SEQIS

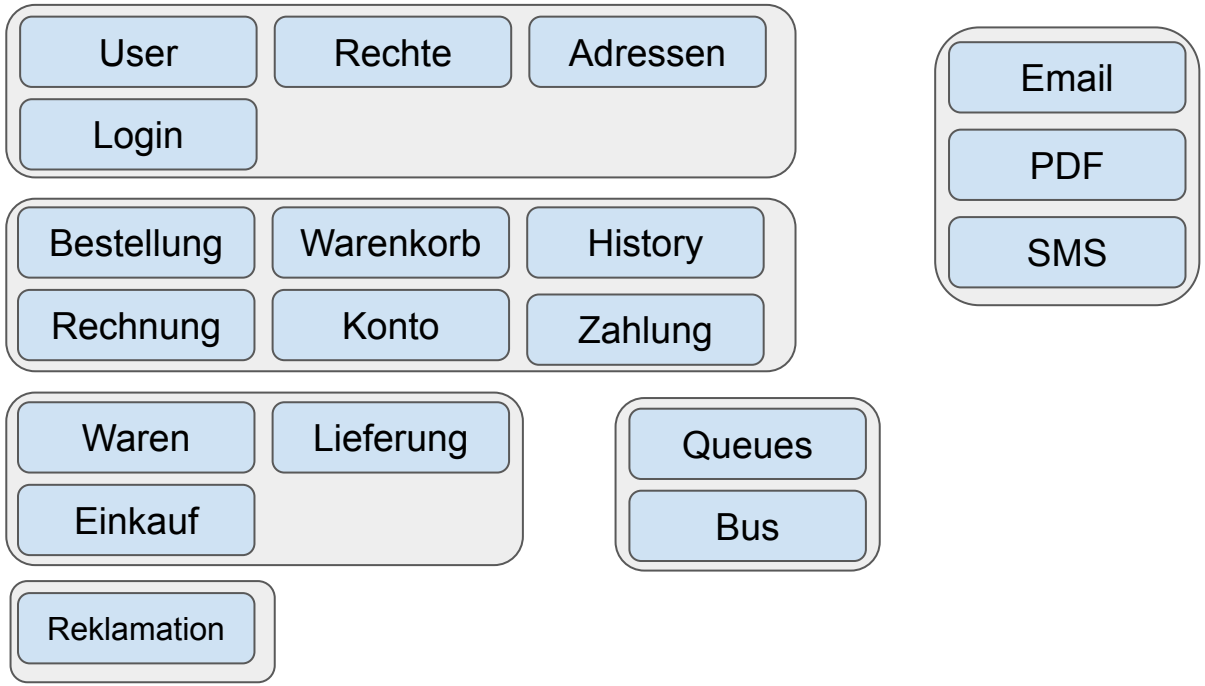
Ten more things

Services - Warum, wann und wie Micro?

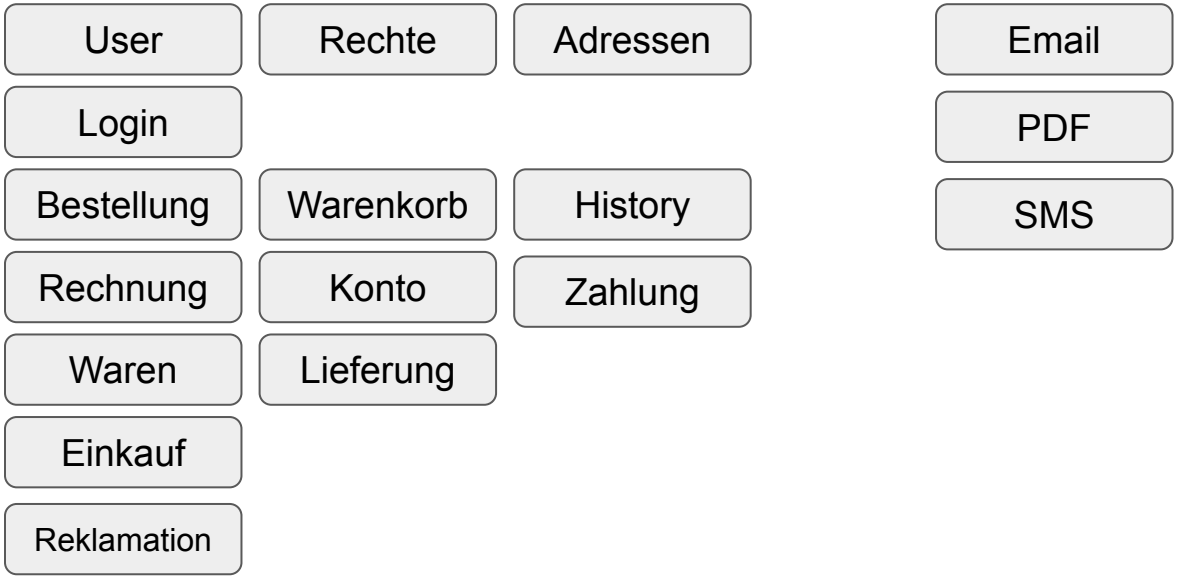
Schwabeneder Markus

(Modularer) Monolith

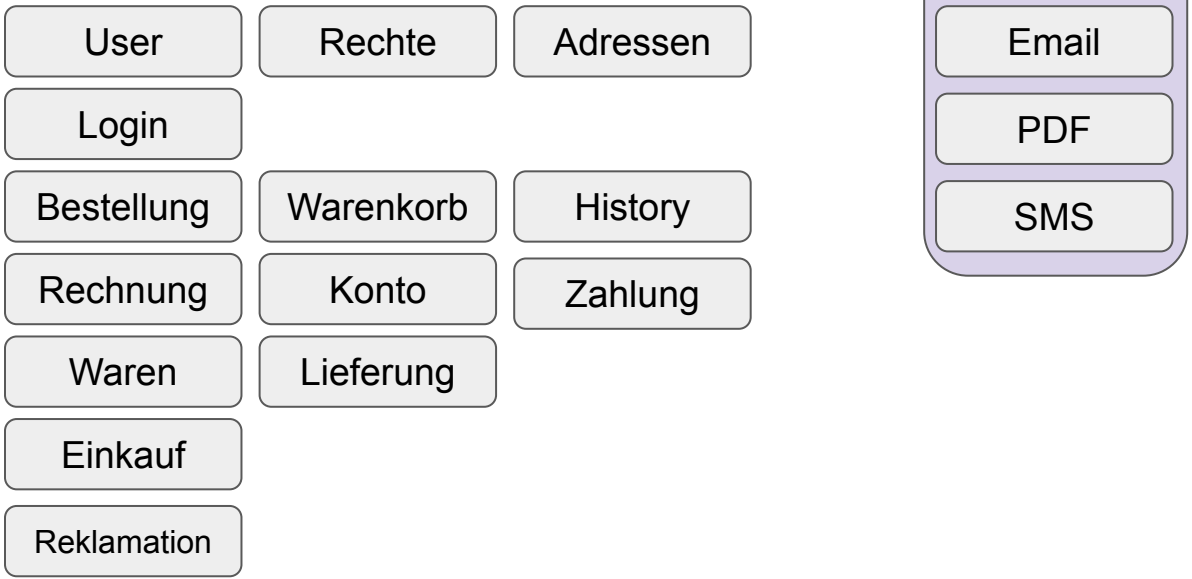
Service



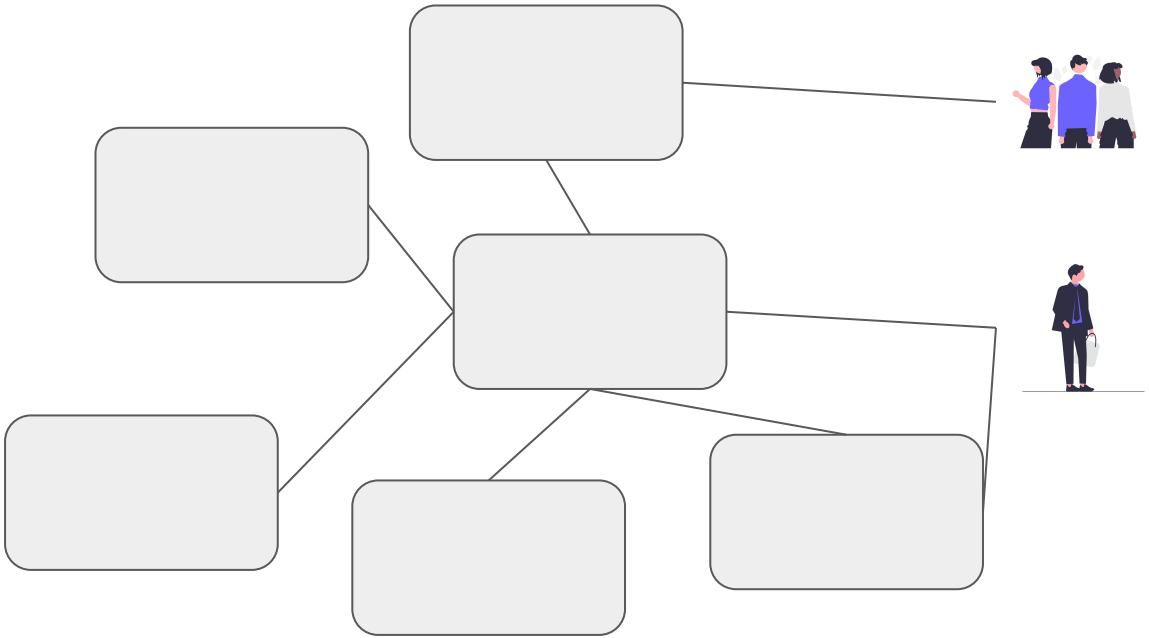
Microservice



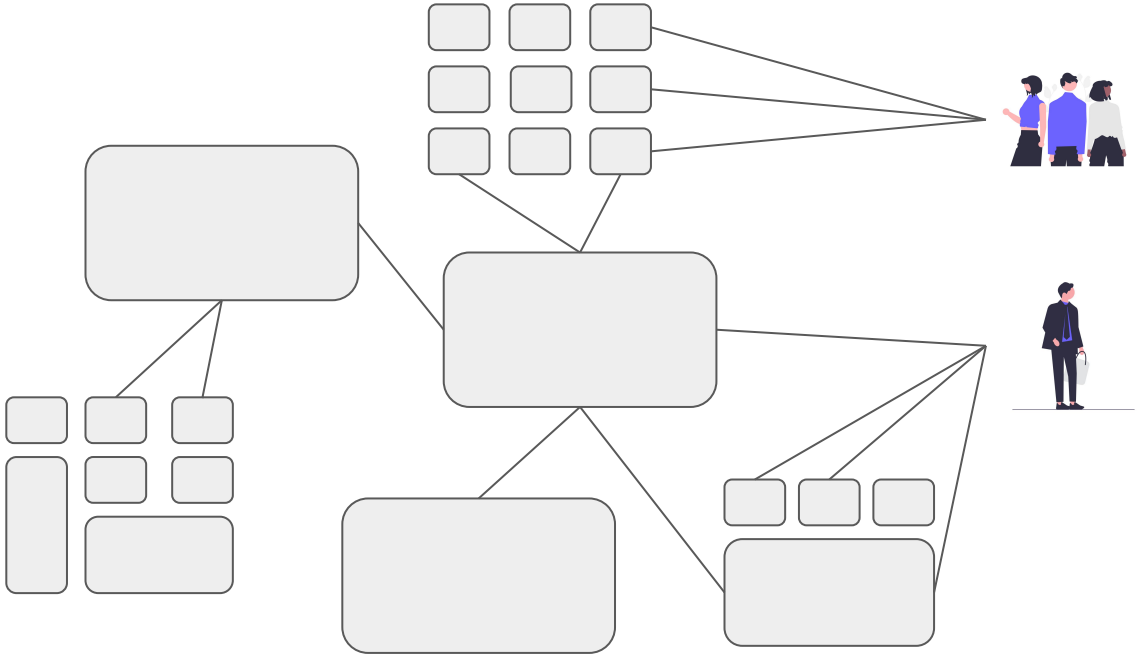
Microservice



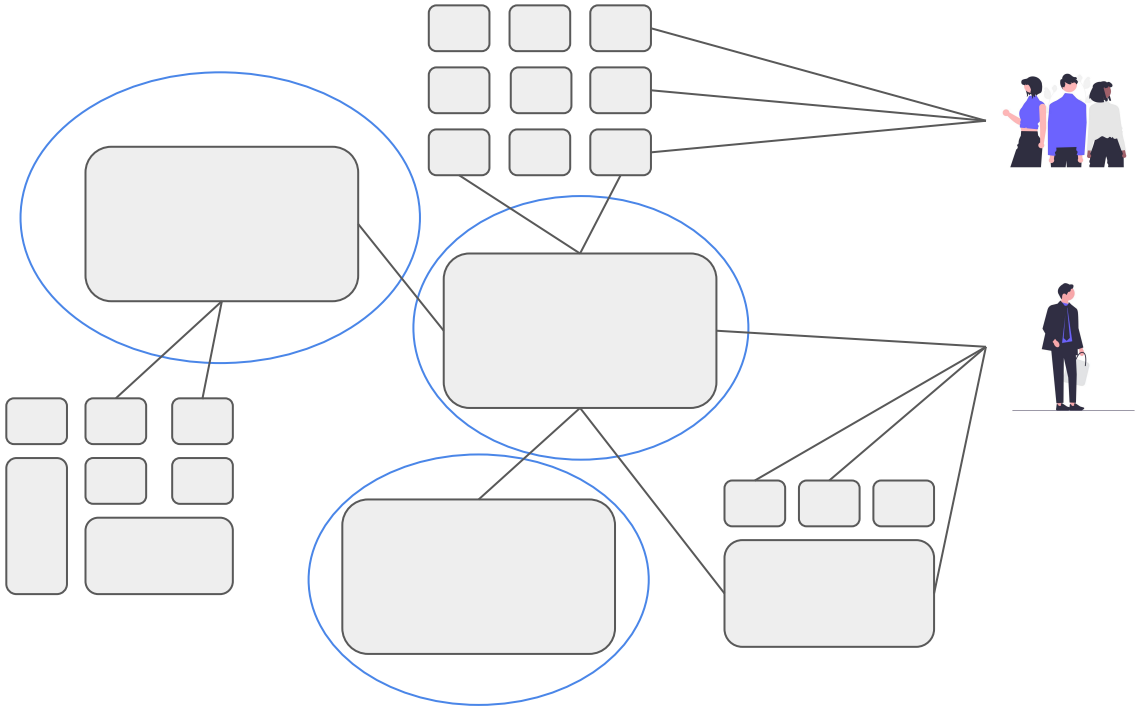
Architekturen und Scope



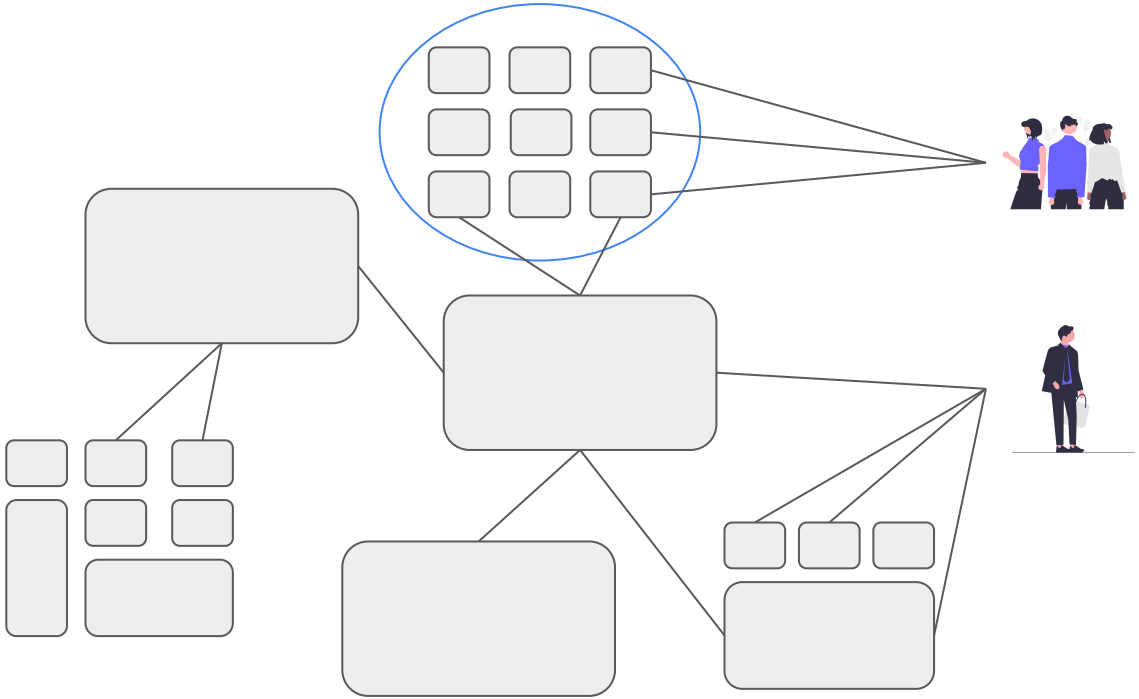
Architekturen und Scope



Monolith



Microservice Architecture



Vorteile des Monolithen

- Kein Kommunikations-Overhead

SEQIS

Vorteile des Monolithen

- Kein Kommunikations-Overhead
- Einfaches Deployment

SEQIS

Vorteile des Monolithen

- Kein Kommunikations-Overhead
- Einfaches Deployment
- Als Gesamtsystem weniger kompliziert

Vorteile von service-basierten Systemen

- Viel besser skalierbar

Vorteile von service-basierten Systemen

- Einzelne Services sind einfach

Vorteile von service-basierten Systemen

- Verschiedene Teams arbeiten unabhängig

Vorteile von service-basierten Systemen

- Verschiedene Technologien

Vorteile von service-basierten Systemen

- Unterschiedliche Security-Levels

Vorteile von service-basierten Systemen

- Unterschiedliche SLAs

Vorteile von service-basierten Systemen

- Keine Big-Bang-Deployments

Vorteile von service-basierten Systemen

- A/B Testing einfach umsetzbar

SEQIS

Vorteile von service-basierten Systemen

- Viel besser skalierbar
- Einzelne Services sind einfach
- Verschiedene Teams arbeiten unabhängig
- Verschiedene Technologien
- Unterschiedliche Security-Levels
- Unterschiedliche SLAs
- Keine Big-Bang-Deployments
- A/B Testing einfach umsetzbar

Microservicearchitecture

- kein zentrales “Steuer”-Service

Microservicearchitecture

- kein zentrales “Steuer”-Service
- deutlich robuster und ausfallsicherer

Microservicearchitecture

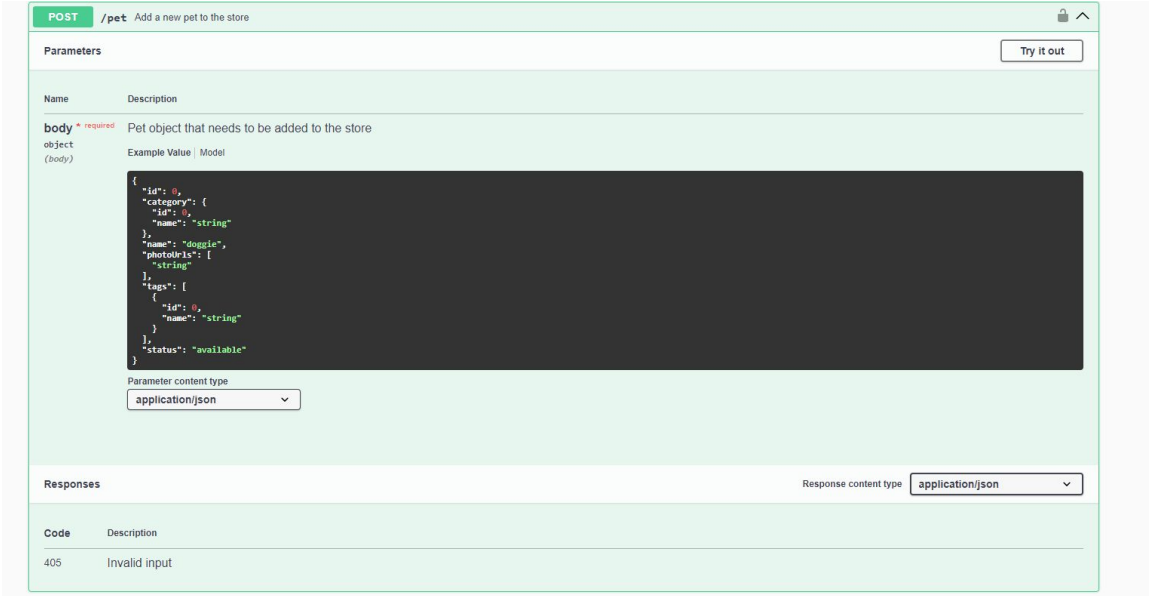
- kein zentrales “Steuer”-Service
- deutlich robuster und ausfallsicherer
- komplexer und schwieriger zu überblicken

SEQIS

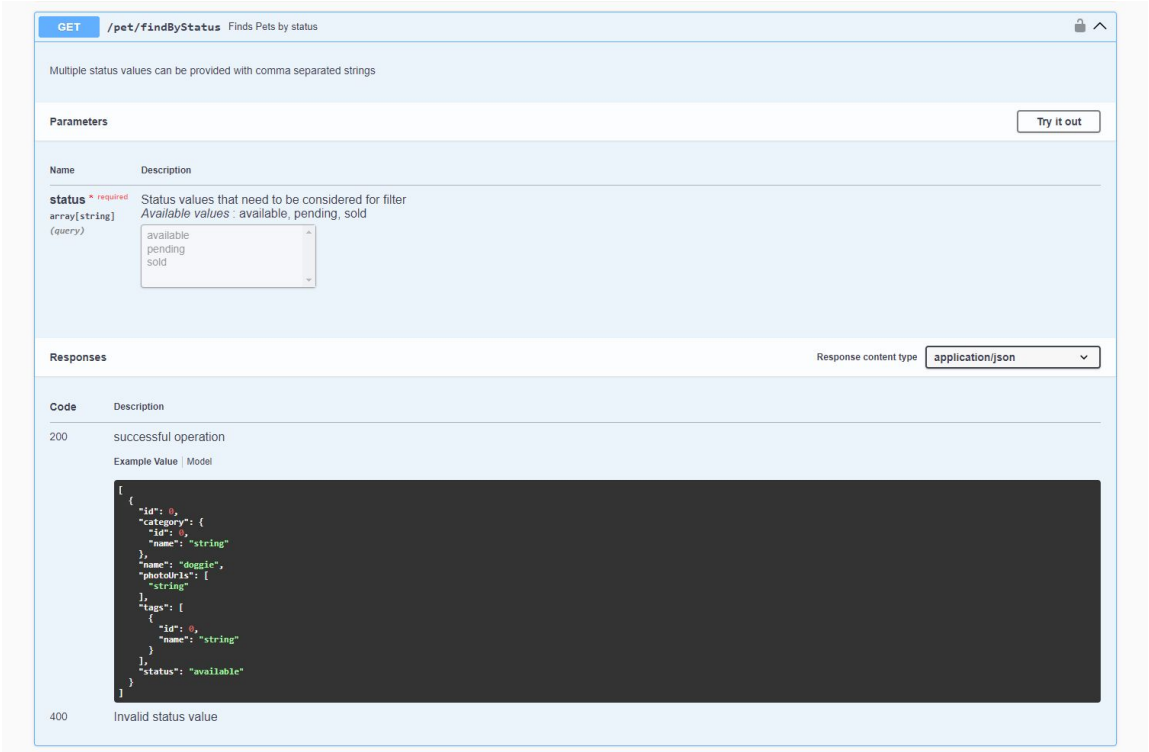
1. Dokumentiere oder noch besser lass den Code dokumentieren!

Swagger

Swagger



Swagger

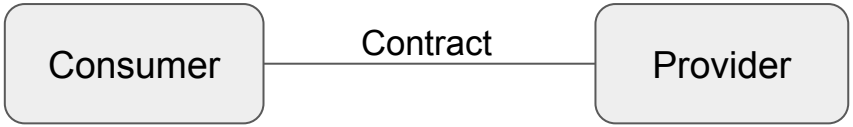


SEQIS

2. Teste die Verträge! Automatisiert!

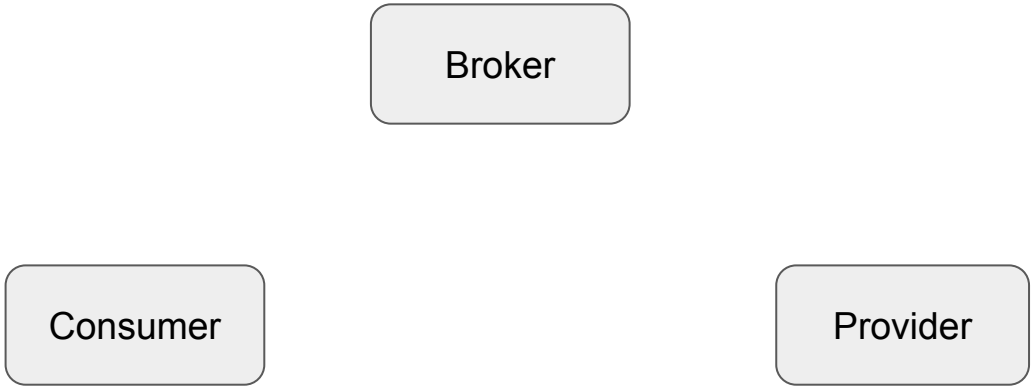
- Contract Testing
- API Testing

Contract Testing



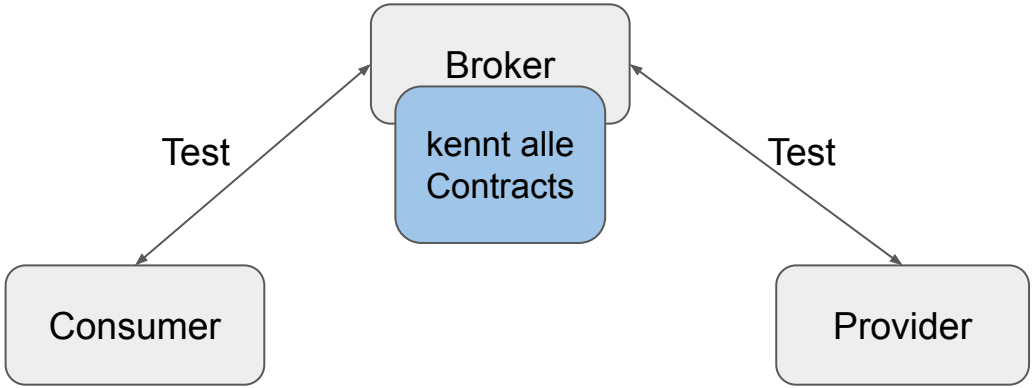
Contract Testing

Beispiel pact.io



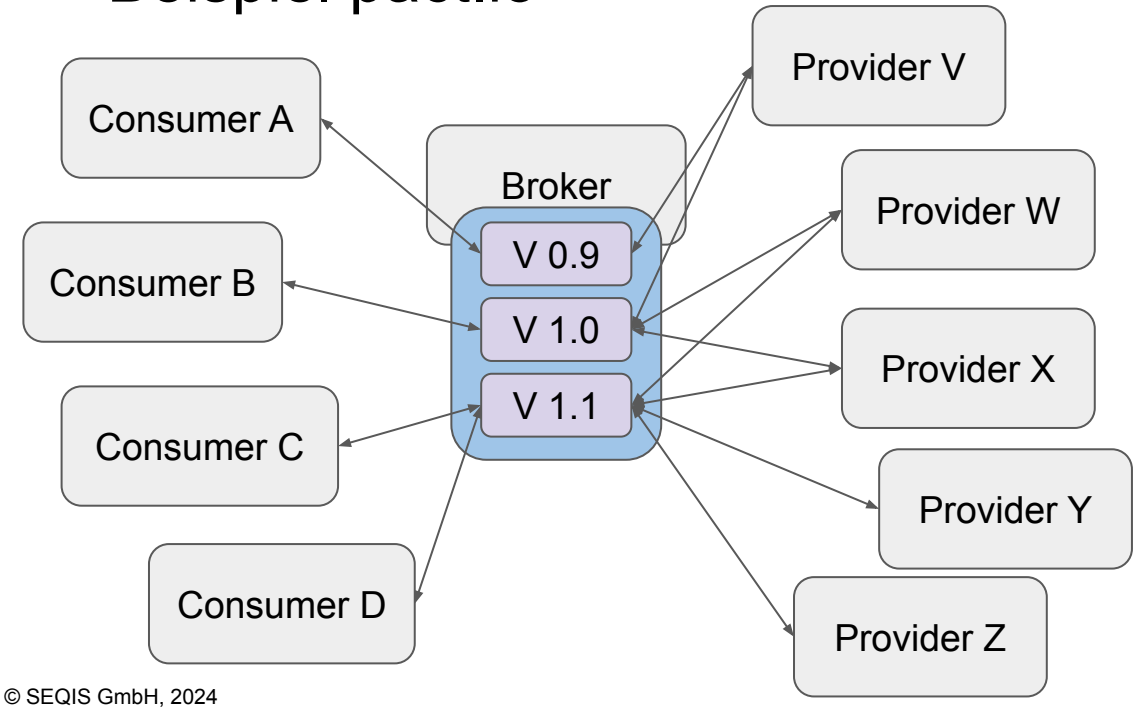
Contract Testing

Beispiel pact.io



Contract Testing

Beispiel pact.io



API-Testing

- Contracts beschreiben keine Logik

SEQIS

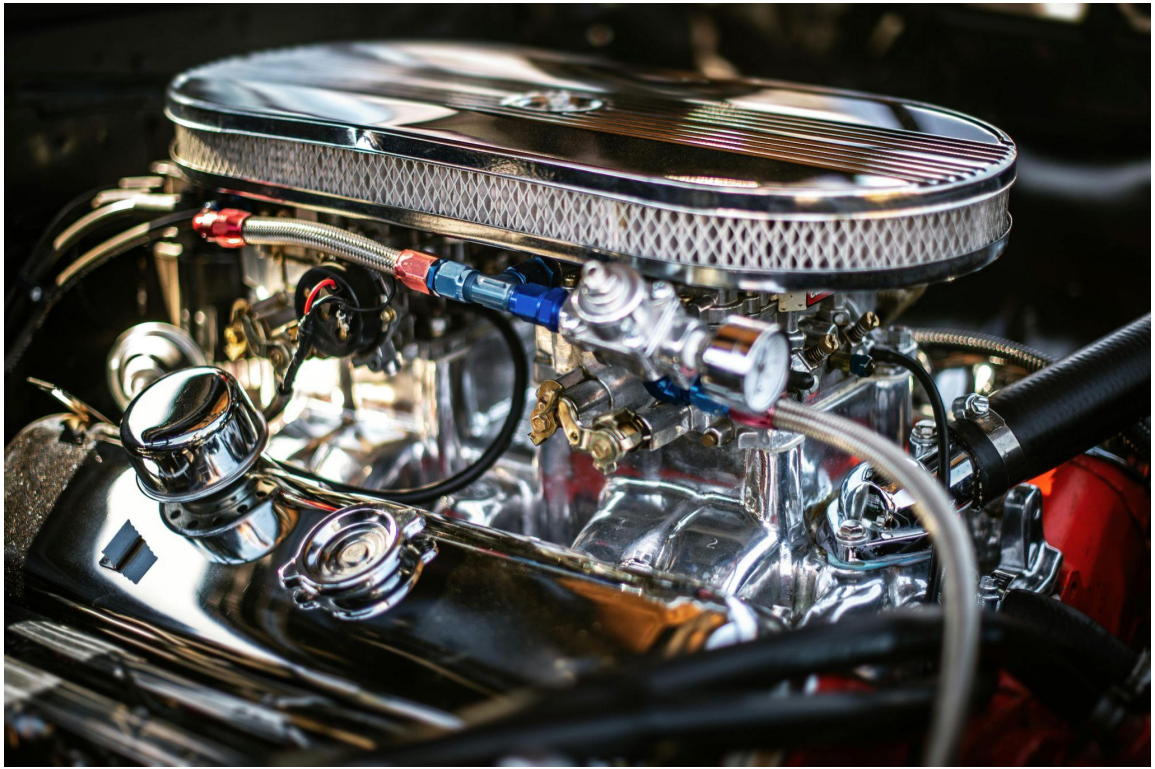
3. Automatisiere dein Deployment!

SEQIS

4. Microservices brauchen kein Mikromanagement!

SEQIS

5. Löse deine Probleme bevor du neue Herausforderungen angehst!



(Micro-) Sevices

- Bieten Lösung für gewisse technische Probleme

(Micro-) Sevices

- Bieten Lösung für gewisse technische Probleme
- Lösen keine organisatorischen Probleme

(Micro-) Sevices

- Bieten Lösung für gewisse technische Probleme
- Lösen keine organisatorischen Probleme
- Lösen keine prozesstechnischen Probleme

(Micro-) Sevices benötigen

- mehr Know-how

(Micro-) Sevices benötigen

- mehr Know-how
- höheren Automatisierungsgrad

SEQIS

(Micro-) Sevices benötigen

- mehr Know-how
- höheren Automatisierungsgrad
- selbständige Teams und Partner mit hohem Qualitätsbewusstsein

SEQIS

(Micro-) Sevices benötigen

- mehr Know-how
- höheren Automatisierungsgrad
- selbständige Teams und Partner mit hohem Qualitätsbewusstsein
- komplexere Infrastruktur

SEQIS

6. Führe Quality Gates für Services ein!

Quality Gates

- Test Coverage

Quality Gates

- Test Coverage
- Statische Codeanalyse

SEQIS

Quality Gates

- Test Coverage
- Statische Codeanalyse
- Security Tests

Quality Gates

- Test Coverage
- Statische Codeanalyse
- Security Tests
- nicht-funktionale Tests

Unterschiedlich für jedes Service

- intern vs. extern (Entwicklung)

Unterschiedlich für jedes Service

- intern vs. extern (Entwicklung)
- intern vs. extern (Benutzer)

Unterschiedlich für jedes Service

- intern vs. extern (Entwicklung)
- intern vs. extern (Benutzer)
- Kritikalität des Services

Unterschiedlich für jedes Service

- intern vs. extern (Entwicklung)
- intern vs. extern (Benutzer)
- Kritikalität des Services
- Zugriffsmöglichkeiten

SEQIS

7. Mache Last- und Performancetests sowohl für die einzelnen Services als auch für das Gesamtsystem.

8. Viele Wege führen nach Rom, nimm den richtigen!

9. Nicht ohne triftigen Grund Micro

10. Kein Gold Plating

Modulare Entwicklung