



# Digitalisierung mit APIs

Möglichkeiten zur massenhaften API-fizierung von Altanwendungen

**Doktorandentag 10. Juli 2023**  
**Konrad Nadobny**



## Historisch gewachsene Systemlandschaften

zahlreiche Direktverbindungen - hohe Komplexität  
direkte Integration auf Datenbank-Level - keine Abstraktion  
technische Heterogenität - problematischer Betrieb  
Versteckte Abhängigkeiten - Anpassungen schwierig



Schlechte Architektur bremst Innovation & Agilität



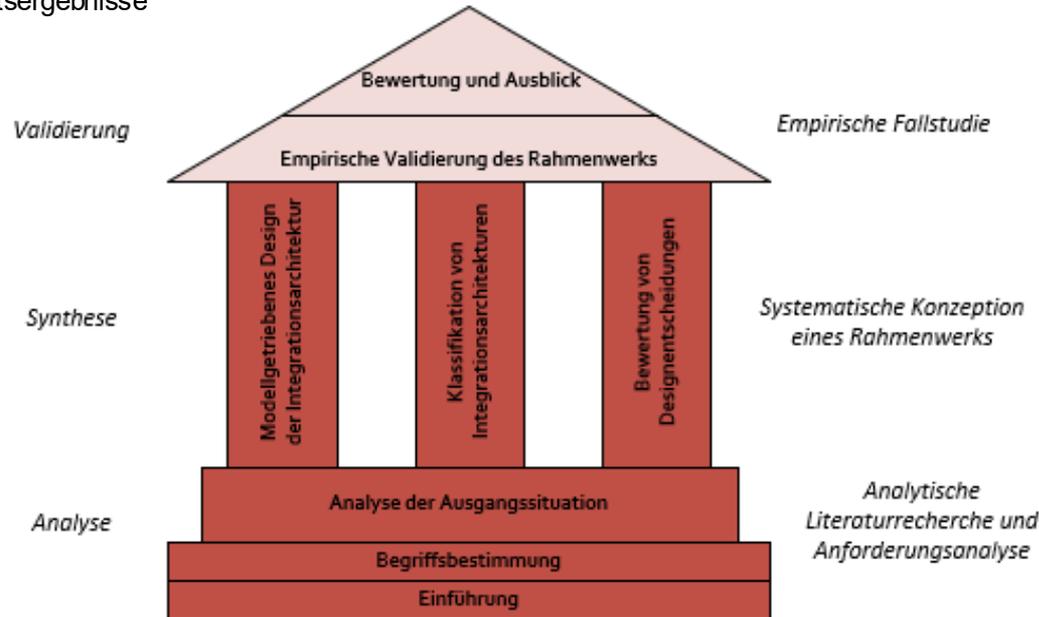
*API-fizierung als Werkzeug zur Transformation  
hin zum modernen Konnektivitäts-Ökosystem*



# Arbeitsstand der Dissertation



- Konsolidierung der Arbeitsergebnisse



# These und Forschungsfragen



**These: Integrationsarchitekturen lassen sich modellgetrieben optimieren.**

## **Modell zur Optimierung der Systemintegration:**

- 1)Wie kann Systemintegration modellgetrieben verbessert werden?
- 2)Wie lassen sich Integrationsarchitekturen modellieren?
- 3)Wie lassen sich Integrationsarchitekturen bewerten?
- 4)Wie lässt sich der Transformationsprozess operationalisieren?

## **Klassifizierung von Integrationsarchitekturen**

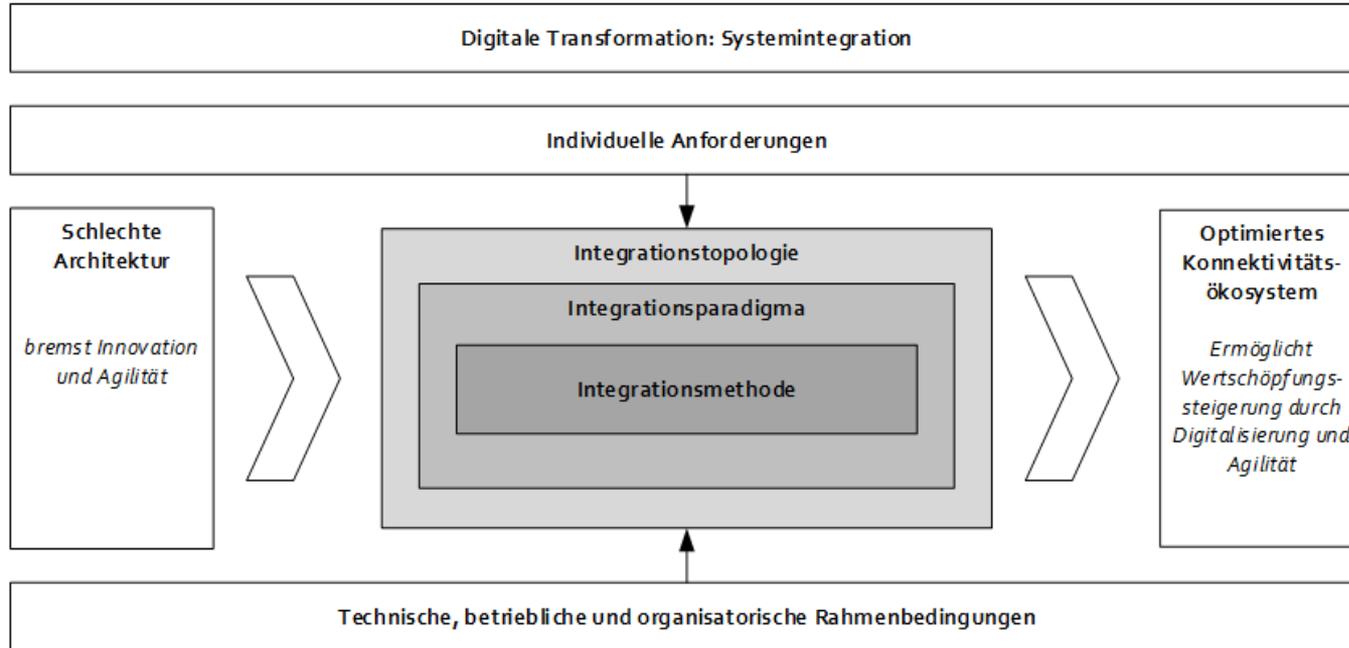
- 1)Nach welcher Systematik können Integrationsarchitekturen unterschieden werden und welche Klassifizierung ergibt sich daraus?

## **Bewertung von Integrationsarchitekturen**

- 1)Wie ist der Einfluss verschiedener Integrationsarchitekturen auf die Qualität der Systemintegration zu bewerten?



# Forschungsergebnis: GAP<sub>1</sub> Framework





# Forschungsergebnis: GAP<sub>1</sub> Framework

## Schritt 1: Bewertungskriterien gewichten

Schritt 2: Architektur Klassifizieren

Schritt 2: Archetypen Bewerten

	Robustheit	Integrität	Komplexität	Durchsatz	Latenz	Gewichtung	
Robustheit	-	2	3	3	3	11	28%
Integrität	2	-	1	2	3	8	20%
Komplexität	1	3	-	3	3	10	25%
Durchsatz	1	2	1	-	3	7	18%
Latenz	1	1	1	1	-	4	10%
<i>Zeile aufsummiert</i>						40	100%

1 = unwichtiger    2 = gleichwertig    3 = wichtiger

### Robustheit

Robustheit beschreibt, wie gut das Konnektivitätsökosystem auf den Ausfall einzelner Komponenten reagiert.

### Integrität

Informationen im Konnektivitätsökosystem werden zuverlässig synchronisiert, ohne dass es zu Konflikten und Uneinigkeiten kommt.

### Skalierbarkeit

Systeme lassen sich massenhaft zum Konnektivitätsökosystem hinzufügen und entfernen.

### Kontrolle

Die Komplexität der Systemlandschaft und der einzelnen Integrationen ist gut beherrschbar.

### Flexibilität

Bestehende Strukturen lassen sich einfach anpassen.

### Durchsatz

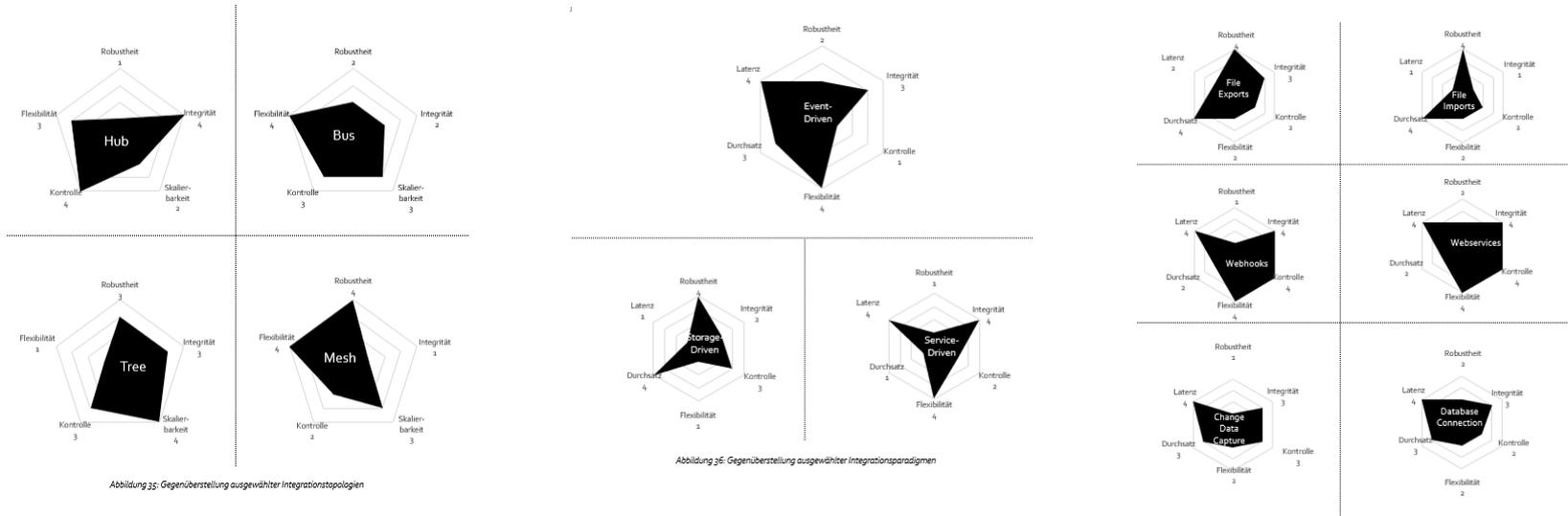
In einem begrenzten Zeitabschnitt können große Datenmengen übertragen werden.

### Latenz

Die verarbeiteten Informationen sind aktuell und die zeitliche Lücke zwischen Informationen in Quell- und Zielsystem ist sehr klein.

# Forschungsergebnis: GAP<sub>1</sub> Framework

Schritt 1: Bewertungskriterien gewichten  
**Schritt 2: Architektur Klassifizieren**  
 Schritt 2: Archetypen Bewerten





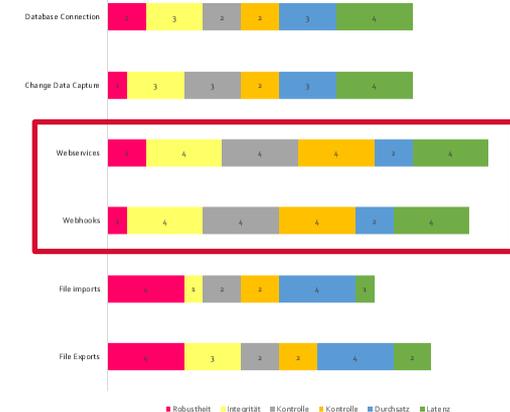
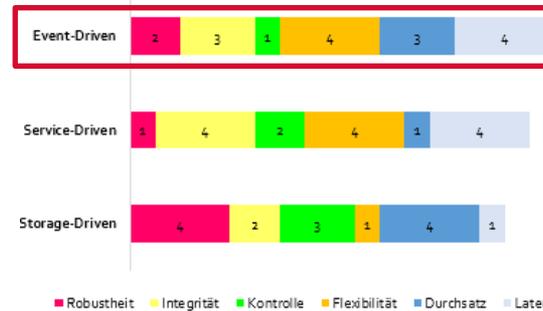
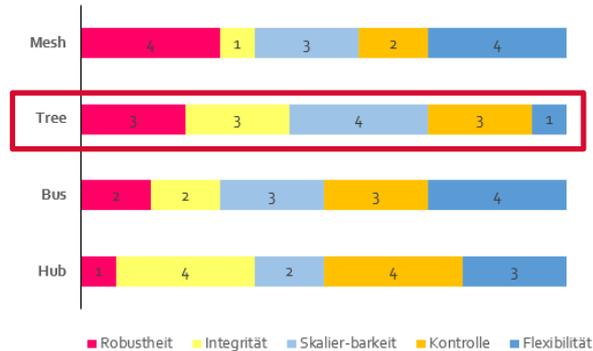
# Forschungsergebnis: GAP<sub>1</sub> Framework

Schritt 1: Bewertungskriterien gewichten

Schritt 2: Architektur Klassifizieren

Schritt 2: Archetypen Bewerten

	Robustheit	Integrität	Komplexität	Durchsatz	Latenz	Bewertung
<b>Gewichtung</b>	28%	20%	25%	18%	10%	
Storage-Driven	3	1	3	3	1	2,4
Service-Driven	1	3	2	1	3	1,85
Event-Driven	2	3	2	3	3	2,475



Skalierbare Lösung mit hohem Durchsatz?

→ Event-Driven Integration in einer Tree-Topologie mit Webservices, oder Webhooks



- 1. Beschreibung des Anwendungsfalls**
- 2. Anwendung des Rahmenwerks**
- 3. Beschreibung der tatsächlichen Implementierung**
- 4. Gegenüberstellung**

- *Anbindung verschiedener Systeme an ein Data Warehouse*
- *Anbindung von Satellitensystem an ein operatives Kernsystems*
- *Bereitstellung von Daten eines operativen Kernsystems*
- *Kontinuierliche Synchronisierung von zweier operativer Kernsysteme*