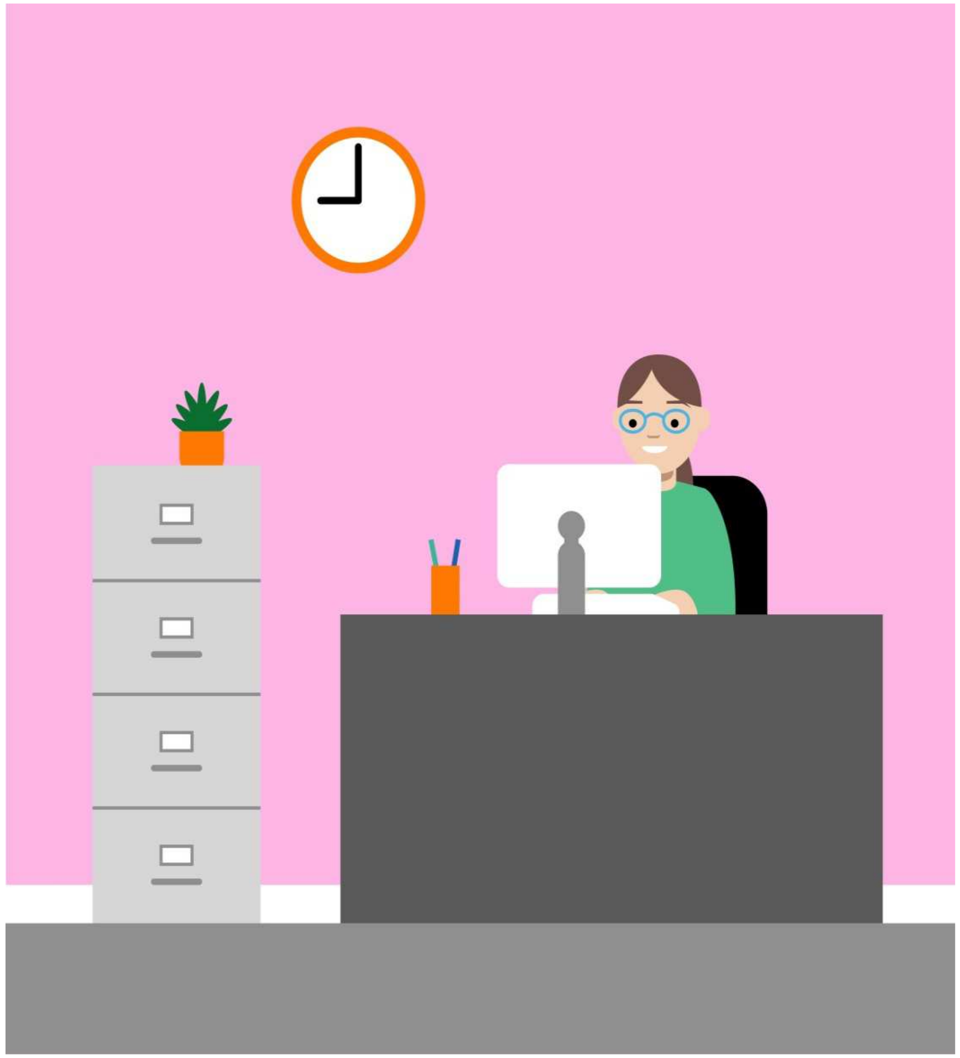
Microservices

What tools do we use?

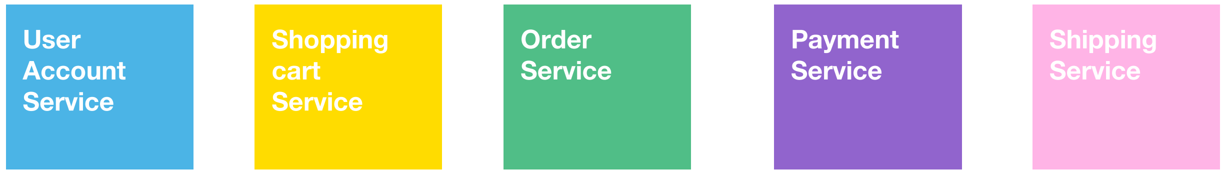
Iuliana Condoiu - Service Platform Developer at Orange Services

Contents



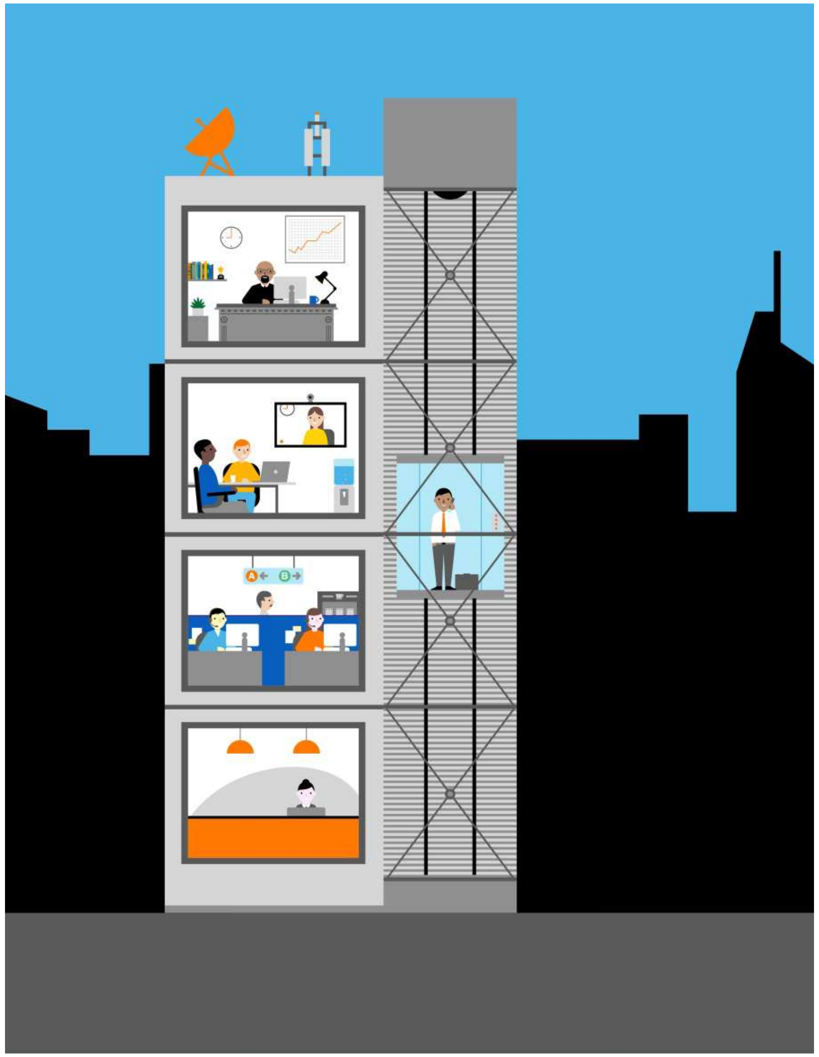
Microservices architecture (MSA)

- Small services
- Highly decoupled
- Modular
- One task per service



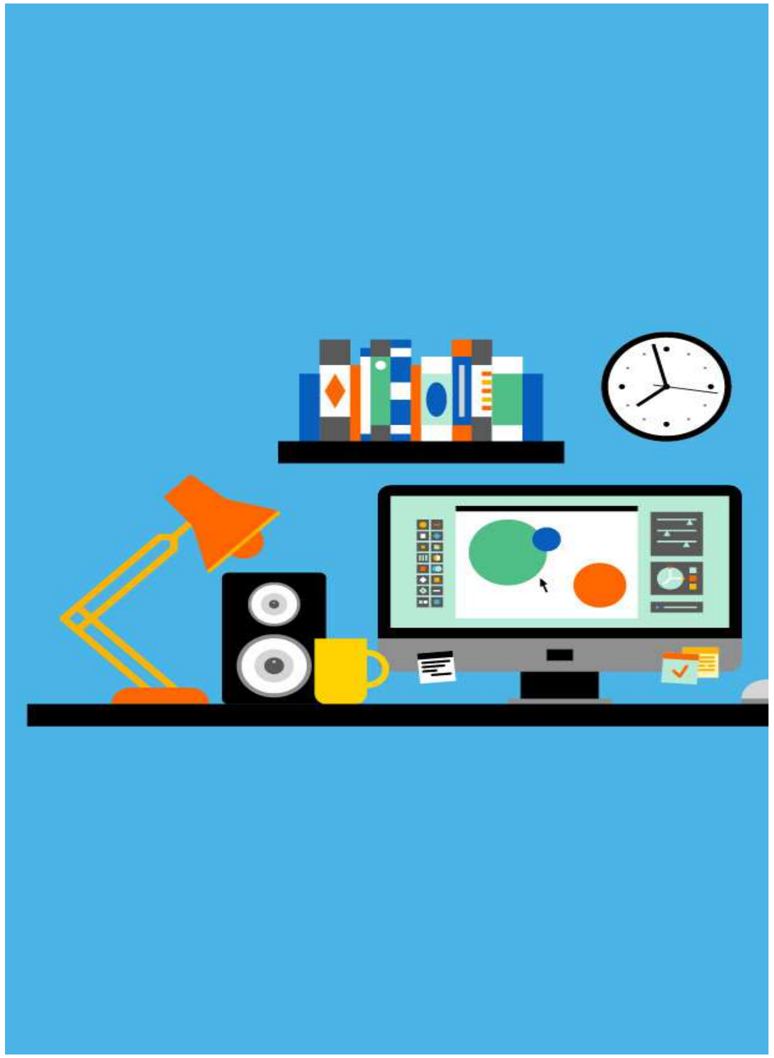
MSA-advantages

- Small modular services
- Services are easy to change/replace
- Scalability
- Independent development and deployment
- Fault isolation
- Continuous Integration
- Continuous Delivery

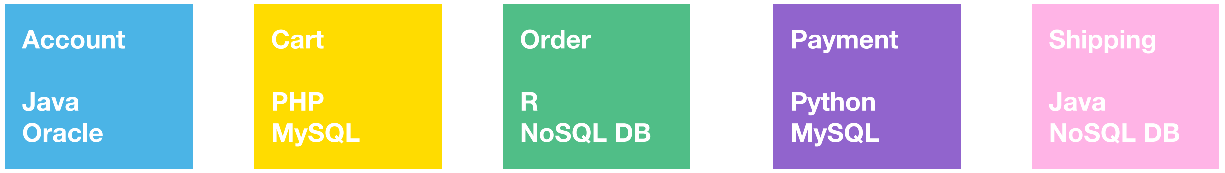


MSA-disadvantages

- Deployment complexity
- Adding communication layer
- Expensive remote calls
- Hard to test use cases that span multiple services
- Increase memory consumption



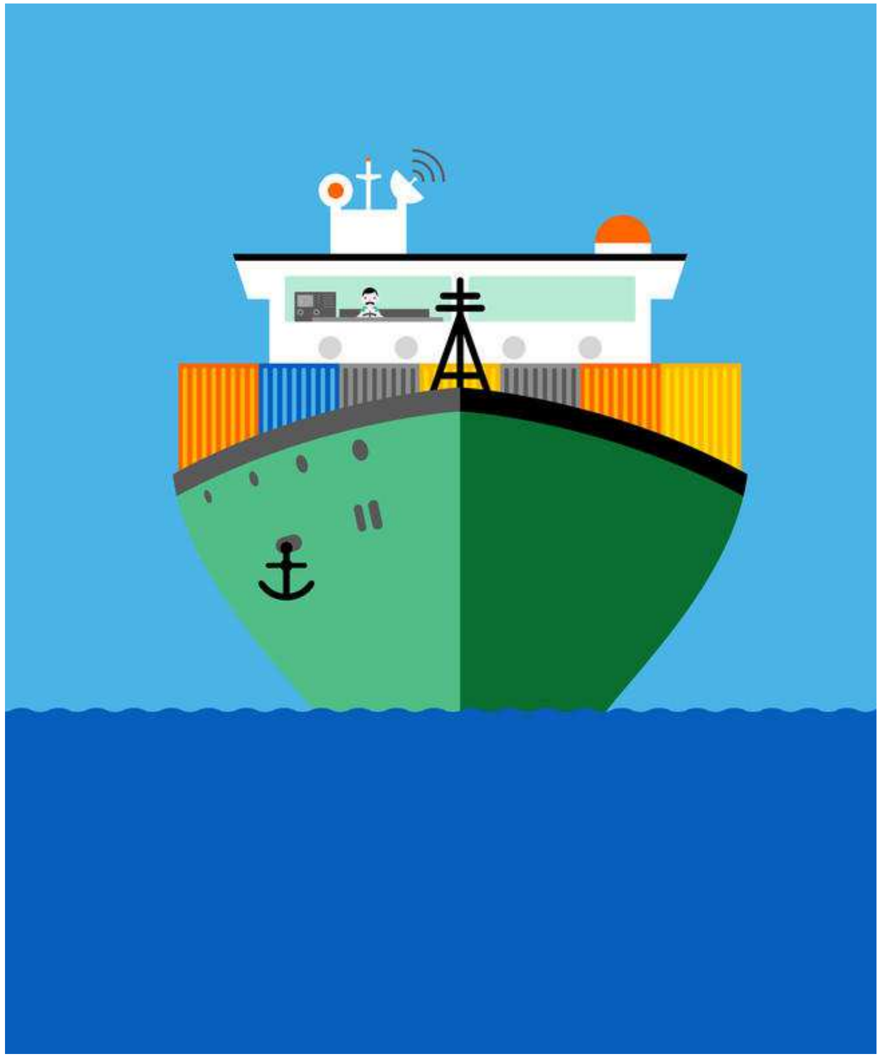
Development



- Standalone application
- Business oriented
- Single Responsibility Principle (SRP)
- Independent development and deployment
- Local data storage

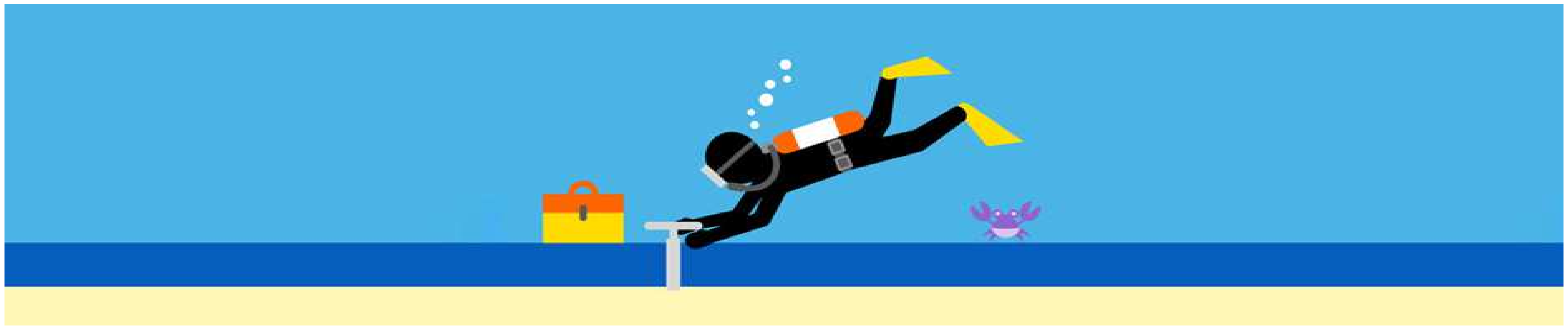
Integration

- Simple communication
- Synchronous: REST, Thrift
- Asynchronous: AMQP, STOMP, MQTT
- Message format: JSON, Thrift, Avro
- Service contracts: Swagger, RAML, Thrift IDL

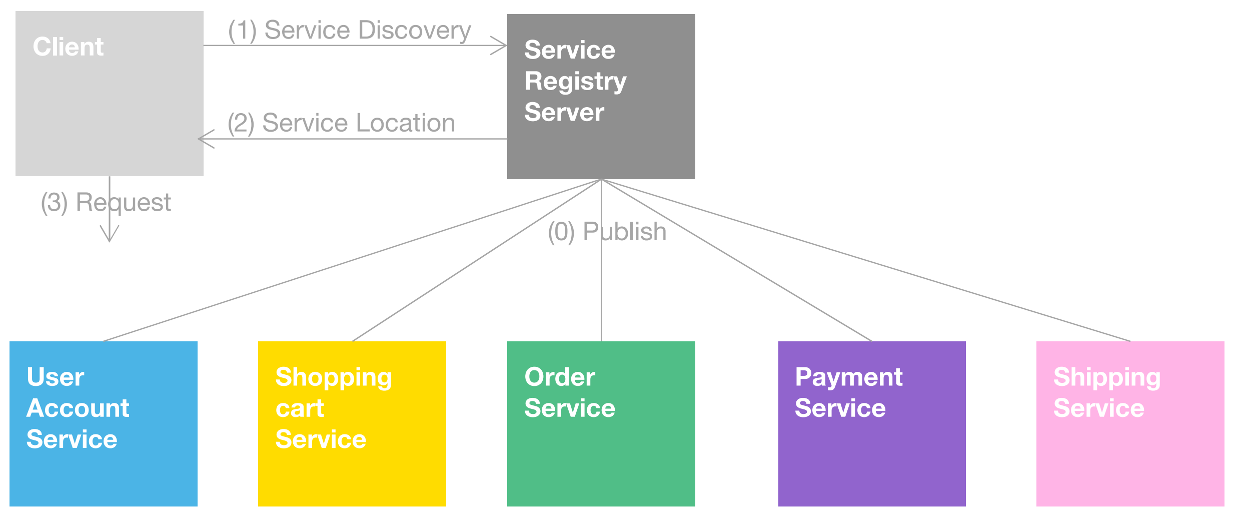


Service Registry and Service Discovery

- High number of MS
- Dynamic nature of locations
- Service Registry – holds MS instances and locations
- Service Discovery – makes sure that SR has real time data
  - ✓ Client Service Discovery
  - ✓ Server Service Discovery

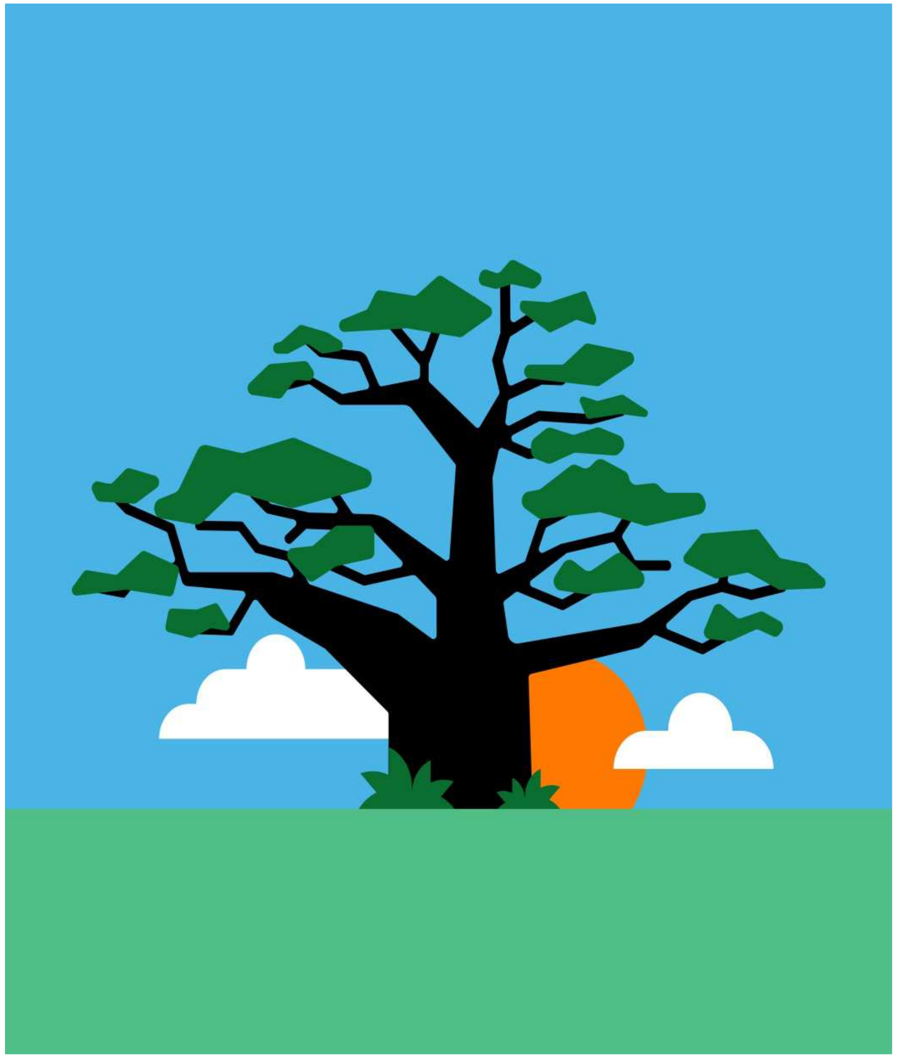


Service Registry and Service Discovery



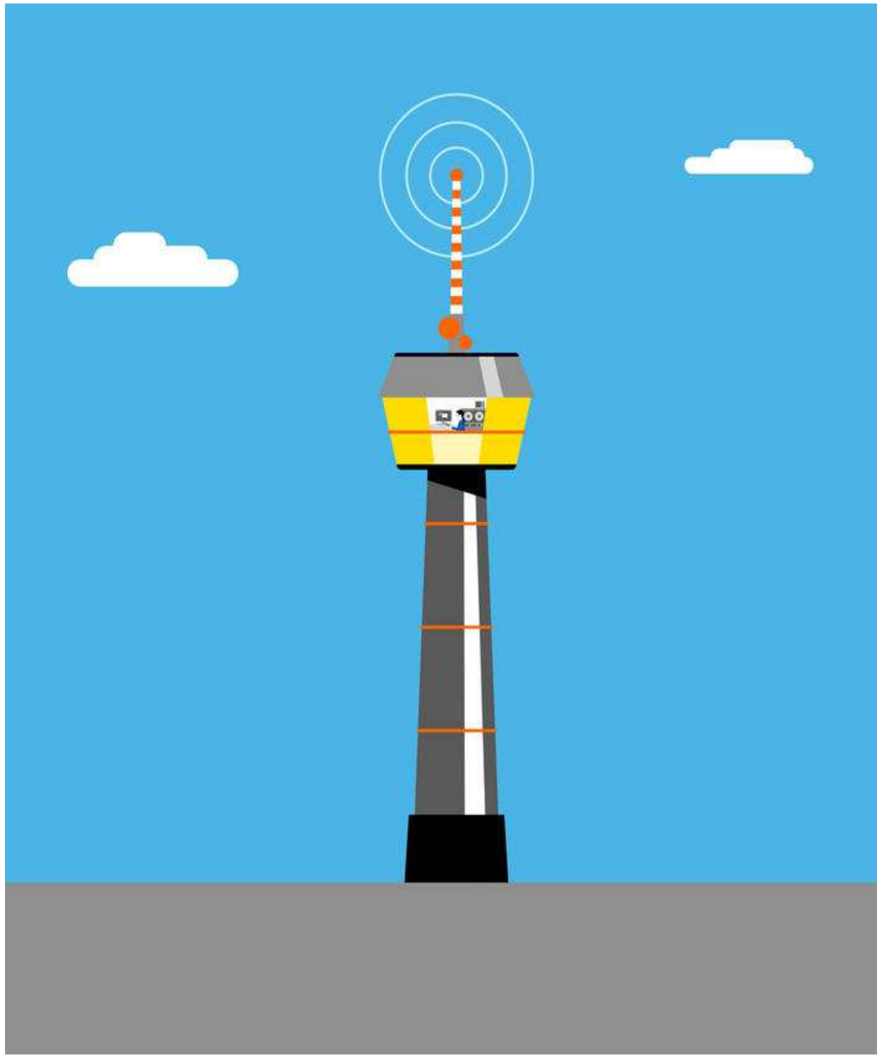
Spring Cloud

- Discovery server (Eureka / Consul)
  - ✓ Spring Boot application
  - ✓ @EnableEurekaServer
- Microservice definition
  - ✓ Spring Boot application
  - ✓ @EnableDiscoveryClient
- Microservice consumption
  - ✓ RestTemplate
  - ✓ serviceUrl – logical host
- Distributed configuration

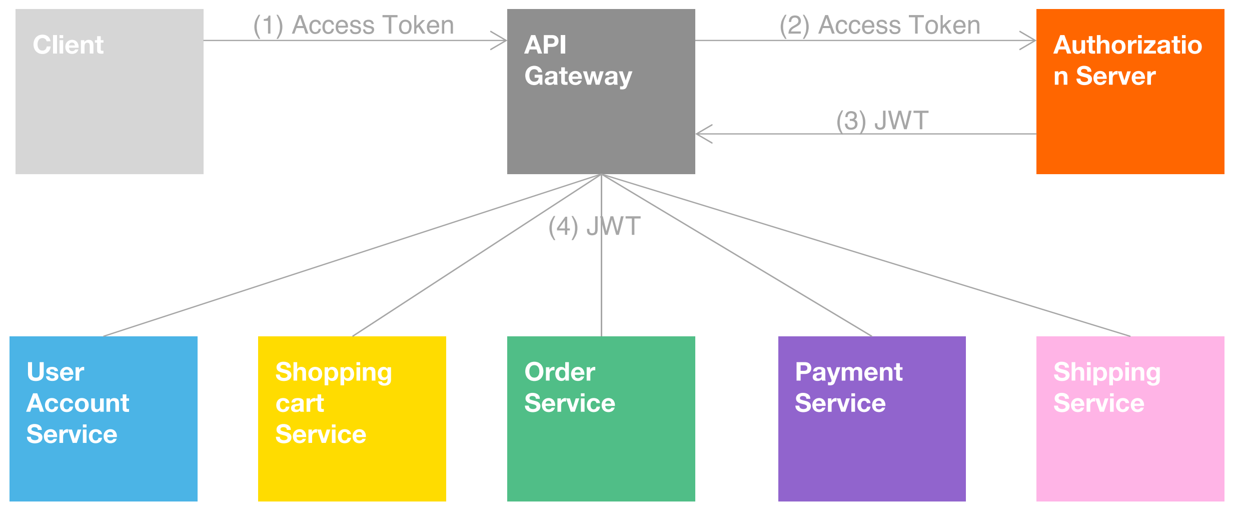


Security

- Authentication & Authorization
- At each MS level
- OAuth2 – access delegation protocol
  (access token)
- Open ID Connect – access token + ID token
- JWT (JSON Web Token)



Security


Client
(1) Access Token
API Gateway
(2) Access Token
Authorization Server
(3) JWT
(4) JWT
User Account Service
Shopping cart Service
Order Service
Payment Service
Shipping Service

Deployment

- Distributed systems
- Containers
- Virtualization

- Dynamic endpoints
- Scale up/down

Docker

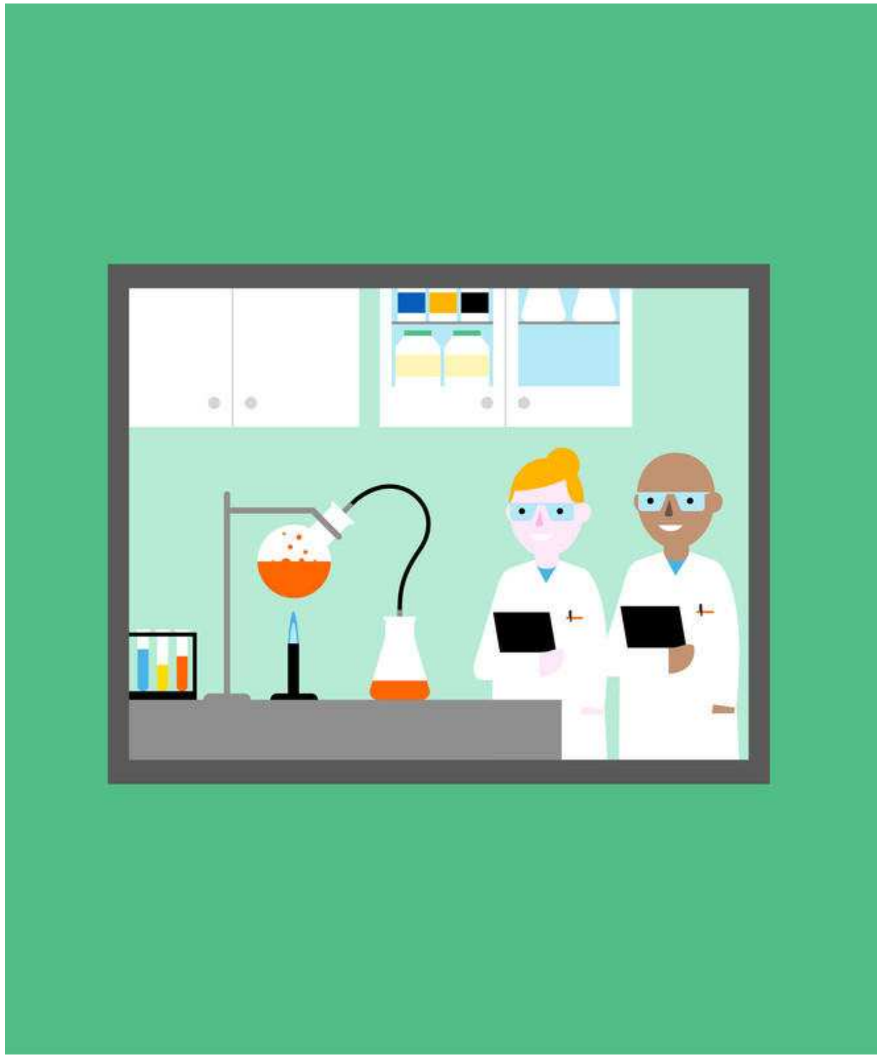
- OS level virtualization
- Highly efficient distribution model
- State encapsulation of an application (environment independency)
- Best-of-breed containers – OS community
- DevOps oriented

Kubernetes

- Cluster of Docker containers
- Colocation of containers
- Service discovery
- Replication control

Testing

- Unit testing ~ REST API testing
  - ✓ vREST, SoapUI, RobotFramework…
- Contract testing
  - ✓ Pact, Pacto
- End-to-end testing



Thank you!