

## 1. Mô hình monolithic

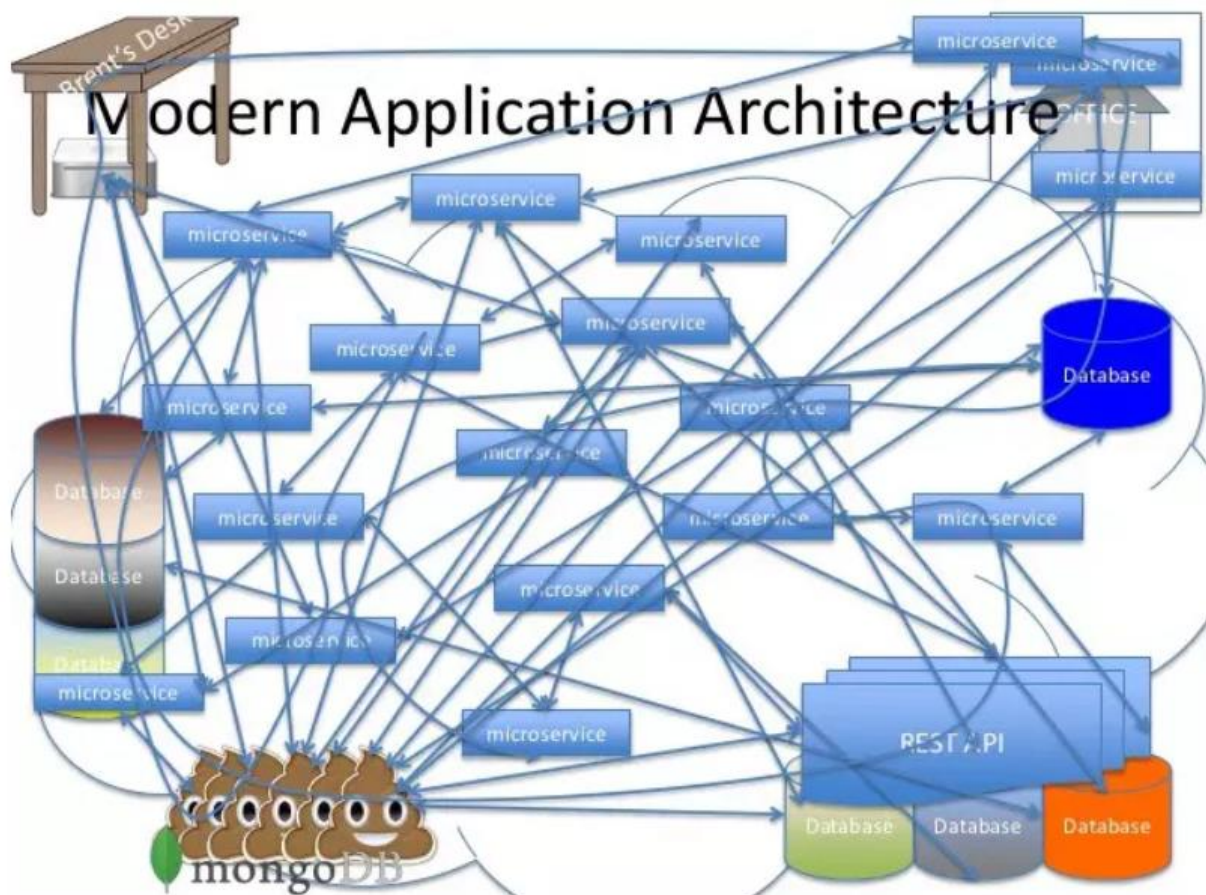
Là mô hình thường được sử dụng cho các ứng dụng nhỏ, ít service. Mô hình xây dựng các service thành 1 khối các service có thể có ràng buộc với nhau.

Ưu điểm:

- Dễ phát triển, tốc độ nhanh, dễ tìm lỗi, sửa lỗi.
- Dễ deploy, không yêu cầu nhiều kiến thức devop

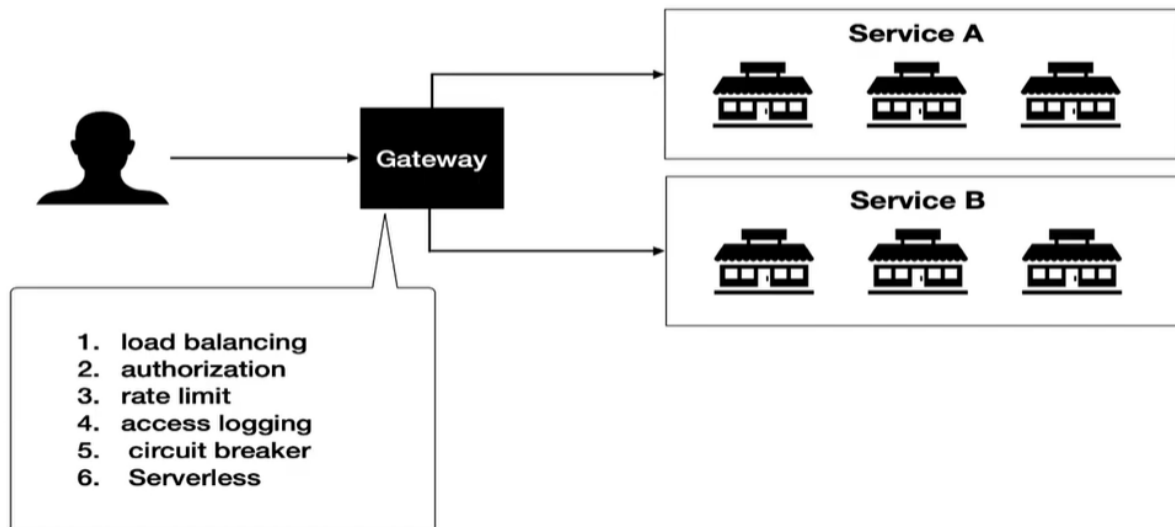
Nhược điểm:

- Khi ứng dụng phát triển nhiều service sẽ khó quản lý và phát triển
- Một service chết thì kéo theo cả hệ thống die
- Thường sử dụng 1 ngôn ngữ lập trình duy nhất => không áp dụng được những tinh túy của các ngôn ngữ khác
- Khi cần thêm một tính năng mới phải lưu ý tới ràng buộc giữa các service với nhau, không thì cũng die luôn.



## 2. Mô hình microservices

Là mô hình chia để trị, cụ thể chia một ứng dụng thành nhiều service nhỏ hơn và độc lập.



Với mô hình này có thể phân tán các service trên các server khác nhau miễn sao chúng có thể liên lạc được.

Ưu điểm:

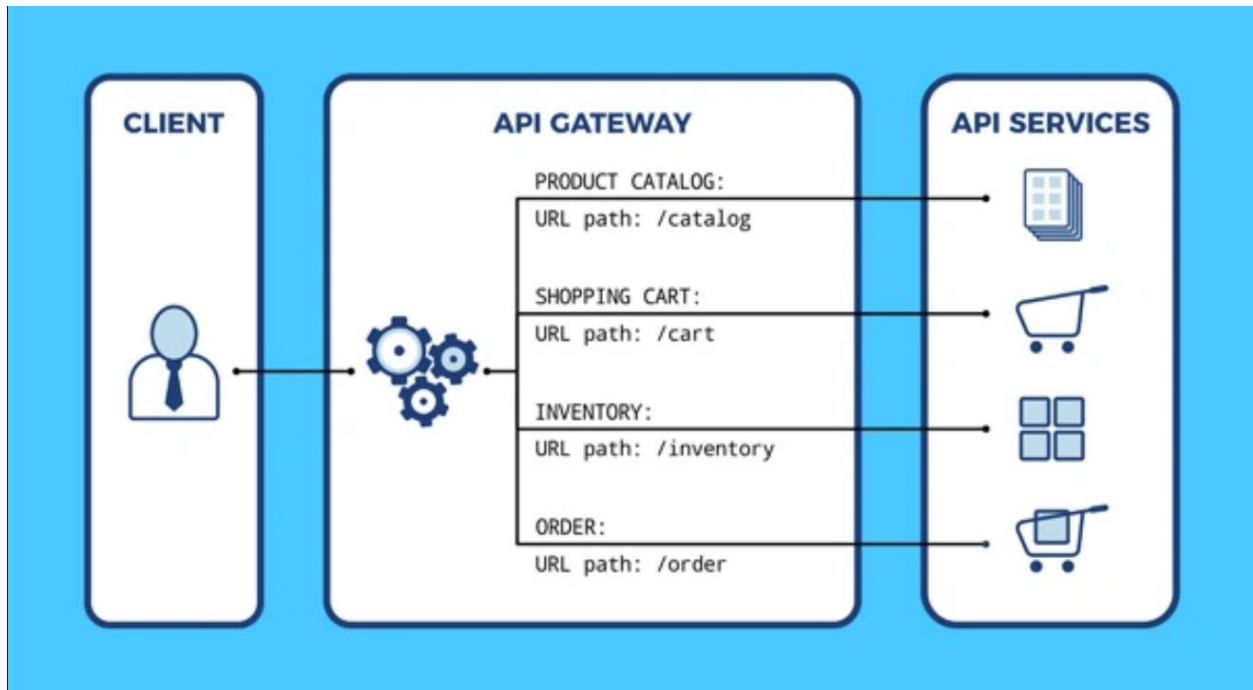
- Các service hoạt động độc lập với nhau, service nào chết thì cũng ko ảnh hưởng tới các service khác
- Dễ nhận biết được hệ thống đang bị chết ở service nào (đối với các lỗi đặc thù)
- Các service nào được ưu ái hơn thì cho vào một server khỏe hơn, tối ưu được tài nguyên phần cứng và phù hợp với từng loại service. Ví dụ: service xử lý ảnh thì cho vào server có Gpu khỏe, service nào xử lý về store thì cho vào server có nhiều disk.
- Thời gian xử lý có thể sẽ nhanh hơn rất nhiều nếu tổ chức hệ thống tốt
- Phù hợp với một đội dev lớn cùng phát triển 1 hệ thống

Nhược điểm:

- Cồng kềnh nếu sử dụng cho những hệ thống nhỏ
- Cần nhiều kiến thức về devops
- Tốc độ phát triển hệ thống chậm hơn (cần có 1 ông devops giỏi)
- Authen, author rườm rà hơn. Mỗi service cần có logic authen author
- Khó fix bug và xử lý lỗi nếu các service call lẫn nhau.

## 3. Api gateway

Là một ông chỉ đường. Ông này có trách nhiệm điều hướng cho các request của client tới đúng microservice mong muốn.



Ưu điểm:

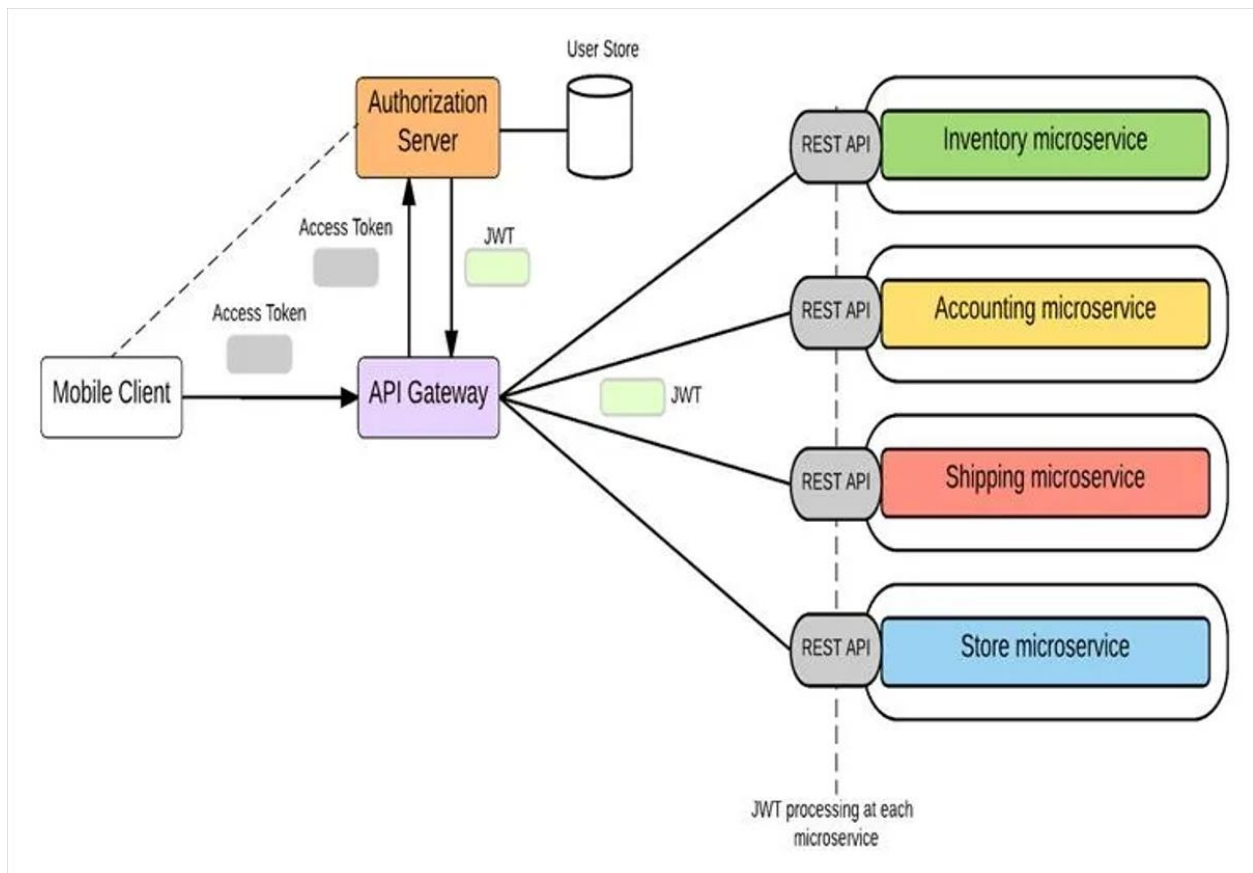
- Che dấu kiến trúc microservice với thế giới bên ngoài
- Phần code frontend sẽ gọn gàng hơn, thay vì frontend call api từ nhiều domain khác nhau thì ở đây chỉ cần 1 domain, rồi api gateway sẽ tìm đến domain, service phù hợp.
- Dễ dàng theo dõi, quản lý traffic, log các kiểu đồ
- Thêm 1 lớp bảo mật nữa cho hệ thống

Nhược điểm:

- Tăng thời gian response
- Thêm tác nhân gây lỗi

#### 4. Authen, Author trong Api gateway tới các microservice

Trong hệ thống microservice có nhiều cách để authen. Dưới đây sẽ giới thiệu về 1 cách authen bằng JWT



Trên đây là cách authen, cần có 1 author, authen server. Mục đích để sinh ra token JWT. Sau khi server sinh ra một token thì qua API Gateway sẽ tạo ra một token (gateway token) nữa từ token đó và trả về cho người dùng gateway token.

Khi request được gửi tới, api gateway tiến hành giải token và truyền vào cho các microservice sử dụng.