Nama : Andre Ferdinan

Kelas : TK2A

Ringkasan Materi

**Authentication dan Authoriazation**

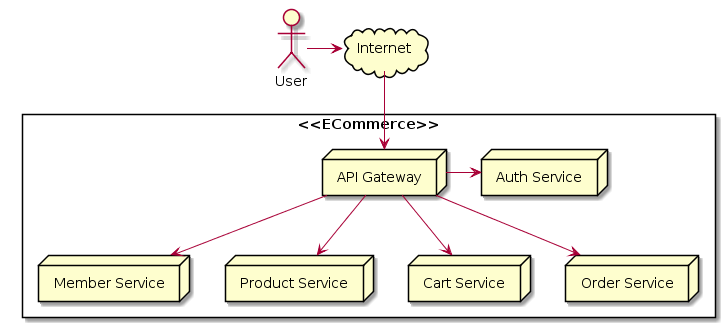
Authentication :

* Memvalidasi kredensial untuk memverifikasi pemilik identitas
* Contoh proses Authentication adalah proses login menggunakan username dan password, dan banyak yang lainnya.

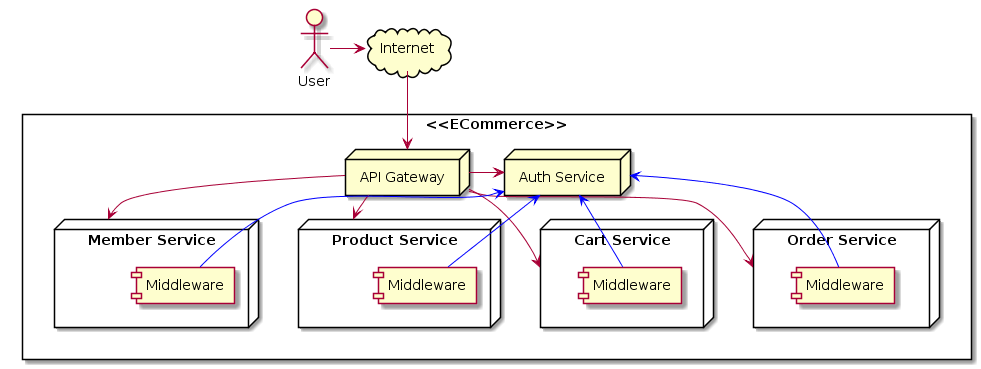
Authorization :

* Authorization adalah proses yang dilakukan setelah proses Authentication
* Memvalidasi apakah pemilik identitas memiliki hak akses untuk mengakses resource yang diminta
* Contoh proses Authorization adalah Access-Control List, dan banyak yang lainnya.

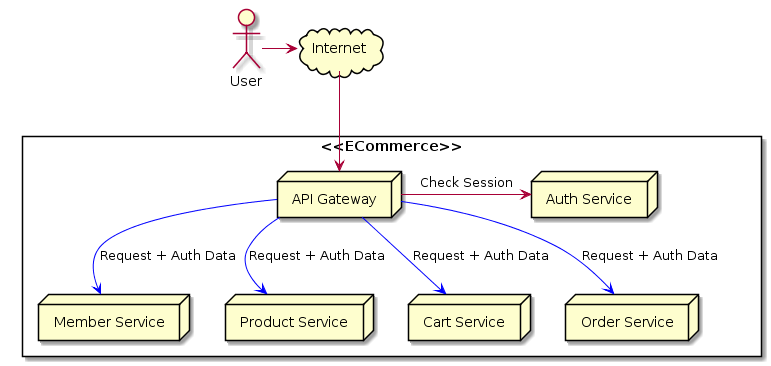
Auth Service :



Integrasi dengan auth service

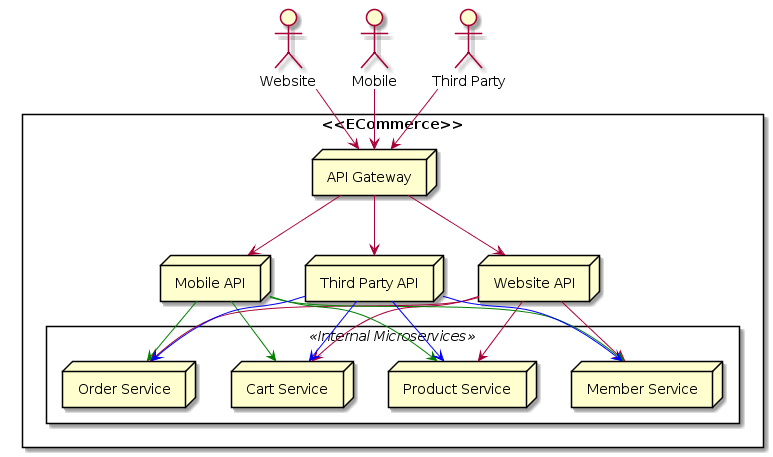


API GATEWAY sebagai middleware



**Backend for Frontend**

* Backend for Frontend adalah menyediakan backend khusus untuk frontend tertentu
* Biasanya satu backend akan melayani satu frontend secara specific
* Makin banyak jenis frontend, makin banyak backend yang dibuat

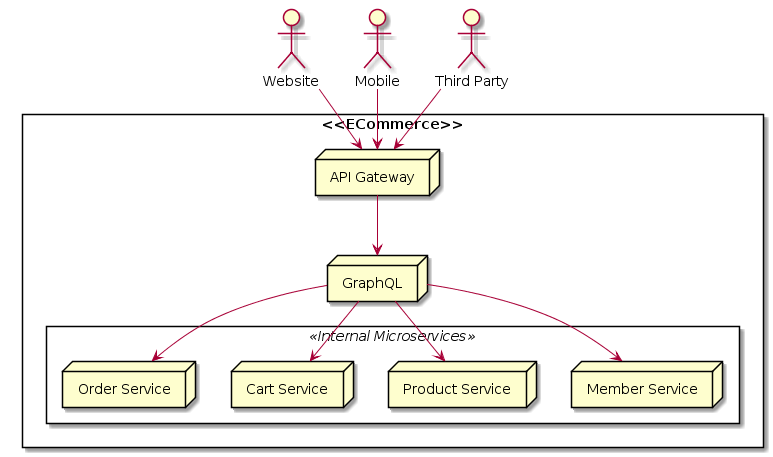


Keuntungan backend for frontend

* Pengembangan backend untuk tiap frontend bisa terisolasi satu sama lain
* Logic untuk frontend tidak tercampur di satu backend

GraphQL : Alternative Backend for Frontend

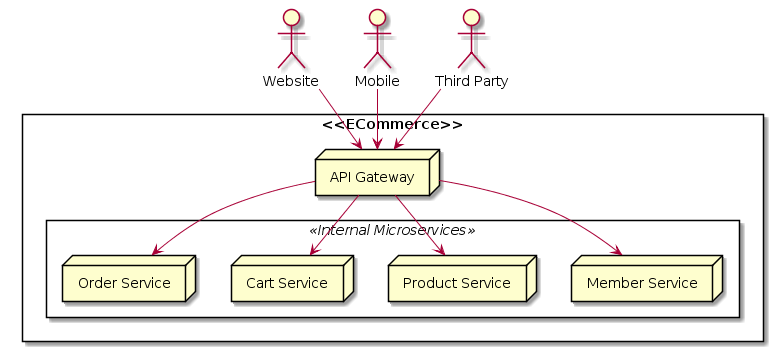
* GraphQL adalah query language untuk API
* GraphQL dapat digunakan untuk memanipulasi response API secara runtime
* Frontend bebas menentukan data apa aja yang ingin didapatkan
* Backend hanya perlu menyediakan data lengkap, dan Frontend bisa dengan bebas menentukan data apa aja yang diinginkan.



Kekurangan Menggunakan GraphQL

* Butuh melakukan development GraphQL Server di Backend
* Butuh melakukan development GraphQL Client di Frontend

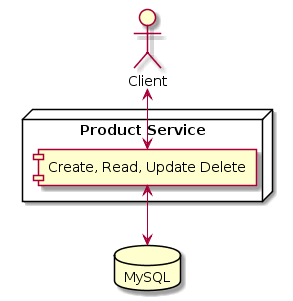
Permasalahan banyak jenis frontend



* Tiap frontend punya mekanisme autentikasi berbeda
* Kecepatan bandwidth tiap frontend berbeda
* API yang dibutuhkan tiap frontend berbeda
* Semua kebutuhan jenis frontend harus diimplementasikan di satu API Gateway

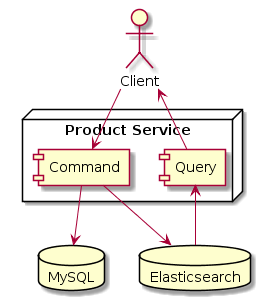
**CQRS (Command Query Responsibility Segregation)**

Persistence Microservice



Command Query Responsibility Segregation

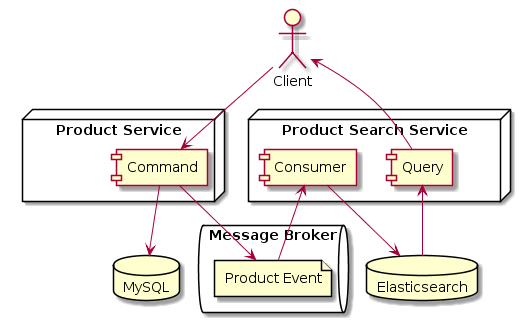
* CQRS adalah proses membedakan operasi Command dan operasi Query
* Operasi Command adalah operasi mengubah data (Create, Update, Delete)
* Operasi Command adalah operasi mengambil data (Get, Search)
* Dalam CQRS, biasanya service atau database dibedakan untuk kebutuhan Command dan kebutuhan Query



Keuntungan CQRS

* Bisa memilih database berbeda yang optimal untuk proses Command dan Query, sehingga operasi Command dan Search bisa lebih cepat, karena database nya sudah disesuaikan dengan kebutuhan
* Membedakan model untuk Command dan Query di aplikasi akan lebih mudah dibanding digabung di satu model yang sama untuk proses Command dan Query
* Performa aplikasi akan lebih baik, karena kita membedakan component untuk Command dan Query

CQRS Menggunakan Messaging

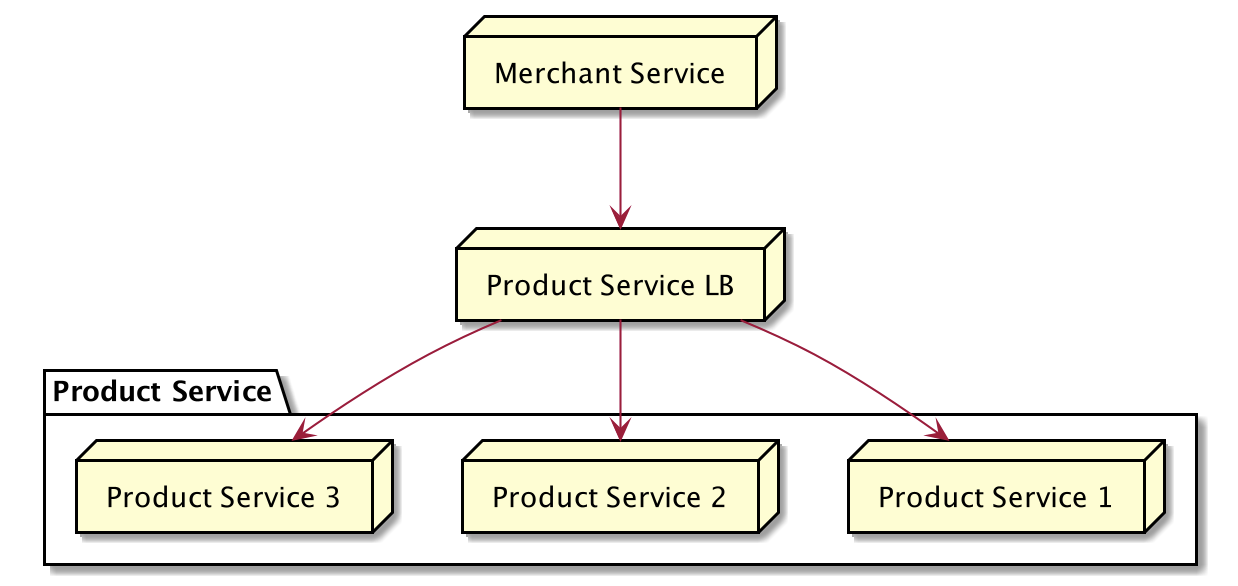


Keuntungan CQRS Menggunakan Messaging

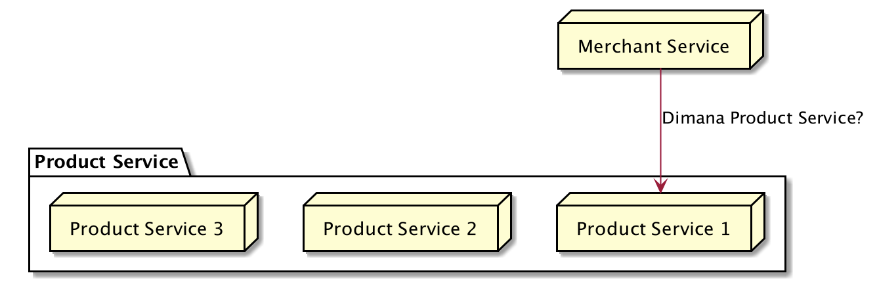
* Aplikasi Command dan Query terpisah, sehingga bisa dikerjakan oleh tim yang berbeda secara paralel
* Aplikasi Command tidak perlu pusing memikirkan struktur data Aplikasi Query, hanya cukup mengirim datanya ke Message Broker
* Scaling aplikasi bisa sesuai dengan kebutuhan, baik itu Command atau Query
* Jika Aplikasi Query sedang stop atau error, data dari Aplikasi Command akan tetap aman tersimpan di Message Broker
* Mekanisme retry akan lebih mudah dilakukan jika melalui Message Broker

**SERVER SIDE DISCOVERY**

* Membuat server khusus sebagai router atau load balancer ke service
* Client hanya butuh terkoneksi ke router atau load balancer
* Jika jumlah node service bertambah atau berkurang, router yang hanya perlu dirubah, client tidak perlu berubah



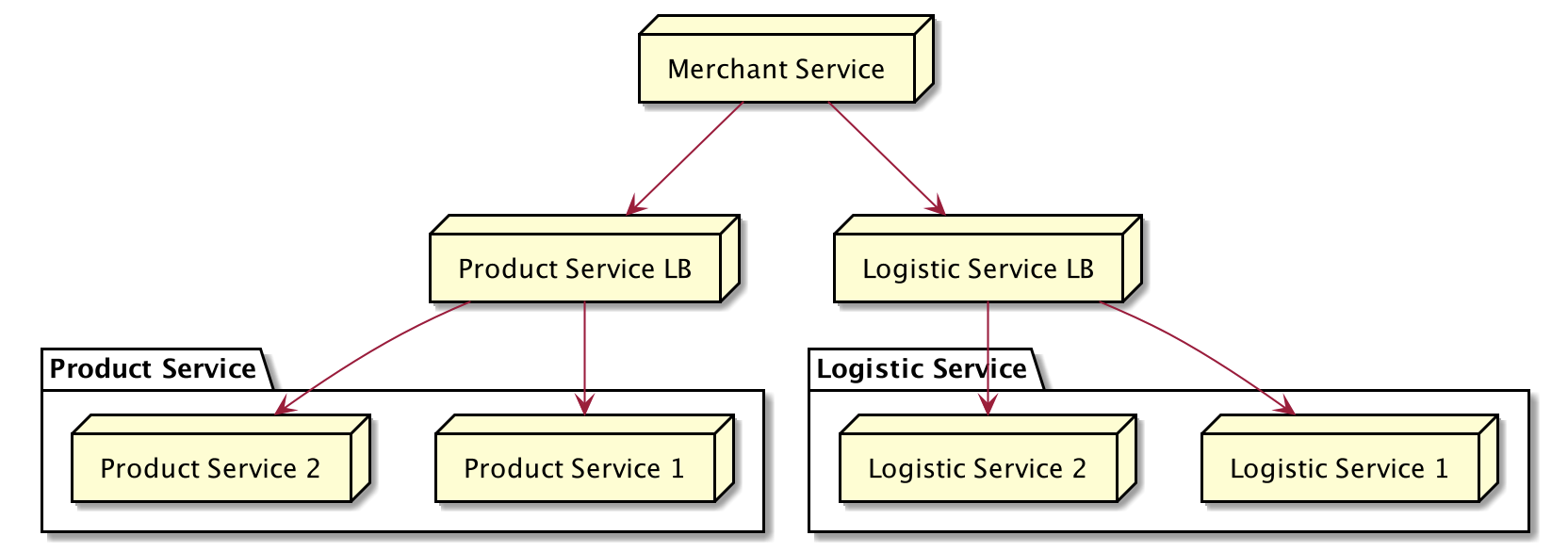
Komunikasi Antar Microservice



Contoh Router atau Load Balancer

* Nginx
* Apache HTTPD
* Traefik

Kekurangan Server Side Discovery



* Tiap service harus memiliki router atau load balancer
* Agar tidak terjadi single point of failure, maka router atau load balancer harus di setup sebanyak 2 instance
* Cost biaya akan lebih mahal, karena 1 service harus menjalankan 2 router