Microservices)
2. Software-Intermediär (REST-APIs)
3. Frameworks (Quarkus)

3. Bearbeitung beider Architekturen

1. Microservices

1. Implementierung

1. Planung

2. Umsetzung

2. Selection von Pro- und Kontra Punkten

2. Monolithen

1. Recherche

2. Selection von Pro- und Kontra Punkten

4. Vergleich

5. Zusammenfassung und Ausblick

6. Literaturverzeichnis

7. Anhang