Microservices

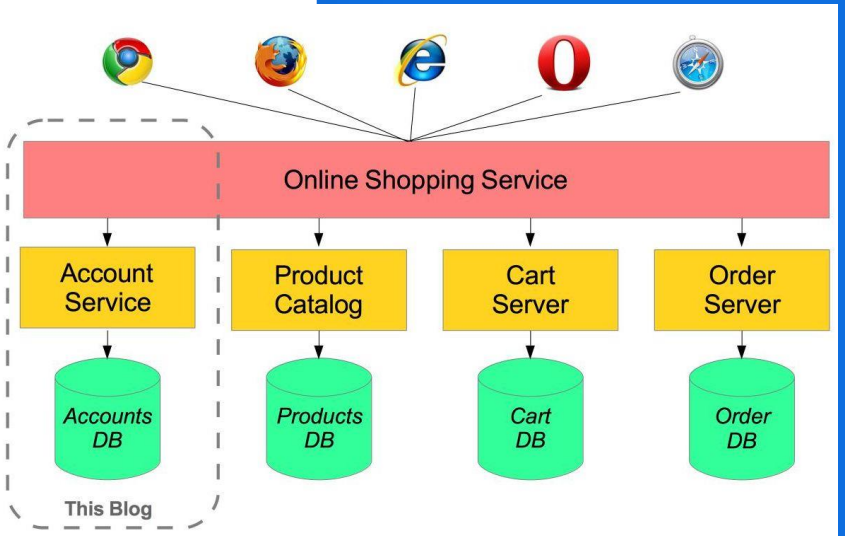
Zaawansowane Zagadnienia Programowanie w Javie – Edycja 2023

Monolith

Shopping system without Microservices (Monolith architecture). In this architecture we are using Monolith architecture i.e. all collaborating components combine all in one application

What is Microservices Architecture?

- Microservices architecture allows to avoid monolith application for large system. It provide loose coupling between collaborating processes which running independently in different environments with tight cohesion.
- Microservices allows us to break our large system into number of independent collaborating processes.
- For example imagine an online shop with separate microservices for user-accounts, product-catalog order-processing and shopping carts



Microservice’s characteristic

Loose coupling

- application build from collaboration services or processes, so any process change without effecting another processes
- effect of changes isolated

Tight cohesion

- an individual service or process that deals with a single view of data
- code perform a single well defined task

Microservice Benefits

- Smaller code base is easy to maintain
- Easy to scale as individual component
- Technology diversity i.e. mixing libraries, databases, frameworks etc.
- Fault isolation i.e. a process failure should not bring whole system down
- Better support for smaller and parallel team
- Independent deployment
- Deployment time reduce

Microservice Challenges

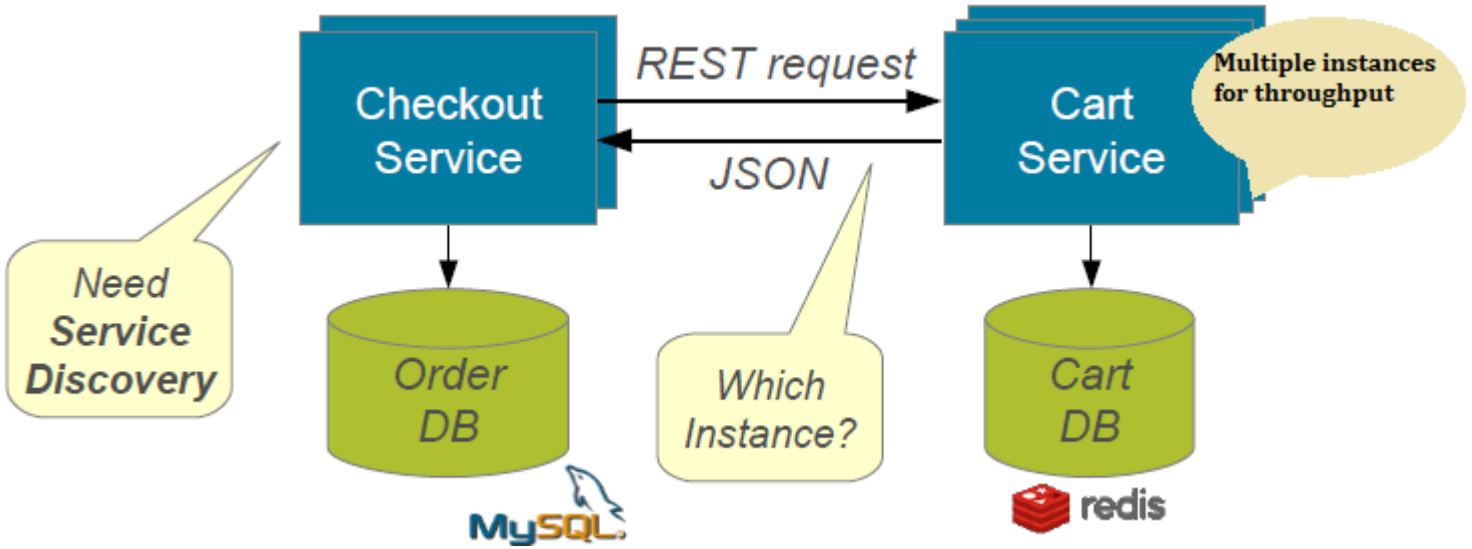
- Difficult to achieve strong consistency across services
- ACID transactions do not span multiple processes
- Distributed System so hard to debug and trace the issues
- Greater need for end to end testing
- Required cultural changes in across teams like Dev and Ops working together even in same team

Spring Cloud and Discovery server

- building blocks for Cloud and Microservices
- provides microservices infrastructure: i.e provides use services such as Service Discovery, Configuration server and Monitoring.
- provides several other open source projects like Netflix OSS.
- uses Spring Boot style starters
- provides Platform as a Service like AWS

Spring Cloud supports

- Cloud Integration
- Dynamic Reconfiguration
- Service Discovery (How do services find each other?)
- Security
- Client-side Load Balancing (How do we decide which service instance to use?)

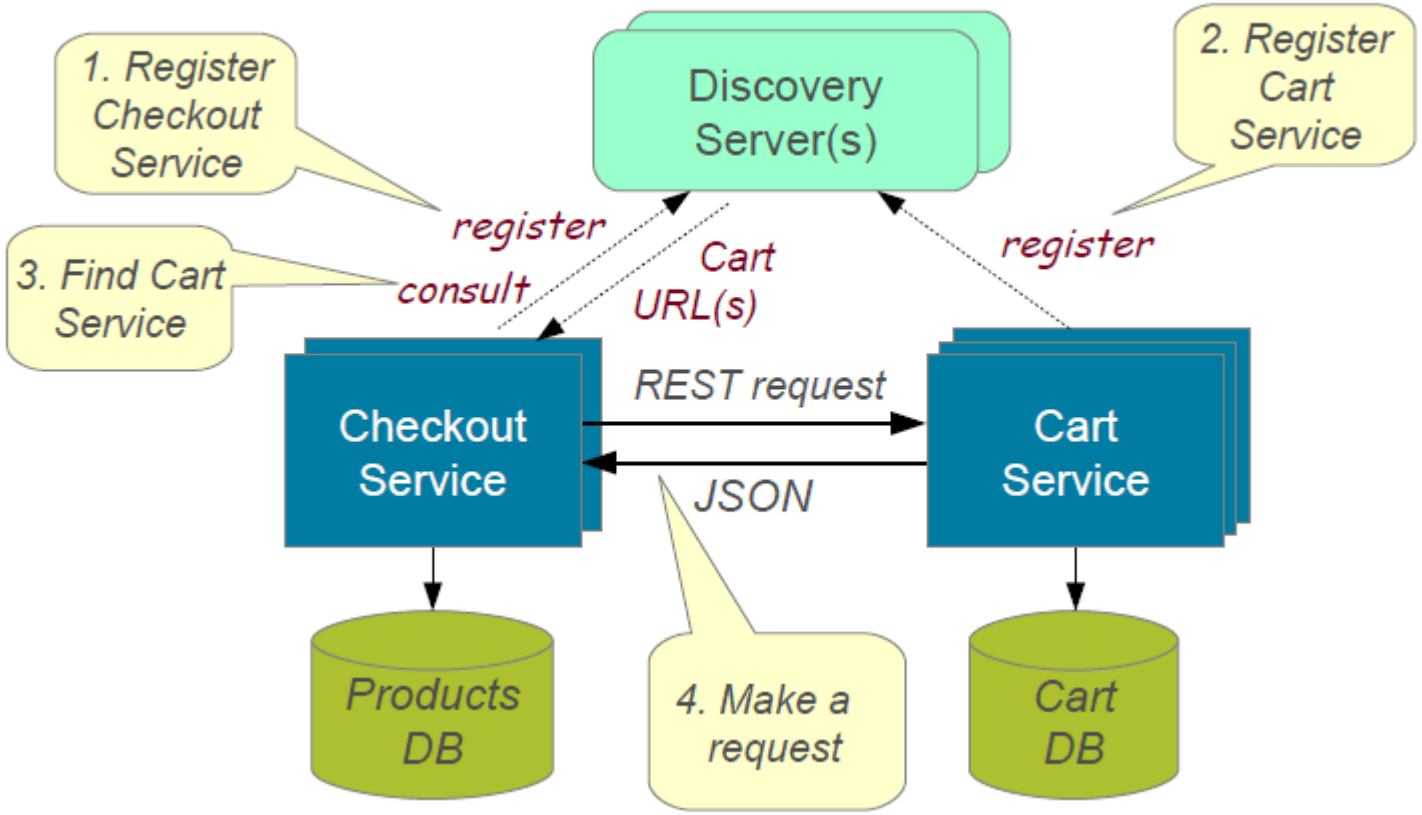


Service Discovery

Problem without discovery:

- Finding right services
- Running multiple instances for a service

Implementing Service Discovery



Rozkład jazdy