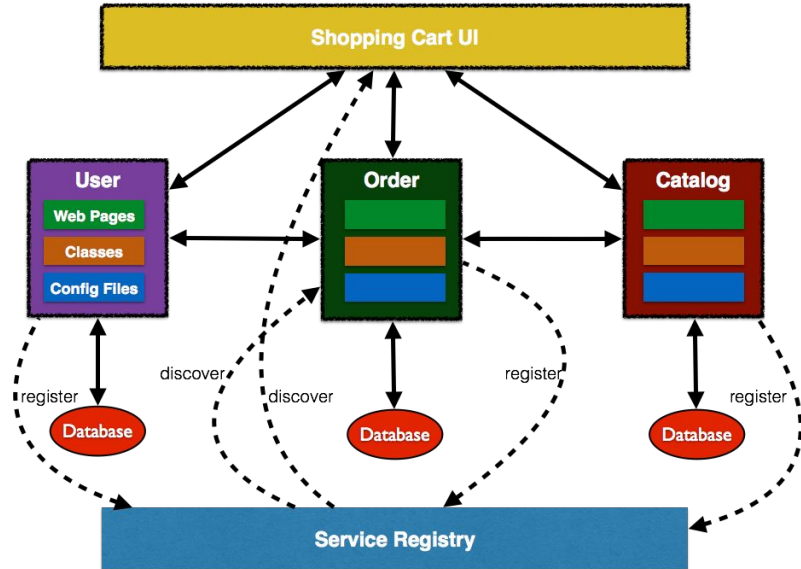
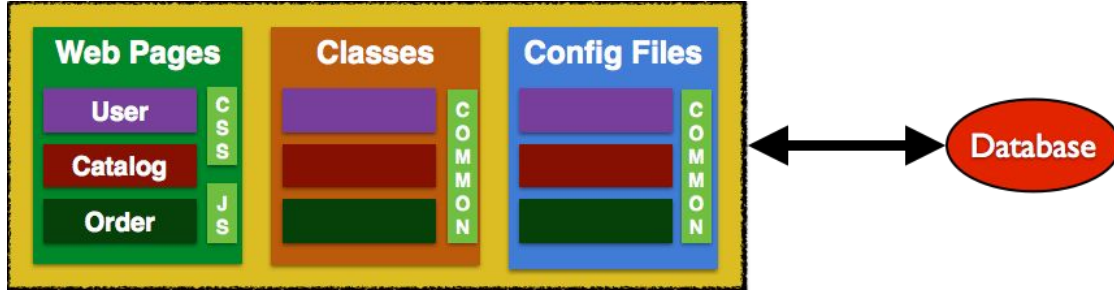


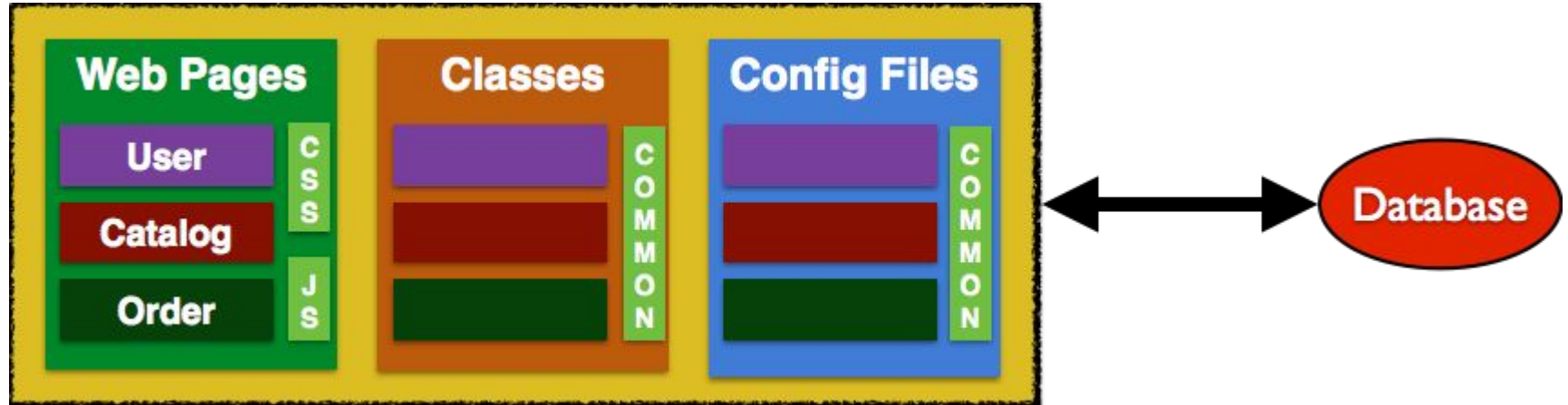


Microservices

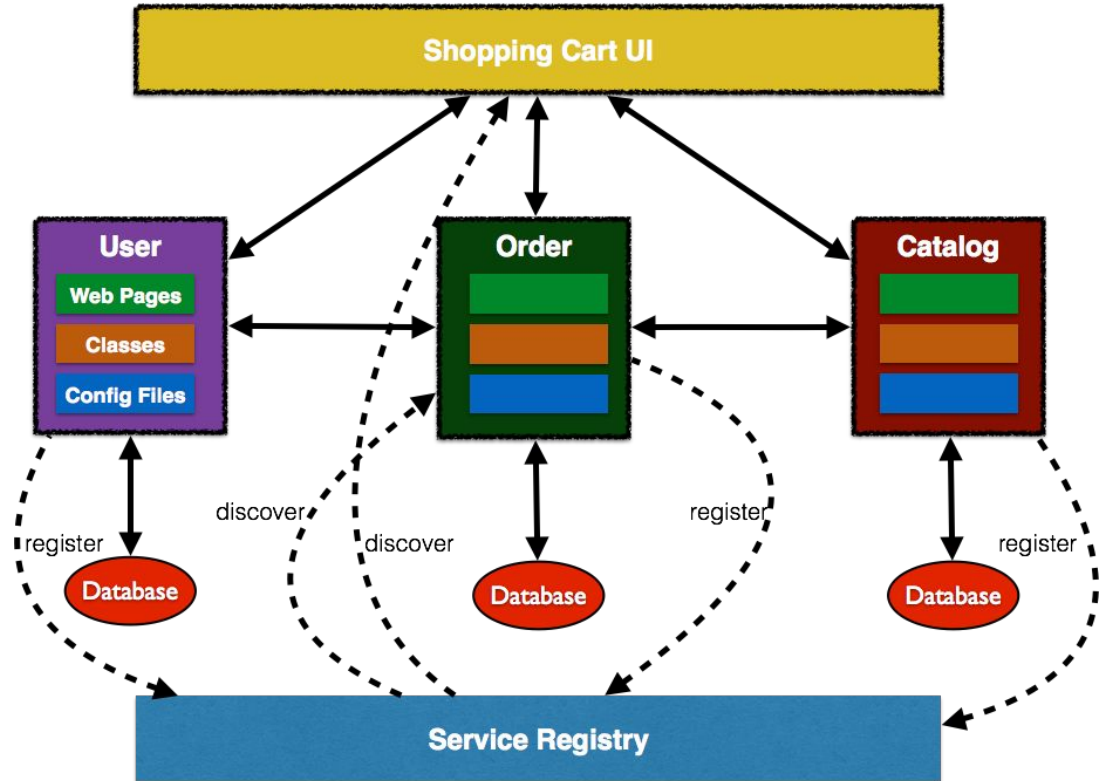
Do one thing and do it well!



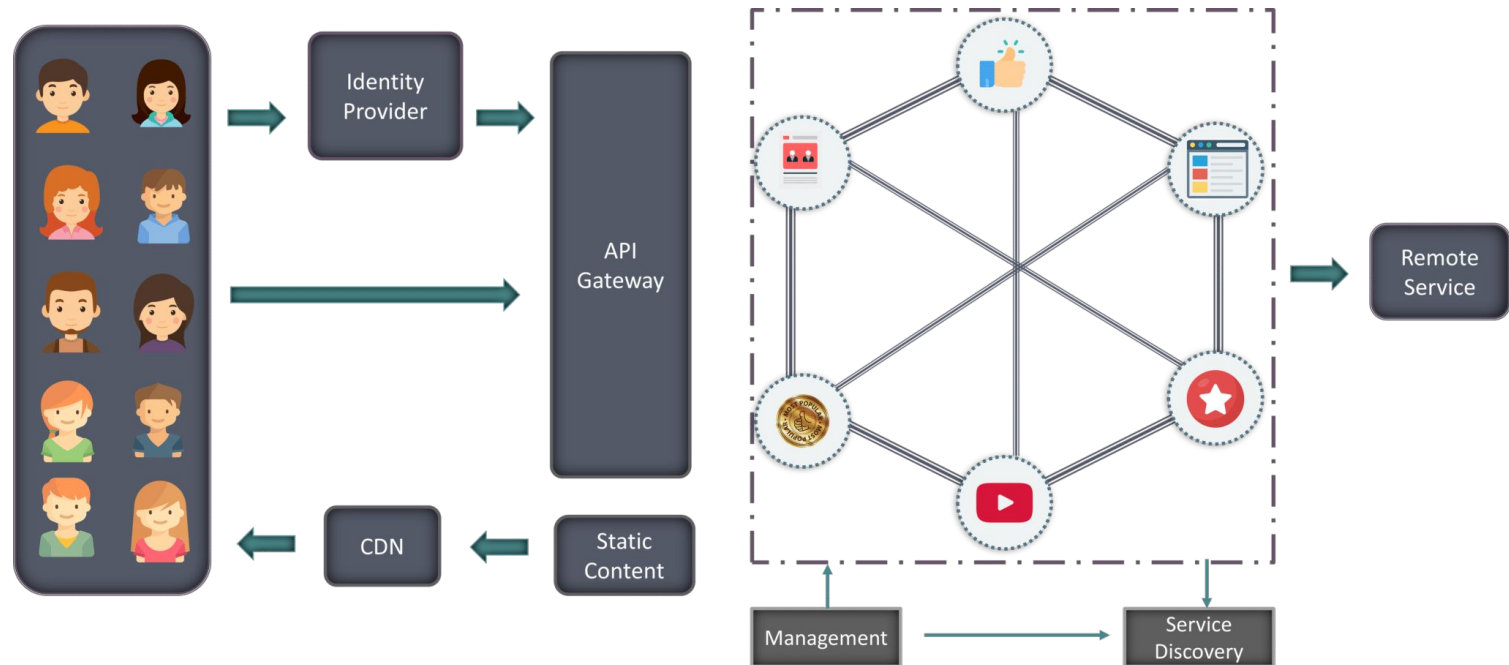
Monolith Architektur



Microservice



Wie funktionieren Microservices





Vorteile von Microservices

- Leicht zum Erstellen und Warten
- Flexible in der Auswahl der Tools
- Higher quality code
- Leichter Skalierbar
- Weniger Fehleranfällig
- Verwenden eine geringe Anzahl an Ressourcen in der Entwicklung
- Fördern die Verwendung von Big Data



Nachteile

- Netzwerklatenz
- komplexere Softwareverteilung und Tests
- komplexeres Logging und Monitoring
- schwere bzw. aufwändige Migration
- Mögliche Probleme mit der Datenkonsistenz



Firmen die Microservices verwenden

- Google
- Spotify
- Amazon
- Twitter
- The Guardian
- Netflix
- SoundCloud
- eBay
- ...



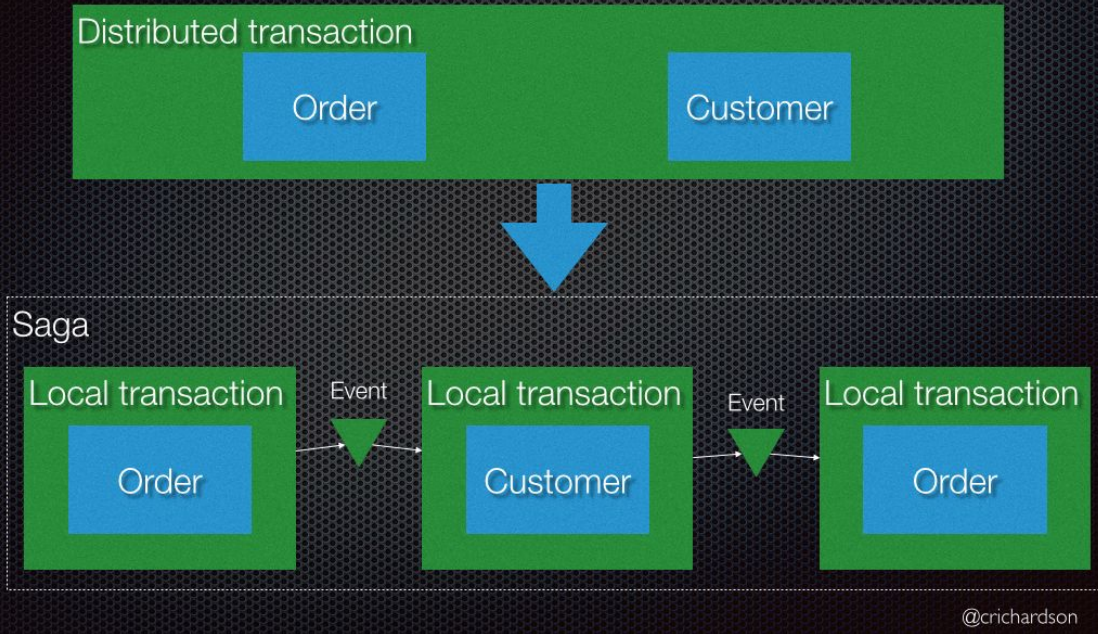
Wie erhält man Datenkonsistenz aufrecht

- 2-Phase-Commit
- Sagas

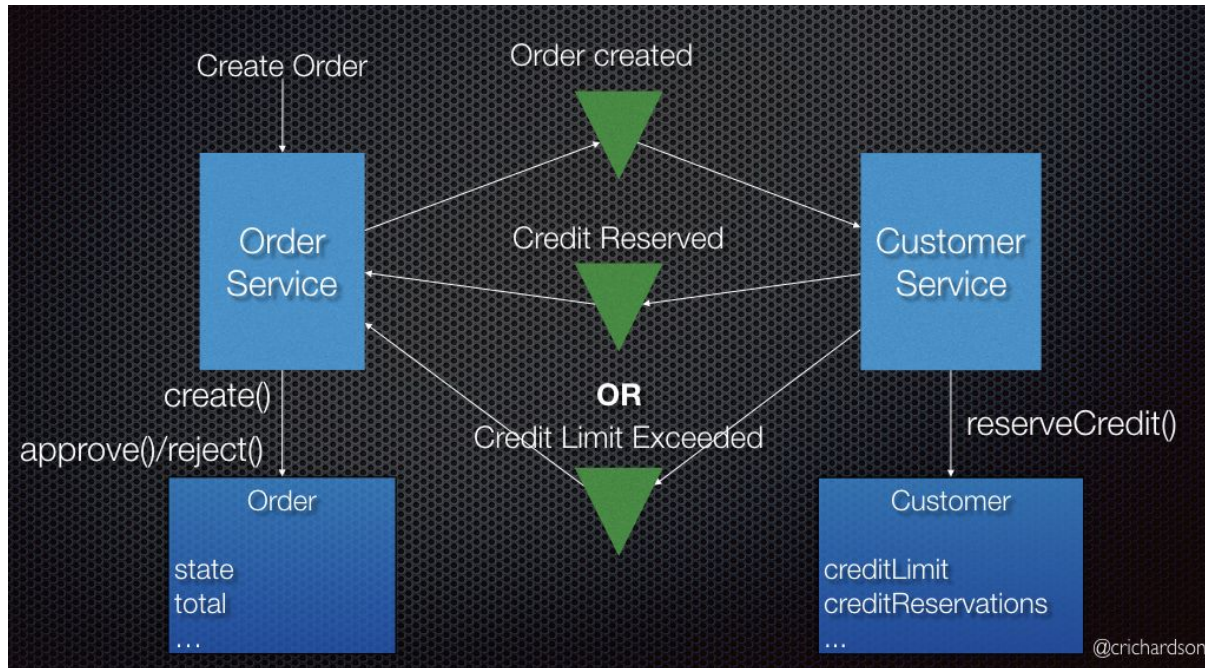
Sagas

- Choreography
- Orchestration

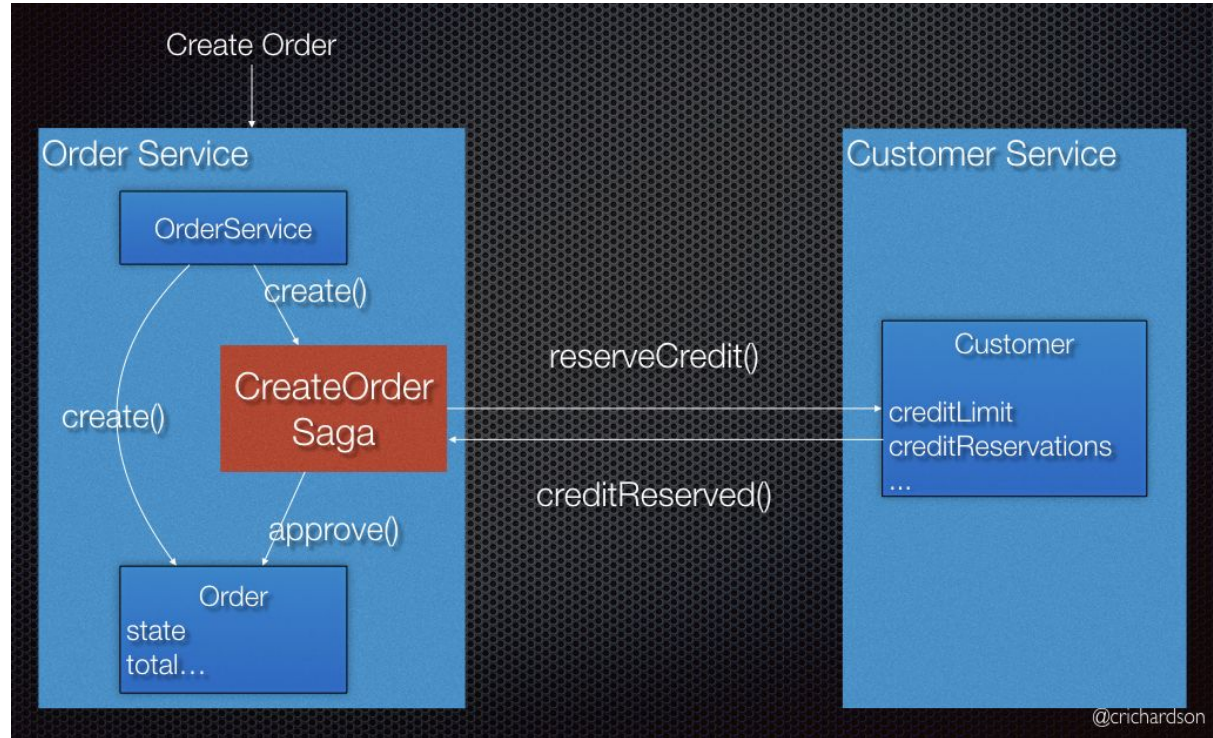
Using Sagas instead of 2PC

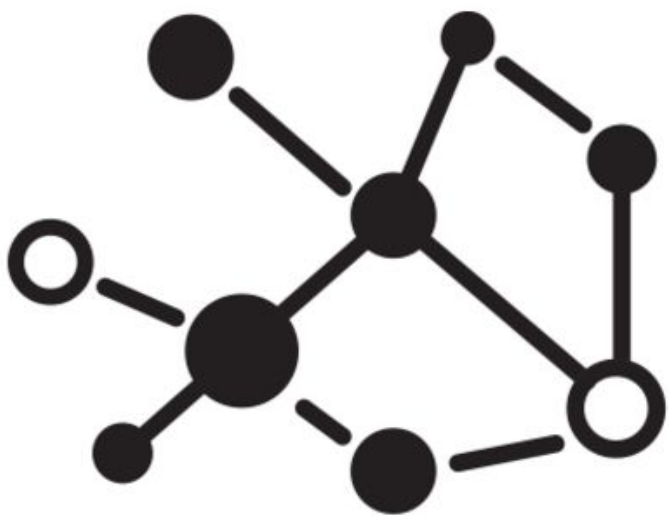


Choreography



Orchestration





Eclipse
MicroProfile



Was ist Microprofile

- Gründer: Red Hat, IBM, TomiTribe, Payara & LJC
- Alternative zu JEE Web bzw. Full profiles
- Community-driven
- Standardisiert



Das Ziel

“An open forum to optimize Enterprise Java for a microservices architecture by innovating across multiple implementations and collaborating on common areas of interest with a goal of standardization.”



Komponenten

- MicroProfile Config 1.3
- MicroProfile Metrics 1.1
- MicroProfile OpenAPI 1.0
- MicroProfile Rest Client 1.1
- CDI 2.0
- Common Annotations 1.3
- JAX-RS 2.1
- ...



einige Standards

- Rest: consumes/produces Json-P
- kein JPA in microProfile



Quellen

<https://hackernoon.com/understand-service-discovery-in-microservice-c323e78f47fd>

<https://www.edureka.co/blog/microservices-tutorial-with-example>

<https://www.edureka.co/blog/top-10-reasons-to-learn-microservices/>

<http://www.jasondl.ee/posts/2018/getting-started-with-eclipse-microprofile-part-3-thorntail.html>