

Es muss nicht immer Microservice sein: Architekturen im Vergleich

Kai und Thomas

13.11.2024 JUG Fra 2024

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

Name: Kai

Wo arbeitest du: selbständig

Hobbies:



Name: Thomas

Wo arbeitest du:



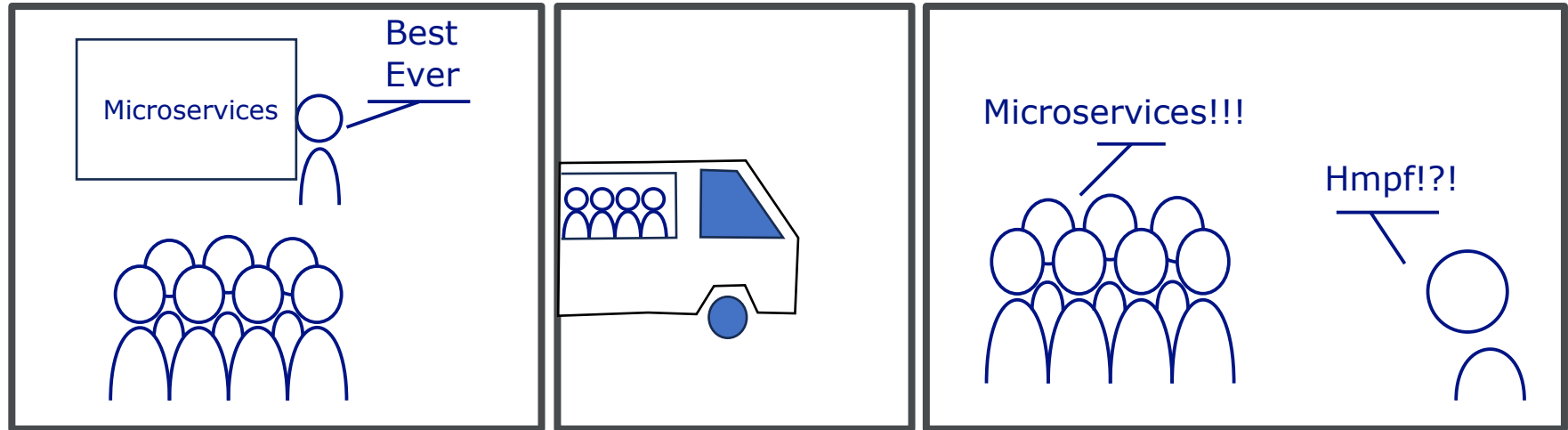
Hobbies:



Unsere Geschichte

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt



Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

Anforderungen
sind der
Kern der Anwendung

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

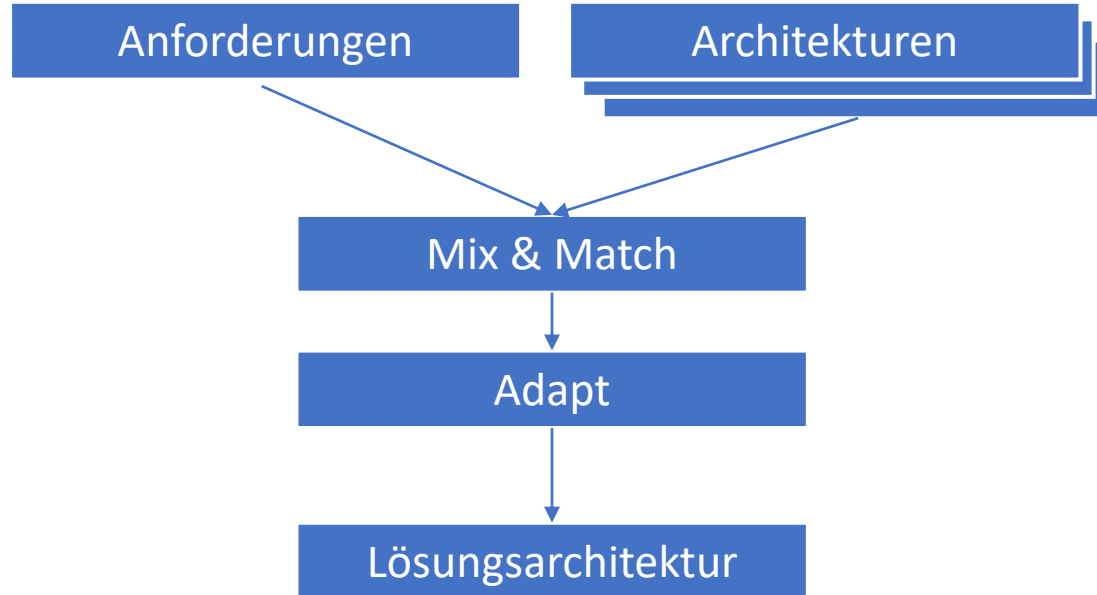
Architekturen
müssen
Anforderungen erfüllen

Agenda

13.11.2024 JUG Fra 2024

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt



Zwei Beispiele – Microservices?

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

Frachtbrief-Verwaltung

FraWa

- Erstellung von Frachtbriefen
- Weltweit in jedem Standort
 - See-/Flughafen
 - Spedition
 - Lager
- Druck von Frachtbriefen
(+ Durchschläge)
- Revisionssicheres Logging

Echtzeitverfolgung von Lieferungen

LiveLieferung

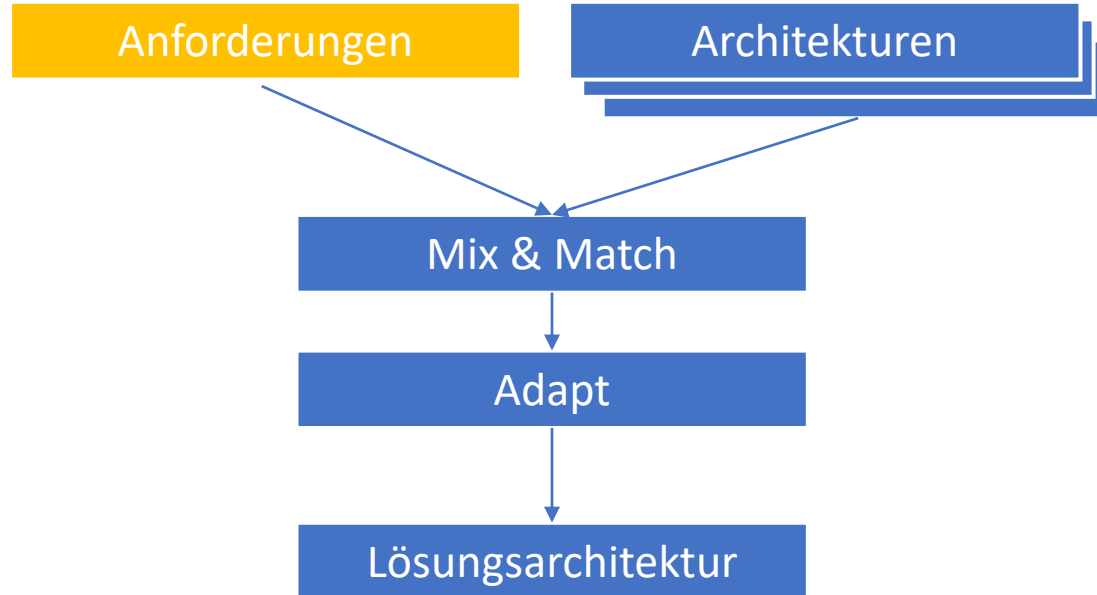
- Auslieferzeitpunkt für Endkunden
- DACH
- Nahezu Echtzeitdaten
aller Lieferfahrzeuge
- Mehrere 10T Fahrzeuge
- Webseite für Lieferstatus

Agenda

13.11.2024 JUG Fra 2024

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt



Anforderungen an das System (anhand von Arc42)

Top Quality Goals

External Systems
Organisation

1. Introduction & Goals

7. Deployment View

2. Constraints

8. Crosscutting Concepts

3. Context and Scope

9. Architectural Decisions

4. Solution Strategy

10. Quality Requirements

5. Building Block View

11. Risks & Technical Debt

6. Runtime View

12. Glossary

Qualitäts-
Anforderungen &
nichtfunktionale
Anforderungen

Anforderung am Beispielprojekt FraWa

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

Standorte mit schlechtem Internet

- Langsame Verbindung
- Regelmäßige lange Ausfälle

Anpassung an sich “regelmäßig”
ändernde nationale und internationale
Zollbestimmungen

Anforderung

Mitarbeiter müssen bei Netzausfall weiter
arbeitsfähig sein

Software muss zeitnah an geänderte
Regelungen angepasst werden können

Wegen der Umgebung muss die
Software auf geschützten (Umwelt und
Zugriff) Terminals laufen

Viele Installationen, daher preiswerter
Betrieb

Anforderung am Beispielprojekt LiveLieferung

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

Ständige Verfügbarkeit der Informationen Lastkurven

- Täglich (Arbeitszeiten der Fahrer, viele Informationen wenn morgen die Wagen beladen werden)
- Wöchentlich (viele Auslieferungen Montag, da sich die Bestellungen über die Woche sammeln)
- Jährlich (Weihnachten, Urlaubszeiten)

Information als Webseite, Handy-App

Anforderung

Sehr hohe Verfügbarkeit

Software muss schnell an geänderte
Regelungen angepasst werden können

Es muss kostensparend den Lastverlauf
eines Tages und einer Woche folgen
können

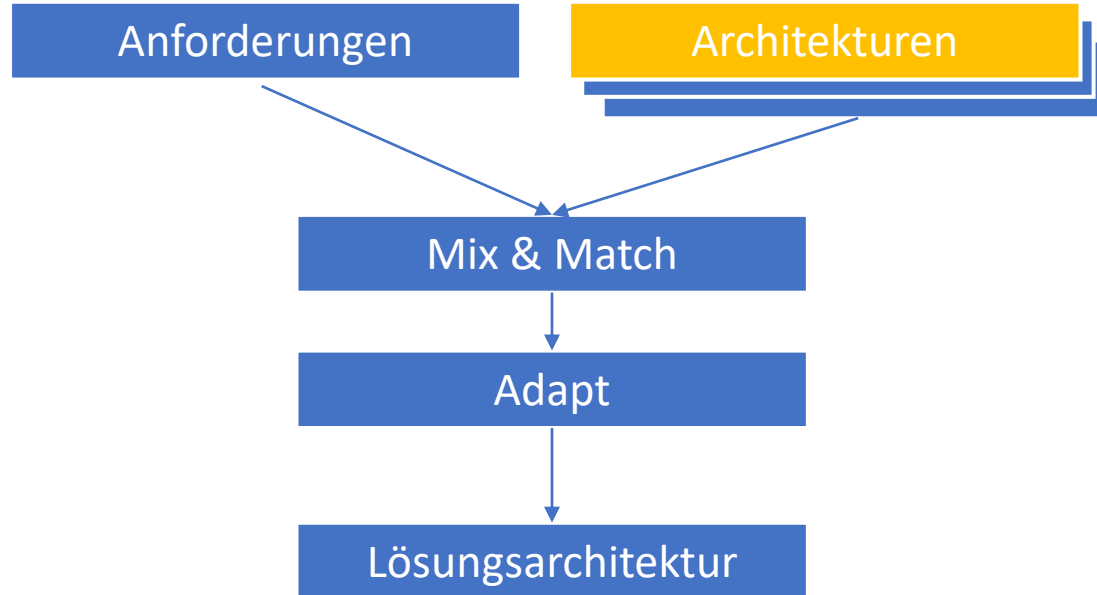
Zentraler Service mit vielen Connections
(in & outbound)

Agenda

13.11.2024 JUG Fra 2024

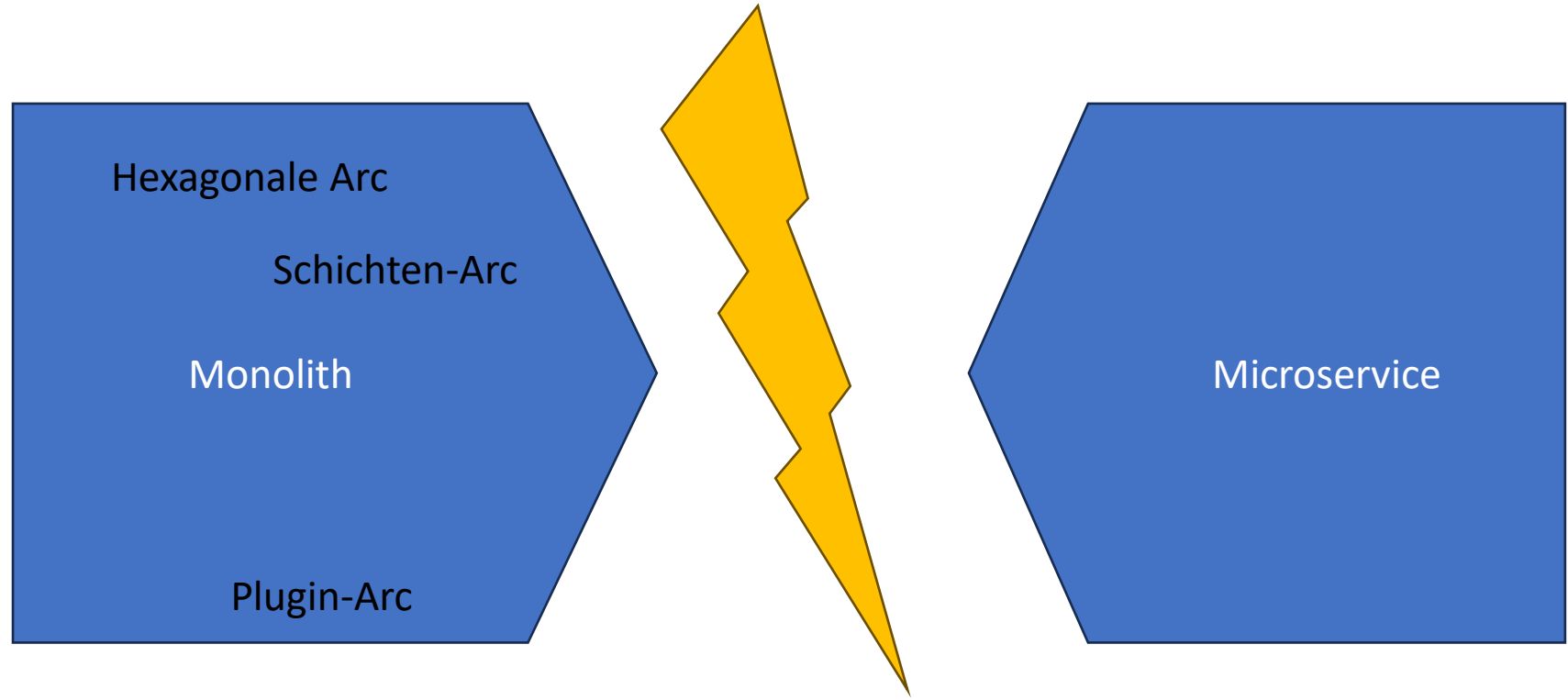
Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt



Falsches Bild

Es muss nicht immer
Microservice sein
Thomas Ruhroth und Kai Schmidt



Keine Alleinstellungsmerkmale von Microservices

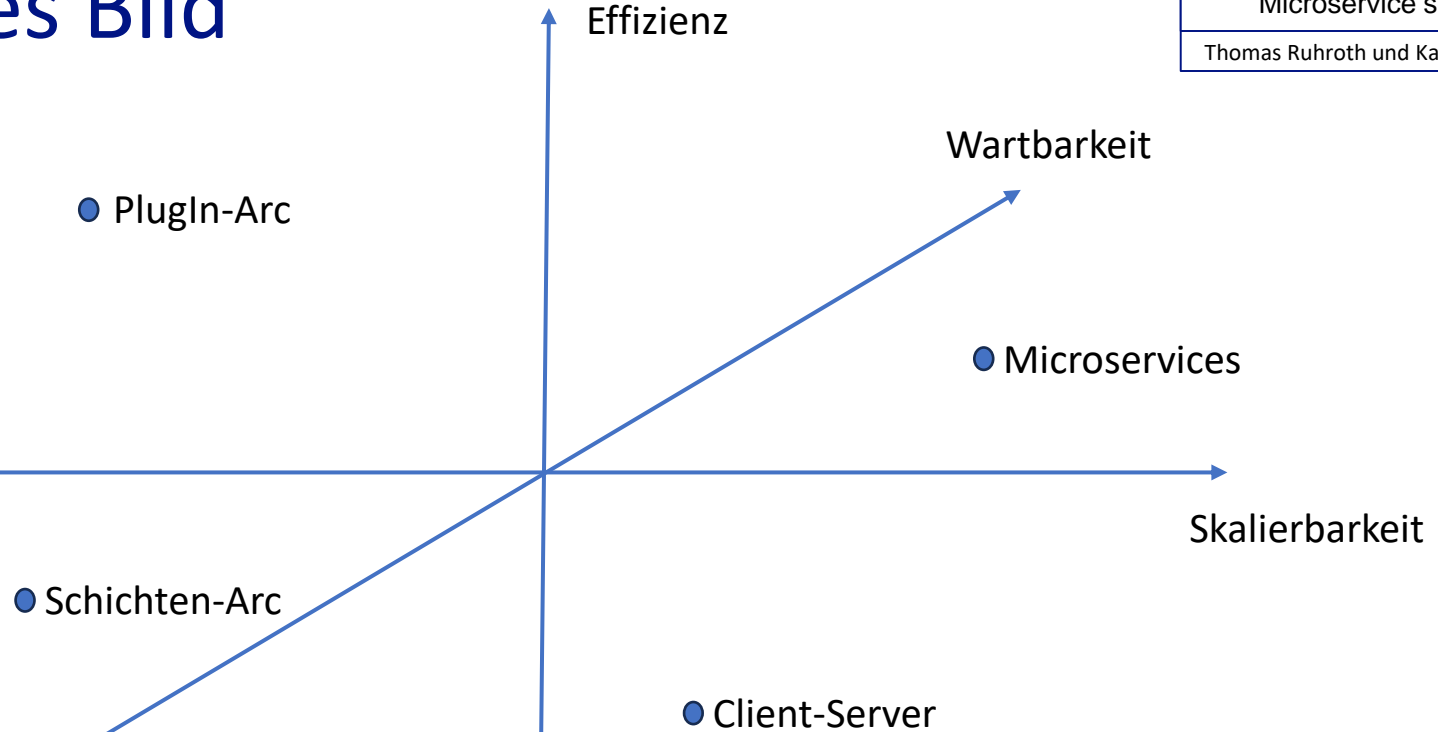
Information Hiding
Architektur Muster
Domain Driven Design
Feature Toggle
Continuous Delivery
Event Driven Design
Cloudfähigkeit

Modularisierung
CQRS
Adapters

Besseres Bild

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt



Besseres Bild

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

Effizienz

Unvollständig !

Die Einteilung in ein solches Bild ist nicht einfach

Es gibt mehr als drei Dimensionen

Die Dimensionen sind nicht nur rein technischer
Natur

Beispielsweise die Organisationsstruktur und deren
Kommunikationswege spielen eine wichtige Rolle

ices

Skalierbarkeit

Organisatorische, technische und rechtliche Rahmenbedingungen

Beispiele

Laufzeitumgebung

Cloud
klassische Server
lokale Programme
IoT-Systeme
Mobile-Devices
Kleinstrechner (Raspberry PI)

Datenschutz

Recht auf Auskunft
Recht auf Löschen
Daten in der EU

Freigabe/Zertifizierung

Medizinische Produkte
Luftfahrt

ISO 25010 Qualitätsanforderungen

Software Product Quality

13.11.2024 JUG Fra 2024

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

Functional Suitability	Performance Efficiency	Compatibility	Usability	Reliability	Security	Maintainability	Portability
Functional Completeness Functional Correctness Functional Appropriateness	Time Behaviour Resource Utilization Capacity	Co-Existence Interoperability	Appropriateness Recognizability Operability User Error Protection User Interface Aesthetics Accessibility	Maturity Availability Fault Tolerance Recoverability	Confidentiality Integrity Non-Reputation Authenticity Accountability	Modularity Reusability Analysability Modifiability Testability	Adaptability Installability Replaceability

ISO 25010 Qualitätsanforderungen

Software Product Quality

Functional Suitability	Performance Efficiency	Compatibility	Usability	Reliability	Security	Maintainability	Portability
Functional Completeness Functional Correctness Functional Appropriateness	Time Behaviour Resource Utilization Capacity	Co-Existence Interoperability	Appropriateness Recognizability Operability User Error Protection User Interface Aesthetics Accessibility	Maturity Availability Fault Tolerance Recoverability	Confidentiality Integrity Non-Reputation Authenticity Accountability	Modularity Reusability Analysability Modifiability Testability	Adaptability Installability Replaceability

Beispiele negativer Wechselwirkung

Eigenschaften von Architekturen

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

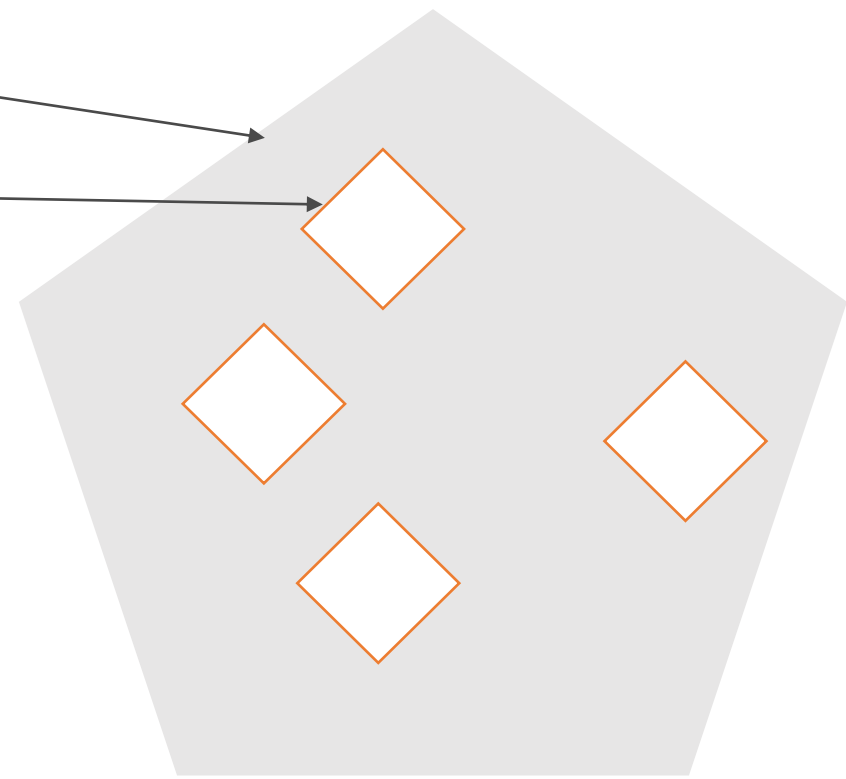
Anforderungen	Plugin-Architektur	Microservice	Schichten-Architektur	Hexagonale Architektur
Skalierbarkeit auf kleiner Ebene	Schlecht	Sehr gut	Schlecht	Schlecht
Modulare Deploybarkeit	Sehr gut	Sehr gut	Schlecht	Schlecht
Geringe Laufzeitkosten	Sehr gut	Mittel	Sehr gut	Sehr gut
Innere Komplexität	Mittel	N/A	Sehr gut	Gut
Eigene Analysen				

Mikro/Makro

Es muss nicht immer
Microservice sein
Thomas Ruhroth und Kai Schmidt

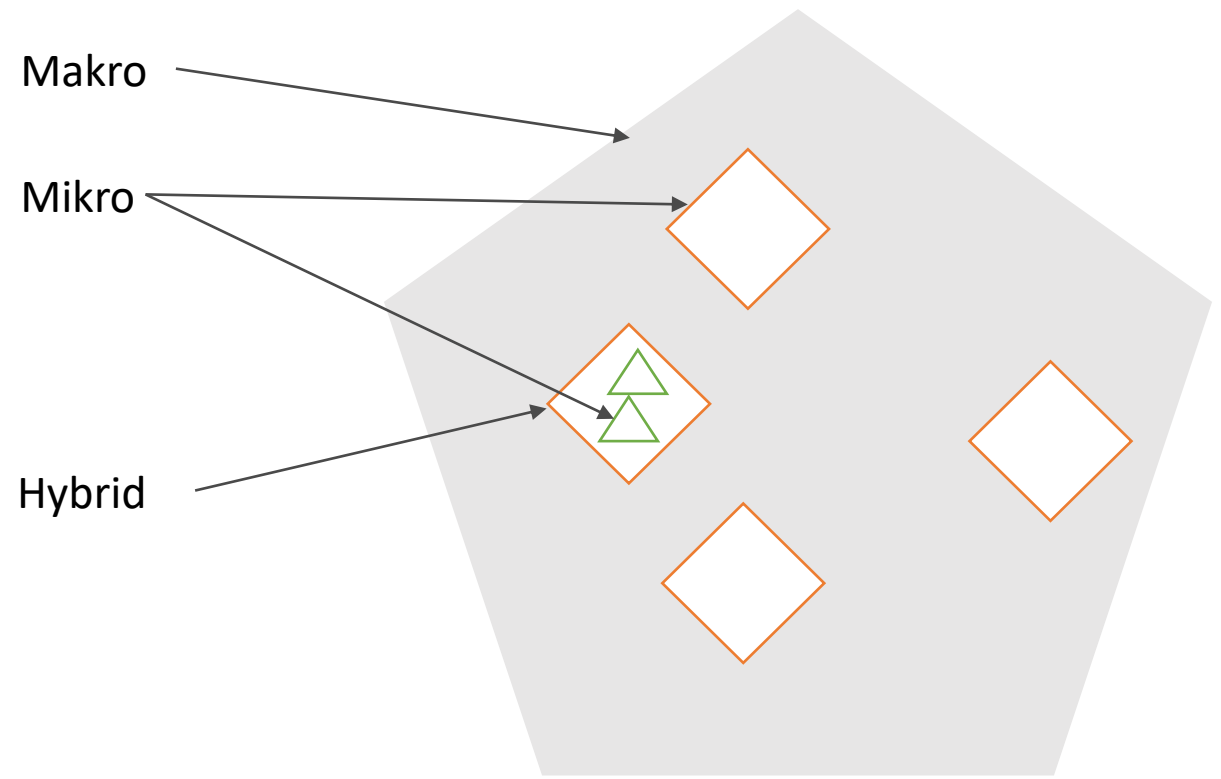
Makro

Mikro



Mikro/Makro/Hybrid

Es muss nicht immer
Microservice sein
Thomas Ruhroth und Kai Schmidt



Eigenschaften von Architekturen

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

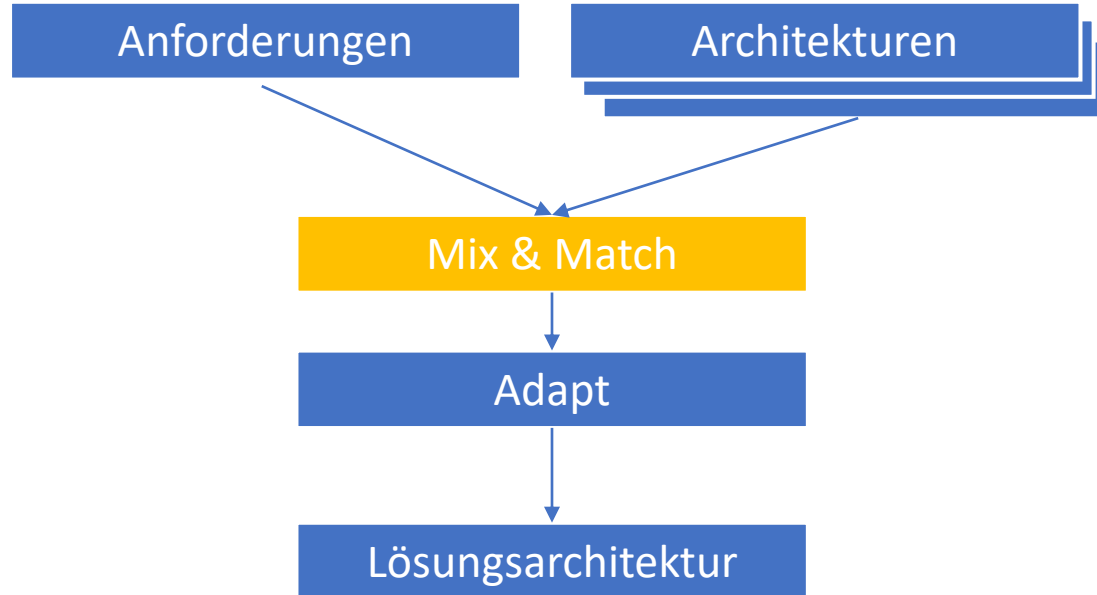
Anforderungen	Plugin-Architektur	Microservice	Schichten-Architektur	Hexagonale Architektur
Scope	Hybride	Makro	Hybride	Mikro
Skalierbarkeit auf kleiner Ebene	Schlecht	Sehr gut	Schlecht	Schlecht
Modulare Deploybarkeit	Sehr gut	Sehr gut	Schlecht	Schlecht
Geringe Laufzeitkosten	Sehr gut	Mittel	Sehr gut	Sehr gut
Innere Komplexität	Mittel	N/A	Sehr gut	Gut
Eigene Analysen				

Agenda

13.11.2024 JUG Fra 2024

Es muss nicht immer
Microservice sein

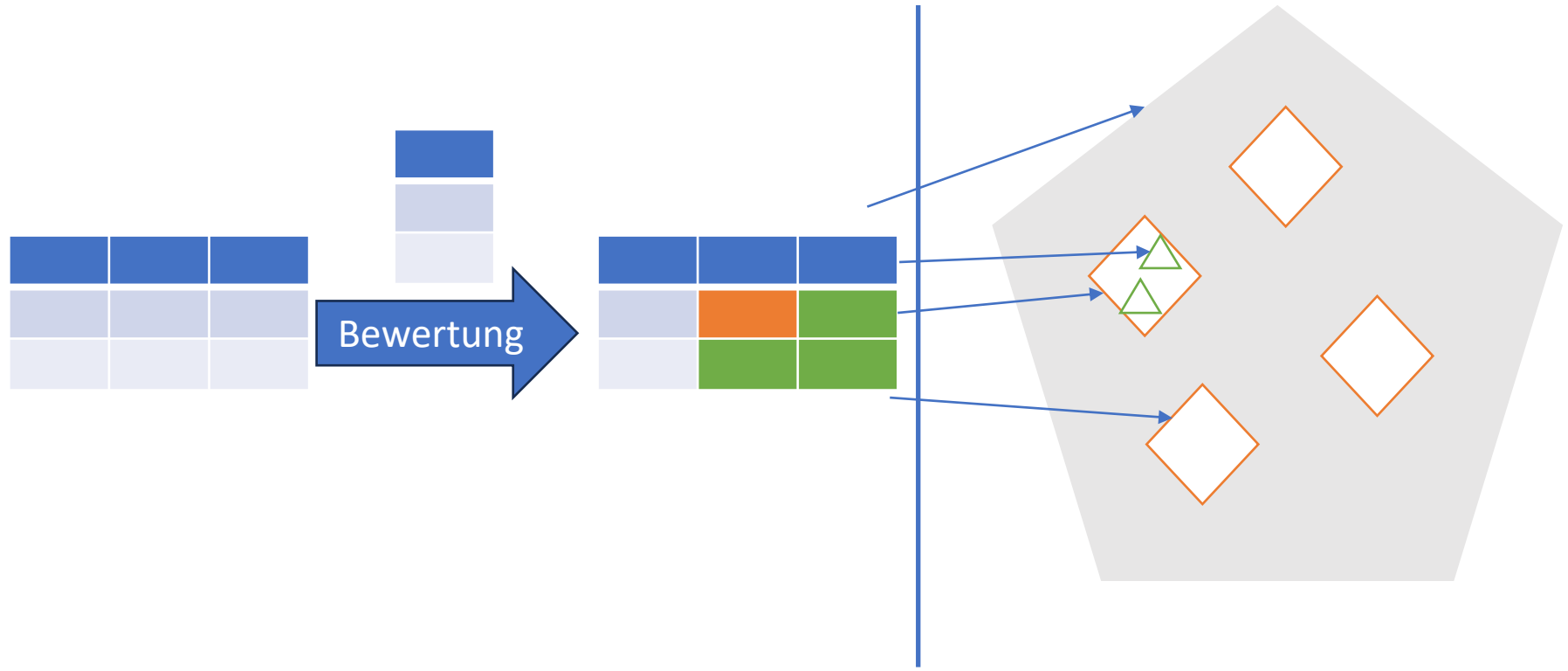
Thomas Ruhroth und Kai Schmidt



Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

Mix and Match



FraWa Makro

13.11.2024 JUG Fra 2024

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

Anforderung	Anforderungen	Plugin-Architektur	Microservice	Schichten-Architektur	Hexagonale Architektur
Mitarbeiter müssen bei Netzausfall weiter arbeitsfähig sein	Scope	Hybride	Makro	Hybride	Mikro
Software muss schnell an geänderte Regelungen angepasst werden können	Skalierbarkeit auf kleiner Ebene	Schlecht	Sehr gut	Schlecht	Schlecht
Wegen der Umgebung muss die Software auf geschützten (Umwelt und Zugriff) Terminals laufen	Modulare Deploybarkeit	Sehr gut	Sehr gut	Schlecht	Schlecht
	Geringe Laufzeitkosten	Sehr gut	Mittel	Sehr gut	Sehr gut
Viele Installationen, daher preiswerter Betrieb	Innere Komplexität	Mittel	N/A	Sehr gut	Gut

Makro: PlugIn-Architektur

PlugIn-Framework

Plugin

Plugin

Plugin

LiveLieferung (Makro)

13.11.2024 JUG Fra 2024

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

Anforderung	Anforderungen	Plugin-Architektur	Microservice	Schichten-Architektur	Hexagonale Architektur
Sehr hohe Verfügbarkeit	Scope	Hybride	Makro	Hybride	Mikro
Software muss schnell an geänderte Regelungen angepasst werden können	Skalierbarkeit auf kleiner Ebene	Schlecht	Sehr gut	Schlecht	Schlecht
Es muss kostensparend den Lastverlauf eines Tages und einer Woche folgen können	Modulare Deploybarkeit	Sehr gut	Sehr gut	Schlecht	Schlecht
Zentraler Service mit vielen Connections (in & outbound)	Geringe Laufzeitkosten	Sehr gut	Mittel	Sehr gut	Sehr gut
	Innere Komplexität	Mittel	N/A	Sehr gut	Gut

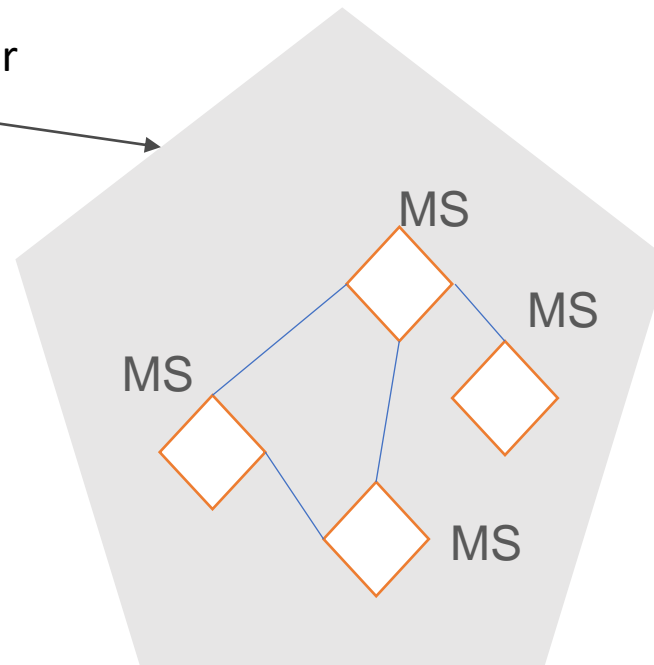
LiveLieferung

13.11.2024 JUG Fra 2024

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

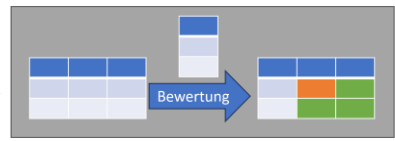
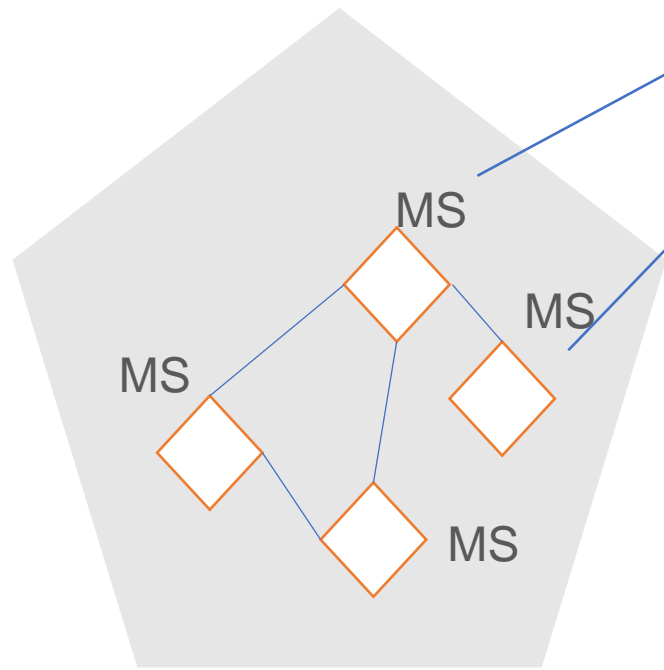
Makro: Microservice-Architektur



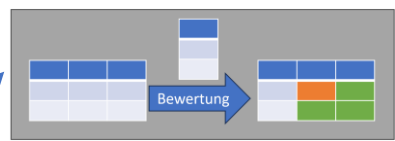
LiveLieferung

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt



Hexagonale Arc



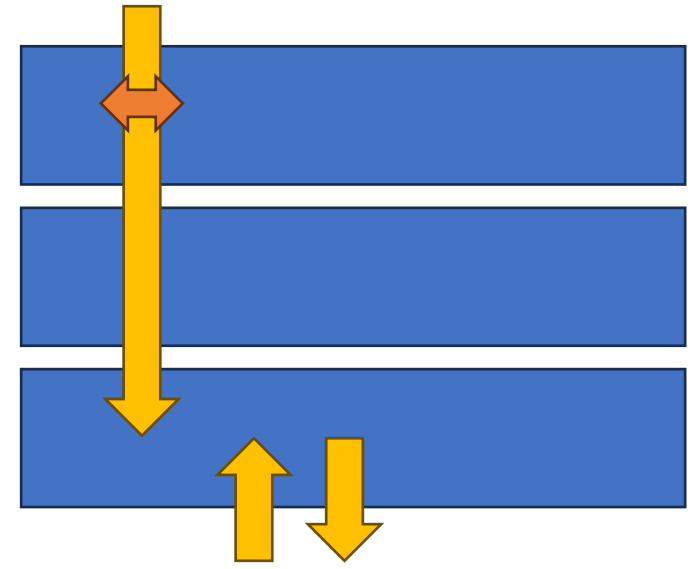
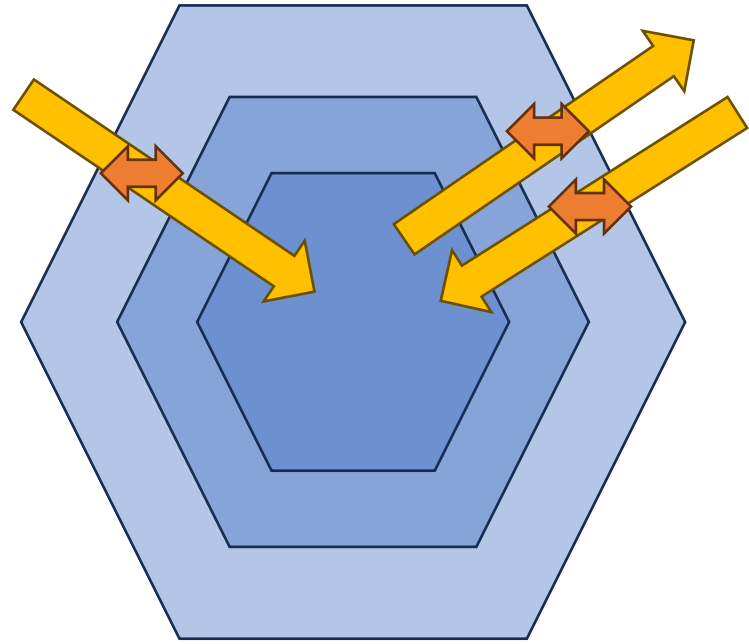
Schichten-Arc:
Revisionssicheres Log

Anforderungen	PlugIn-Architektur	Microservice	Schichten-Architektur	Hexagonale Architektur
Scope	Hybride	Makro	Hybride	Mikro
Skalierbarkeit auf kleiner Ebene	Schlecht	Sehr gut	Schlecht	Schlecht
Modulare Deploybarkeit	Sehr gut	Sehr gut	Schlecht	Schlecht
Geringe Laufzeitkosten	Sehr gut	Mittel	Sehr gut	Sehr gut
Innere Komplexität	Mittel	N/A	Sehr gut	Gut
Performance Data to DB	N/A	N/A	Gut	Mittel

Hexagonal vs. Schichten

Es muss nicht immer
Microservice sein

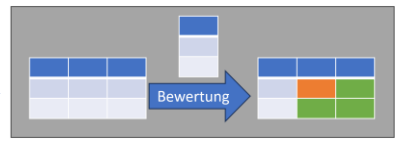
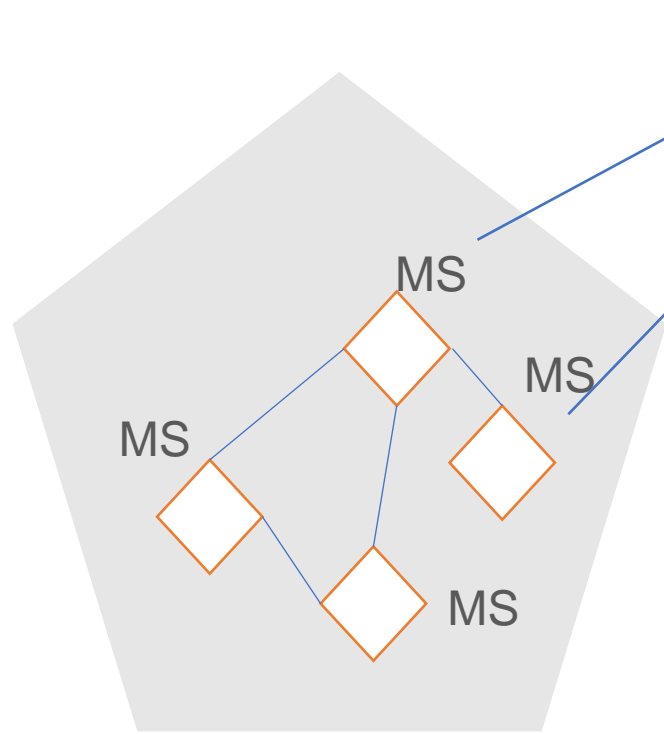
Thomas Ruhroth und Kai Schmidt



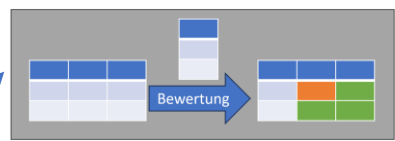
LiveLieferung

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt



Hexagonale Arc



Schichten-Arc:
Revisionssicheres Log

Anforderungen	Plugin-Architektur	Microservice	Schichten-Architektur	Hexagonale Architektur
Scope	Hybride	Makro	Hybride	Mikro
Skalierbarkeit auf kleiner Ebene	Schlecht	Sehr gut	Schlecht	Schlecht
Modulare Deploybarkeit	Sehr gut	Sehr gut	Schlecht	Schlecht
Geringe Laufzeitkosten	Sehr gut	Mittel	Sehr gut	Sehr gut
Innere Komplexität	Mittel	N/A	Sehr gut	Gut
Performance Data to DB	N/A	N/A	Gut	Mittel

Wechselwirkungen zwischen Architekturen - Beispiele

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

Microservices und Plugin-Arc
Überschneidende Lösungen

Hexagonale Arc und
Functional Core/Imperative Shell
Testbarkeit der Services und Domains

Negative Wechselwirkungen

Positive Wechselwirkungen

Schichten-Arc und Plugin-Arc
Plugins über Schichten

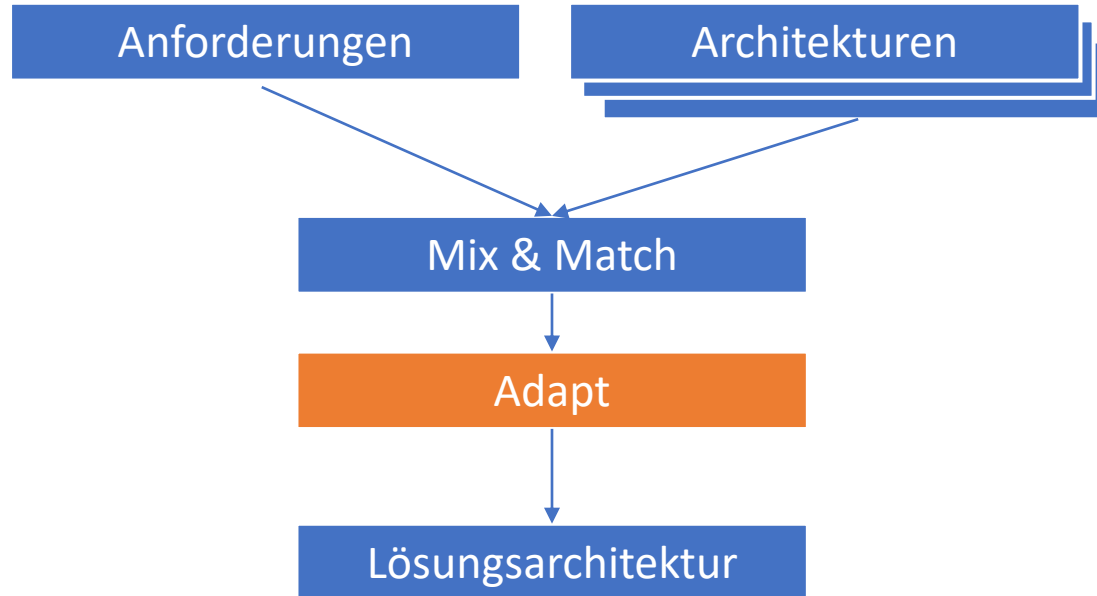
Schichten-Arc und Plugin-Arc
Plugins jeweils als Schichtenarchitektur

Agenda

13.11.2024 JUG Fra 2024

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt



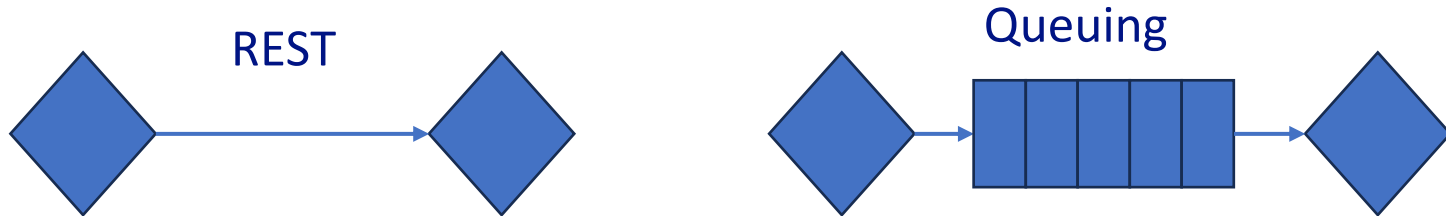
Adapt - LiveLieferung

13.11.2024 JUG Fra 2024

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

Revisionssichere Logs



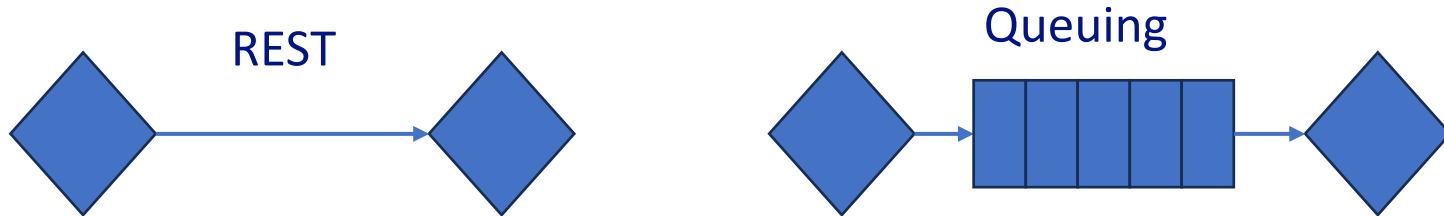
Adapt - LiveLieferung

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt

Was für den einen gut ist, muss für den anderen nicht gut sein.

Don't bend it till it breaks.

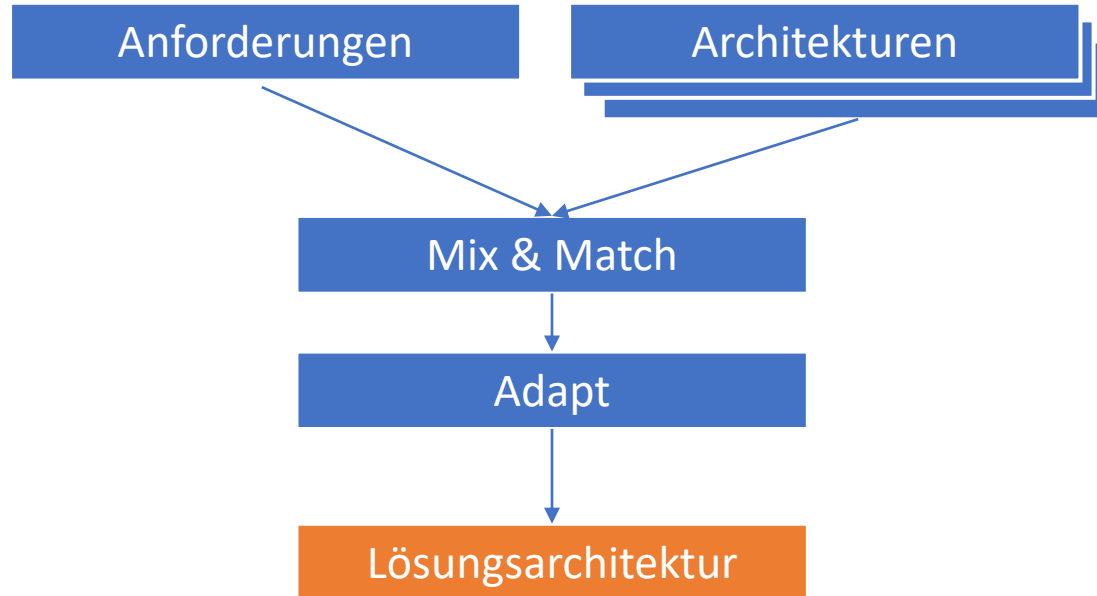


Agenda

13.11.2024 JUG Fra 2024

Es muss nicht immer
Microservice sein

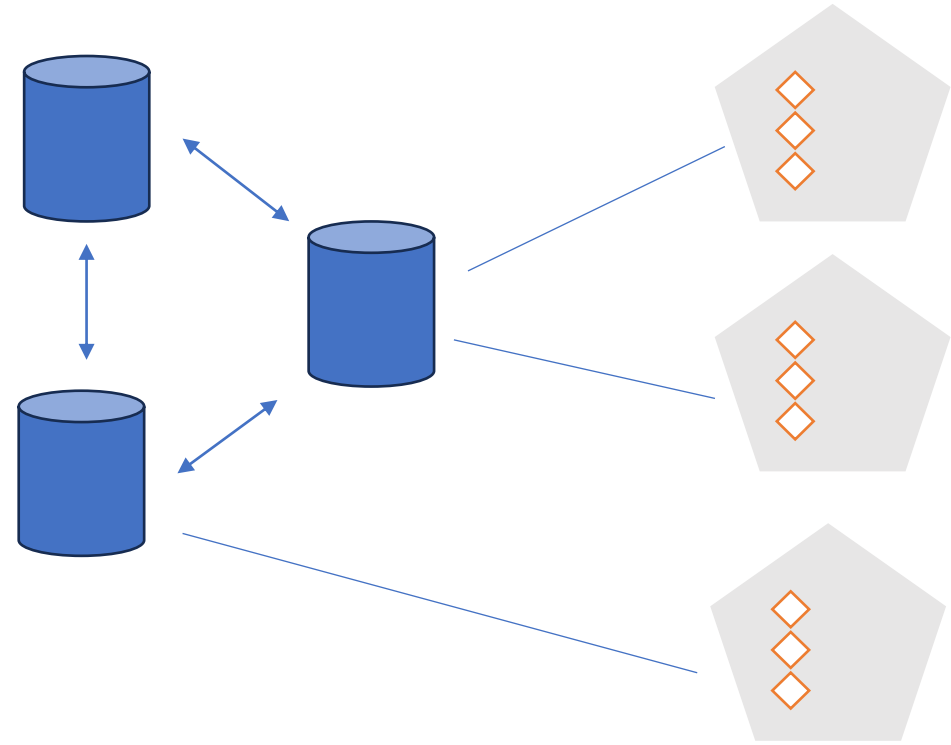
Thomas Ruhroth und Kai Schmidt



Lösungsarchitektur - FraWa

Es muss nicht immer
Microservice sein

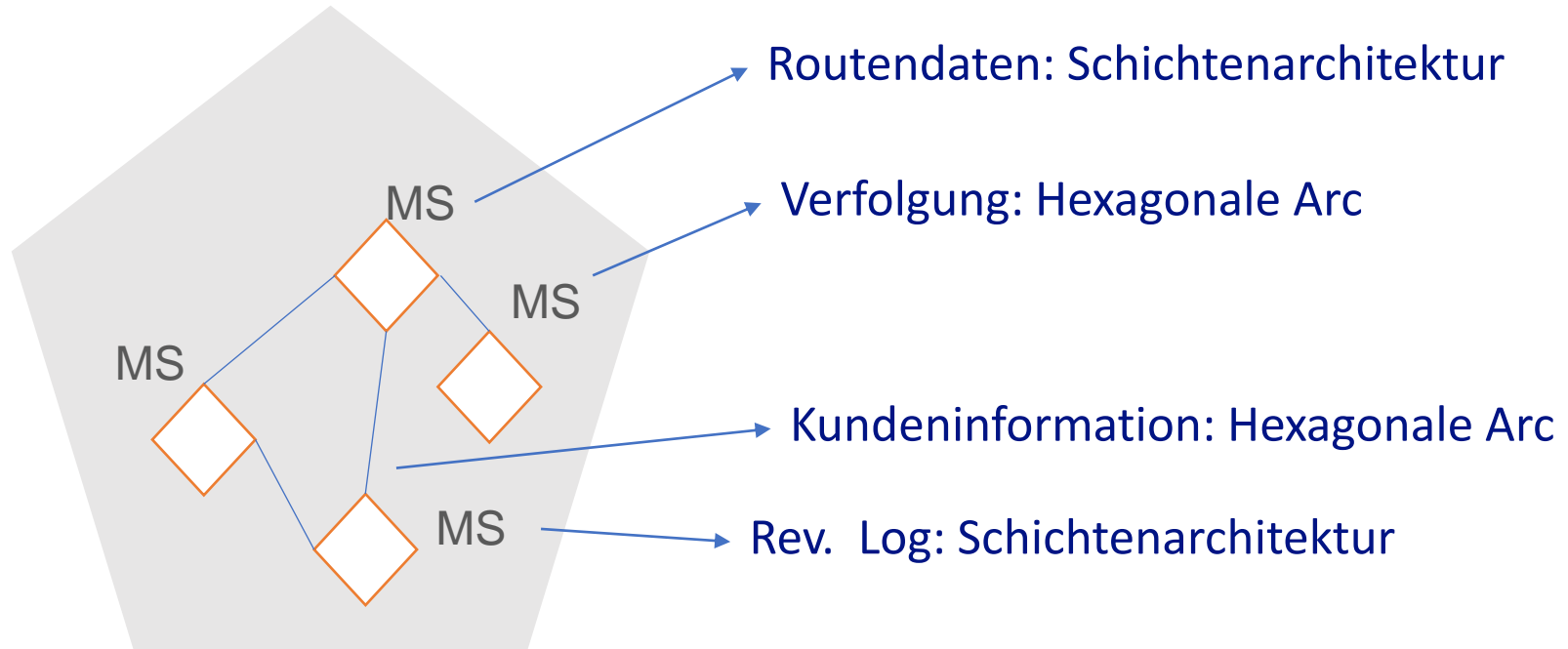
Thomas Ruhroth und Kai Schmidt



Lösungsarchitektur - LiveLieferung

Es muss nicht immer
Microservice sein

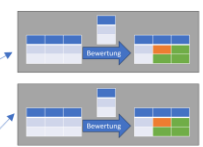
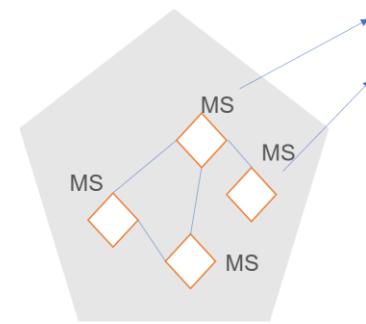
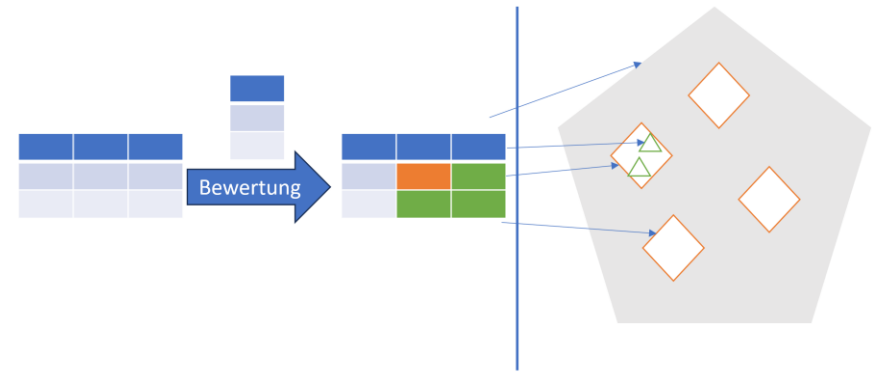
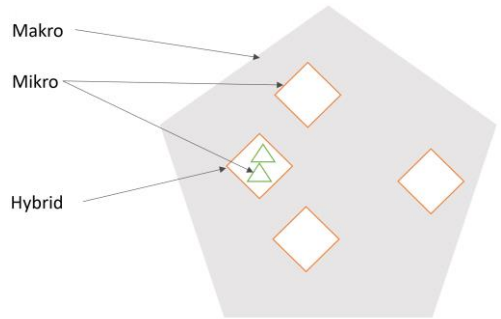
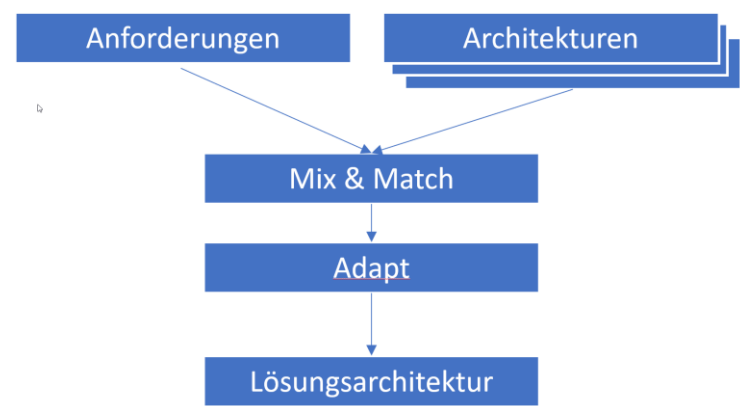
Thomas Ruhroth und Kai Schmidt



Zusammenfassung

Es muss nicht immer
Microservice sein

Thomas Ruhroth und Kai Schmidt



Hexagonale Arc

Schichten-Arc:
Revisionsssicheres Log

Anforderungen	Plugin-Architektur	Microservice	Schichten-Architektur	Hexagonale Architektur
Scope	Hybrid	Makro	Hybrid	Mikro
Skalierbarkeit auf kleiner Ebene	Schlecht	Sehr gut	Schlecht	Schlecht
Modulare Deploybarkeit	Sehr gut	Sehr gut	Schlecht	Schlecht
Geringe Laufzeitkosten	Sehr gut	Mittel	Sehr gut	Sehr gut
Innere Komplexität	Mittel	N/A	Sehr gut	Gut
Performance Data to DB	N/A	N/A	Gut	Mittel