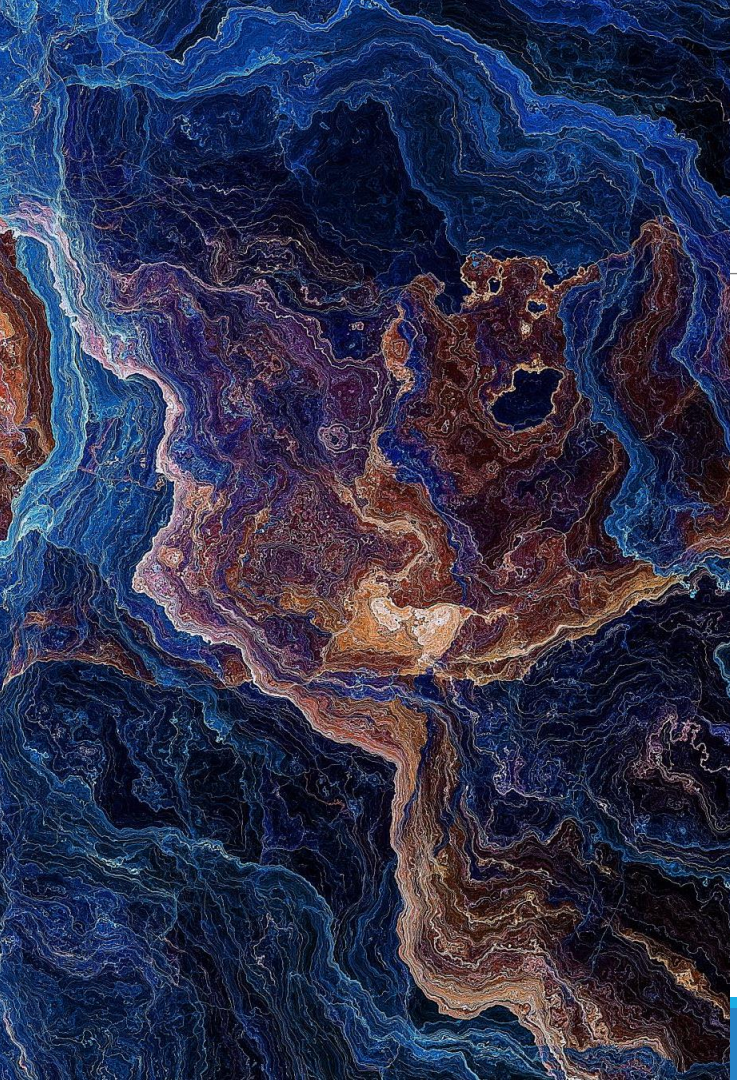
Microservices

MOCHAMMAD ZEN SAMSONO HADI, ST. MSC. PH.D

Kenapa Perlu Belajar Microservices?

- Banyak Digunakan di Tech Company
- Sudah Jadi Pengetahuan untuk Engineer

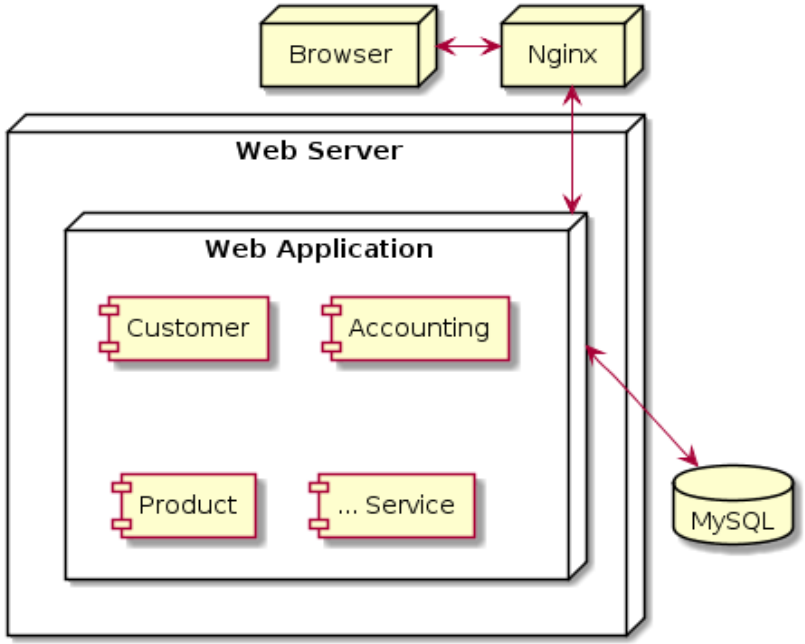


Arsitektur Monolith

Apa itu Arsitektur Monolith?

- Single Deployment Unit
- Dimana semua fitur dibuat dalam sebuah aplikasi besar

Arsitektur Monolith



Kelebihan Arsitektur Monolith

- Mudah di Develop
- Mudah di Deploy
- Mudah di Test
- Mudah di Scale

Masalah di Arsitektur Monolith

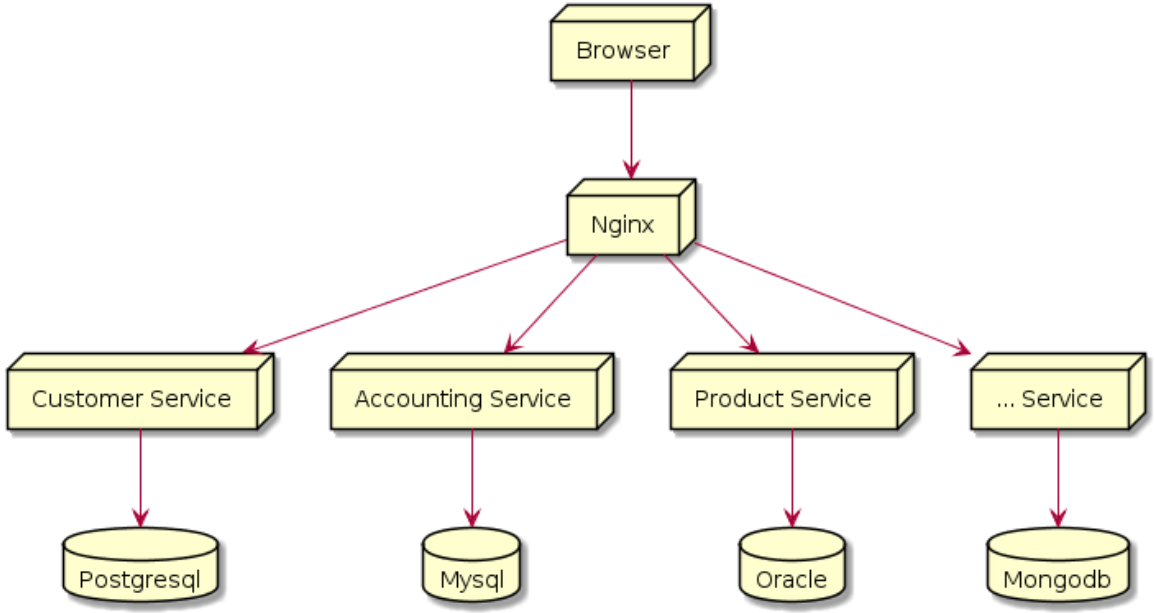
- Mengintimidasi Developer yang baru bergabung
- Scaling development dengan banyak Developer agak menyulitkan
- Butuh kontrak panjang dengan teknologi yang digunakan (bahasa pemrograman, database, dan lain-lain)
- Scaling pada bagian tertentu tidak bisa dilakukan
- Running app Monolith sangat berat

Arsitektur Microservice

Apa itu Arsitektur Microservices

- Aplikasi-aplikasi kecil yang saling bekerja sama.
- Fokus mengerjakan satu pekerjaan dengan baik
- Independent, dapat di deploy dan diubah tanpa tergantung dengan aplikasi lain
- Setiap komponen pada sistem dibuat dalam service
- Komunikasi antar service biasanya melalui network-call

Arsitektur Microservices



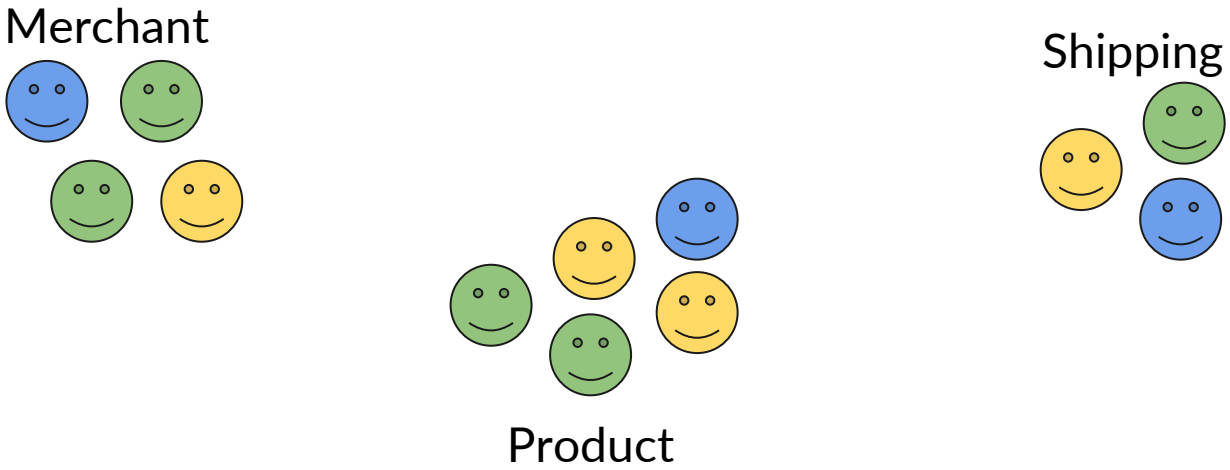
Kelebihan Arsitektur Microservices

- Mudah dimengerti, karena relative kecil ukuran service nya
- Lebih mudah di develop, di maintain, di test dan di deploy
- Lebih mudah bergonta-ganti teknologi
- Mudah di scale sesuai kebutuhan
- Bisa dikerjakan dalam tim-tim kecil

Masalah di Arsitektur Microservices

- Distributed system
- Komunikasi antar service yang rawan error
- Testing interaksi antar service lebih sulit

Pembagian Aplikasi Microservices



Seberapa Kecil Aplikasi Microservices?

- Single responsibility
- Sekecil mungkin sehingga bisa dimengerti oleh satu orang
- Bisa di kerjakan sejumlah X developer

Monolith

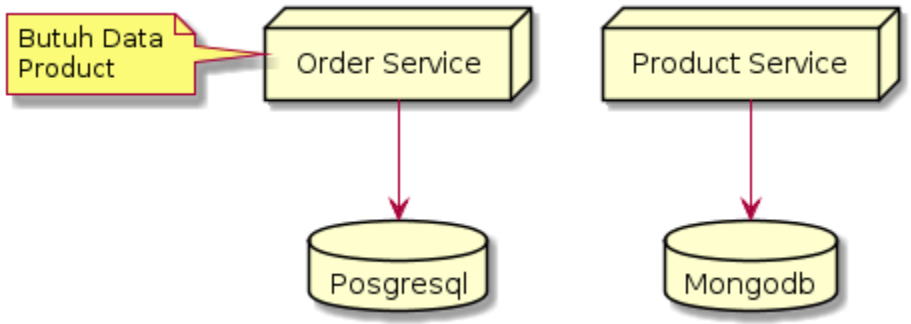
- Simplicity
- Consistency
- Easy to Refactor

Microservices

- Partial Deployment
- Availability
- Multiple Platform
- Easy to Scale

Remote Procedure Invocation

Ketika Service butuh Data Service Lain



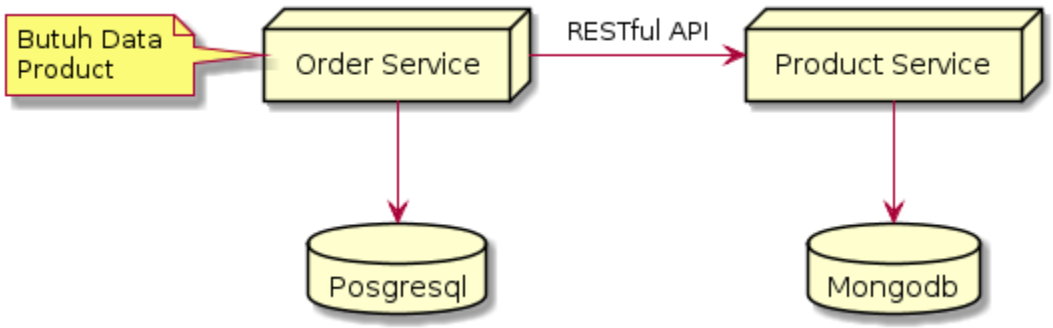
Komunikasi Antar Service

- Idealnya komunikasi dilakukan melalui RPI (Remote Procedure Invocation) atau RPC (Remote Procedure Call)
- Tidak direkomendasikan komunikasi dilakukan via database

Contoh Remote Procedure Invocation

- RESTful API (HTTP)
- gRPC
- Apache Thrift
- SOAP
- Java RMI
- Corba (Common Object Request Broker Architecture)
- dan lain-lain

Ketika Service butuh Data Service Lain



Keuntungan Menggunakan RPI

- Sederhana dan Mudah
- Biasanya digunakan untuk komunikasi Request - Reply
- Biasanya digunakan untuk proses Sync (yang butuh menunggu jawaban)

Type of Microservices

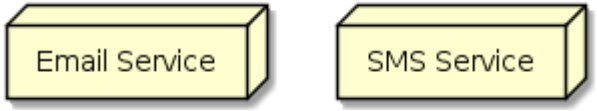
Tipe Microservices

- Stateless Microservice
- Persistence Microservice
- Aggregation Microservices

Stateless Microservices

- Biasanya tidak memiliki database
- Digunakan untuk melakukan tugas sederhana
- Biasa digunakan juga sebagai utility untuk microservice lain
- Tidak bergantung dengan microservice lain

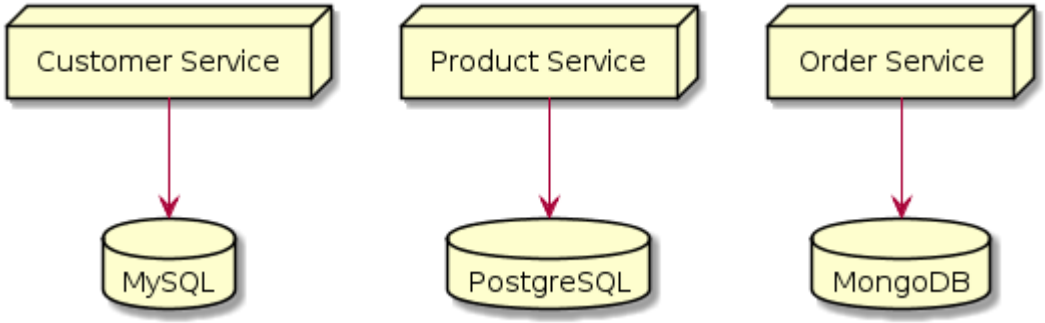
Contoh Stateless Microservices



Persistence Microservices

- Biasanya memiliki database
- Bisa juga disebut sebagai Master Data Microservice
- Biasa digunakan untuk mengolah data di database (CRUD)

Contoh Persistence Microservices

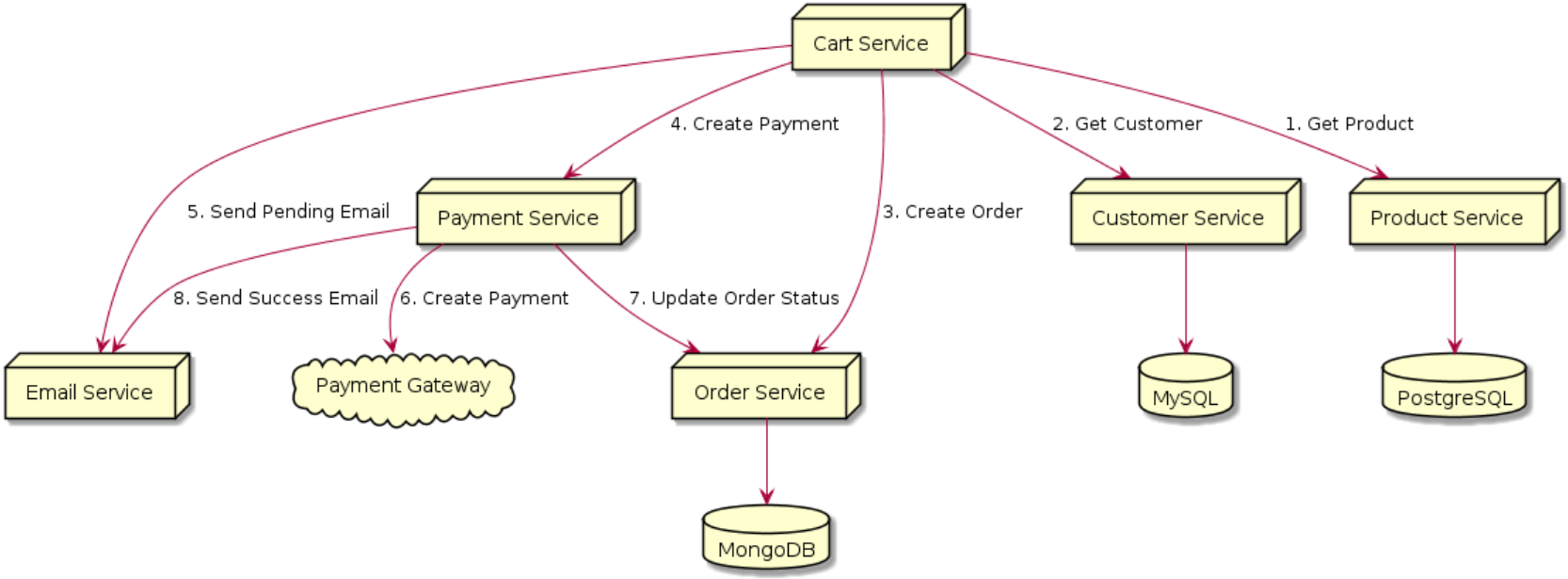


Aggregation Microservices

- Tergantung dengan microservice lain
- Biasa digunakan sebagai pusat business logic aplikasi
- Boleh memiliki database ataupun tidak
- Tidak bisa berdiri sendiri

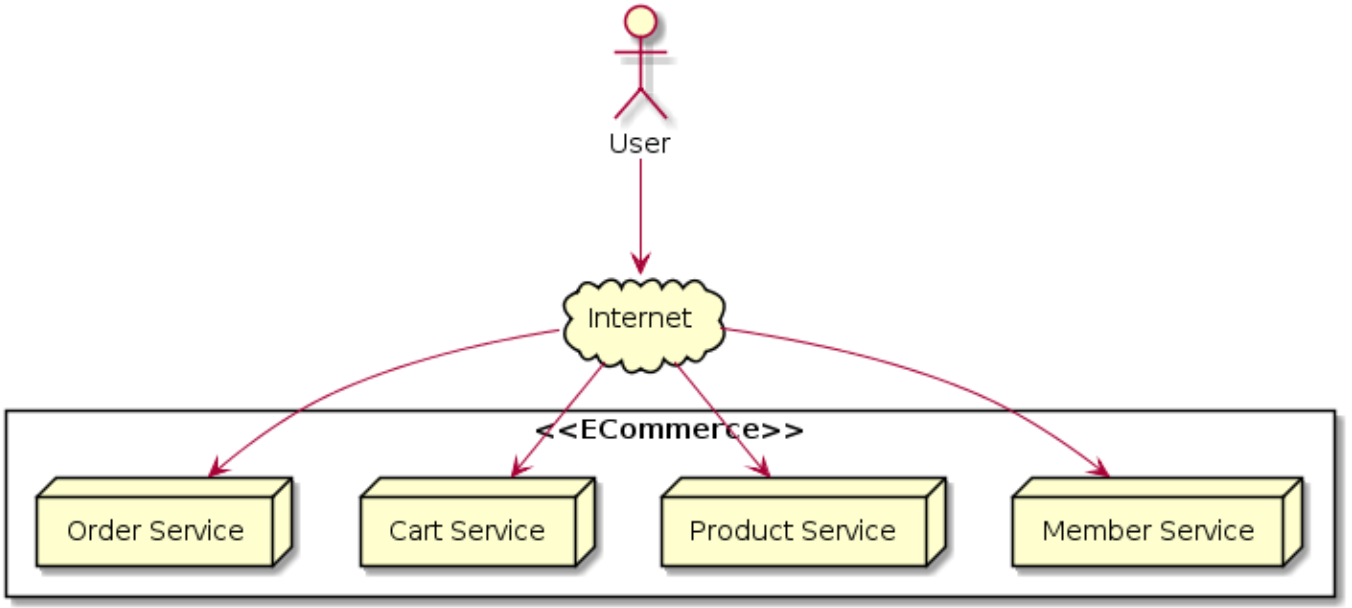
Contoh Aggregation Microservices

Contoh Kasus



API Gateway

Mengekspos Microservices



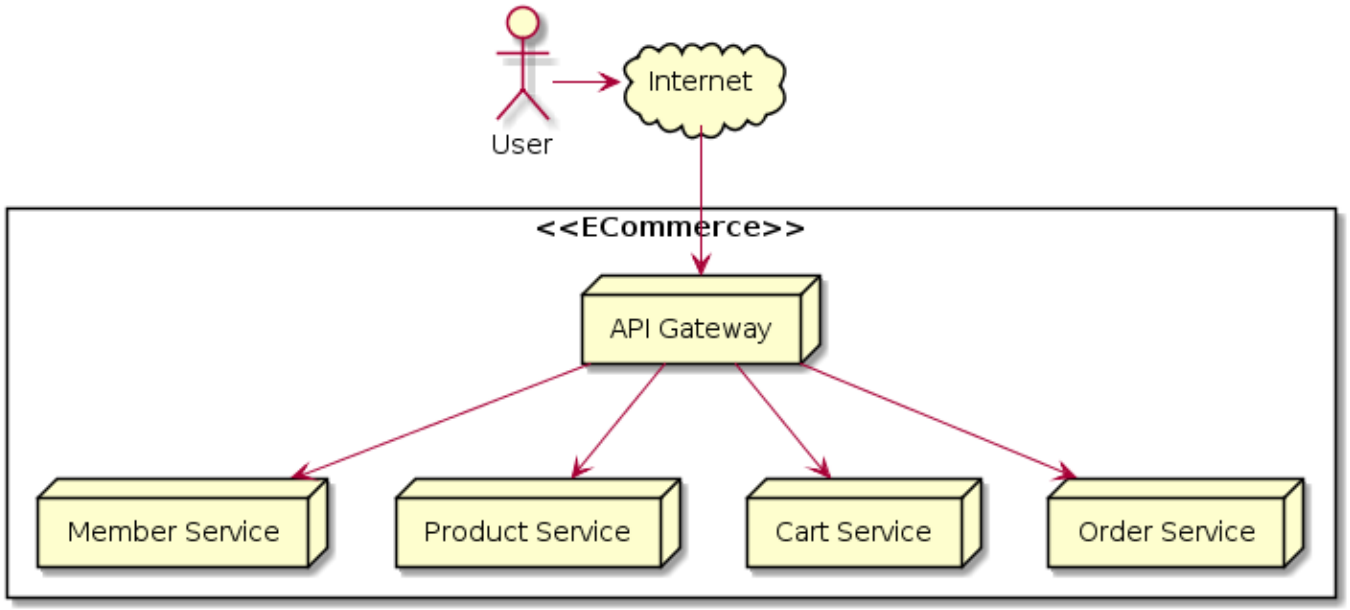
Masalah Mengekspos Microservices

- Semua service bisa diakses dari luar
- Jika butuh Autentikasi, harus diimplementasikan di semua service
- Rawan terjadi kebocoran data

API Gateway

- API Gateway adalah aplikasi yang bertugas sebagai gerbang dari luar ke dalam
- Luar adalah akses dari internet, dan Dalam adalah aplikasi microservices
- API Gateway bertugas sebagai proxy server ke semua aplikasi microservices
- Aplikasi microservices hanya bisa diakses dari luar melalui API Gateway

API Gateway


User
Internet
<<ECommerce>>
API Gateway
Member Service
Product Service
Cart Service
Order Service

Keuntungan API Gateway

- Lebih aman karena satu gerbang
- Service tidak perlu mengimplementasikan proses Autentikasi, cukup dilakukan di API Gateway
- API Gateway juga bisa digunakan sebagai load balancer
- Bisa digunakan sebagai rate limiter
- Bisa digunakan sebagai pengaman sehingga error dari service tidak terekspos

Contoh API Gateway

- Nginx
- Apache HTTPD
- Kong
- Netflix Zuul
- Spring Cloud Gateway