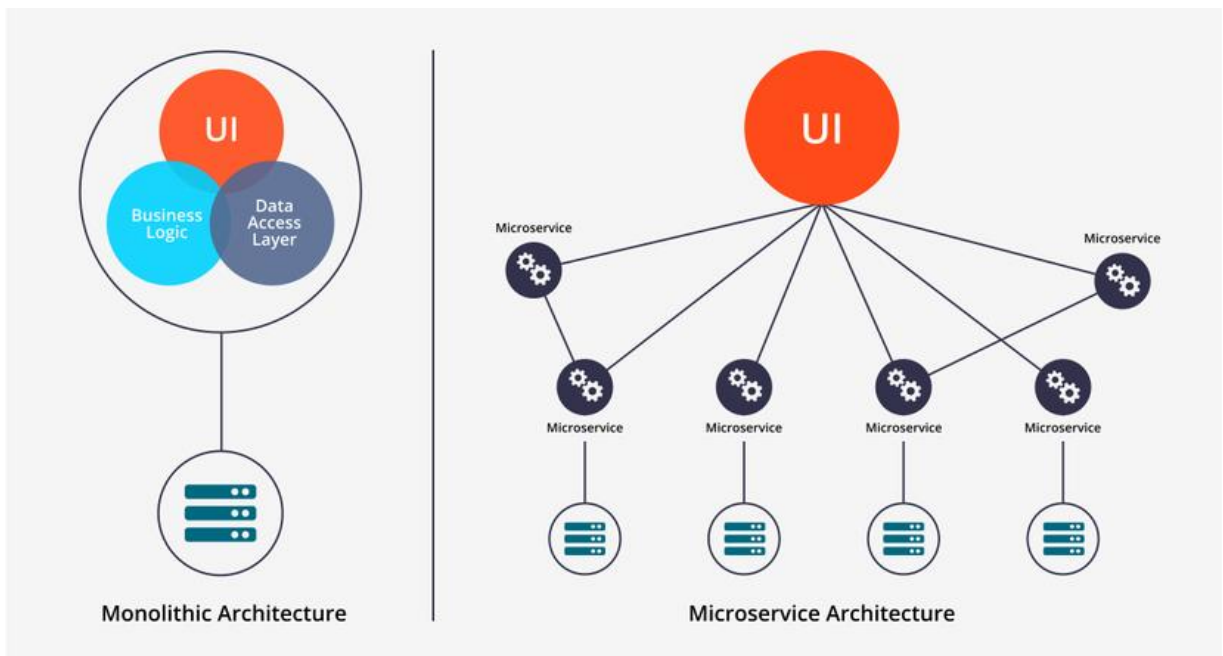


1. Microservice là gì?

Microservices là kiến trúc tập hợp các service nhỏ, độc lập hoạt động cùng nhau.

Mỗi microservice sẽ chịu trách nhiệm riêng, các team có thể phát triển chúng độc lập với các microservice khác.

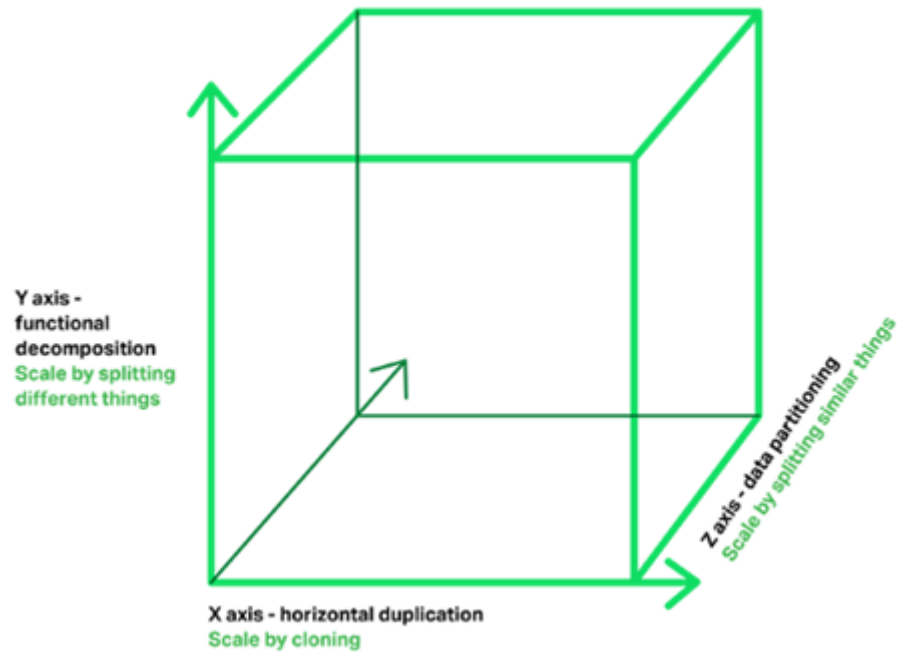
Chúng giao tiếp với nhau qua rất nhiều phương thức có thể là Rest API hoặc là GRPC hoặc Lambda hoặc bất cứ phương thức nào có thể giao tiếp được.



Hình 1.1 Microservices và Monolithic Architecture

2. Scale Cube

Một trong những ưu điểm lớn nhất của microservices là về khả năng mở rộng.

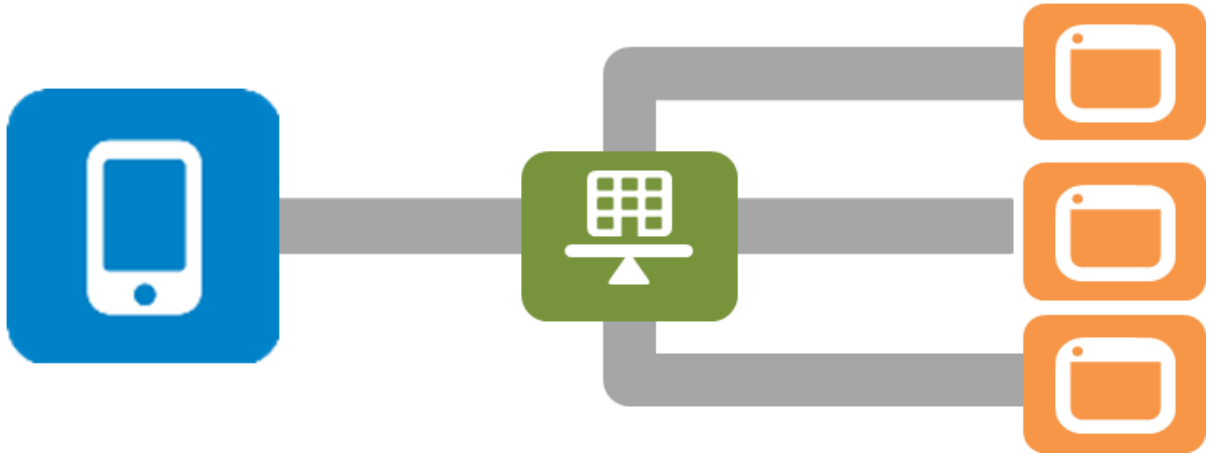


Hình 2.1 The Scale Cube

Kiến trúc microservice tương đương trục Y của 3 chiều mở rộng chịu tải (Scale Cube).

X-AXIS SCALING

Network name: Horizontal scaling, scale out



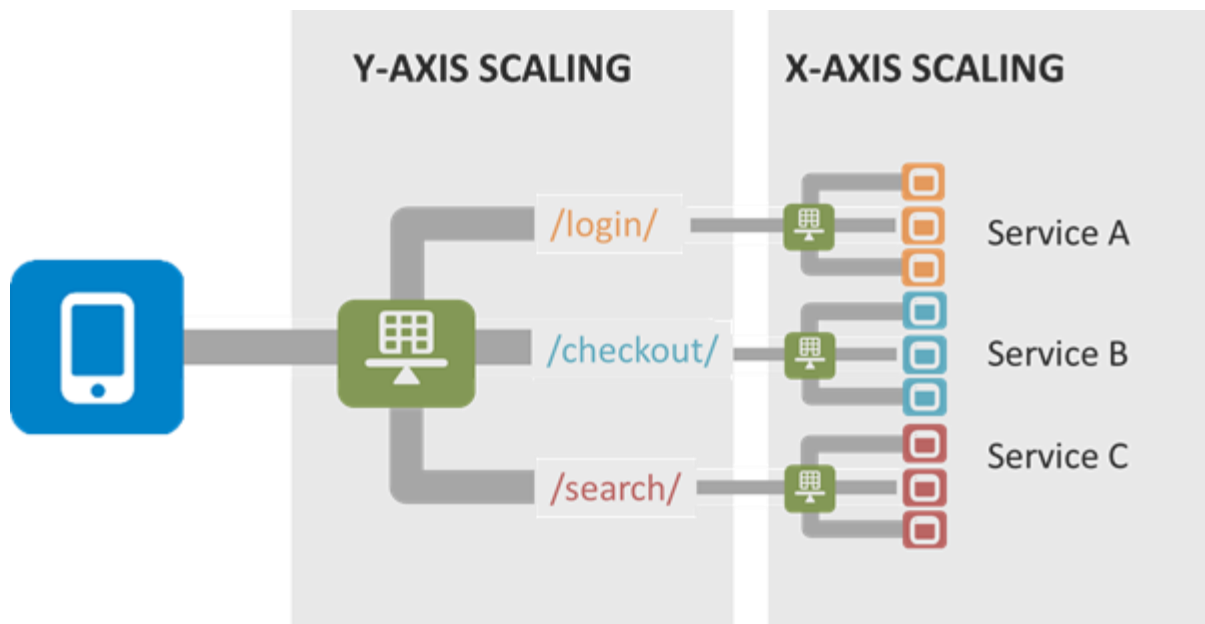
Hình 2.2 Trục X: Mở rộng theo việc clone các service đằng sau bộ cân bằng tải

Y-AXIS SCALING

Network name: Layer 7 Load Balancing, Content switching, HTTP Message Steering



Hình 2.3 Trục Y: Mở rộng theo chức năng



Hình 2.4 Mở rộng theo trục X và Y

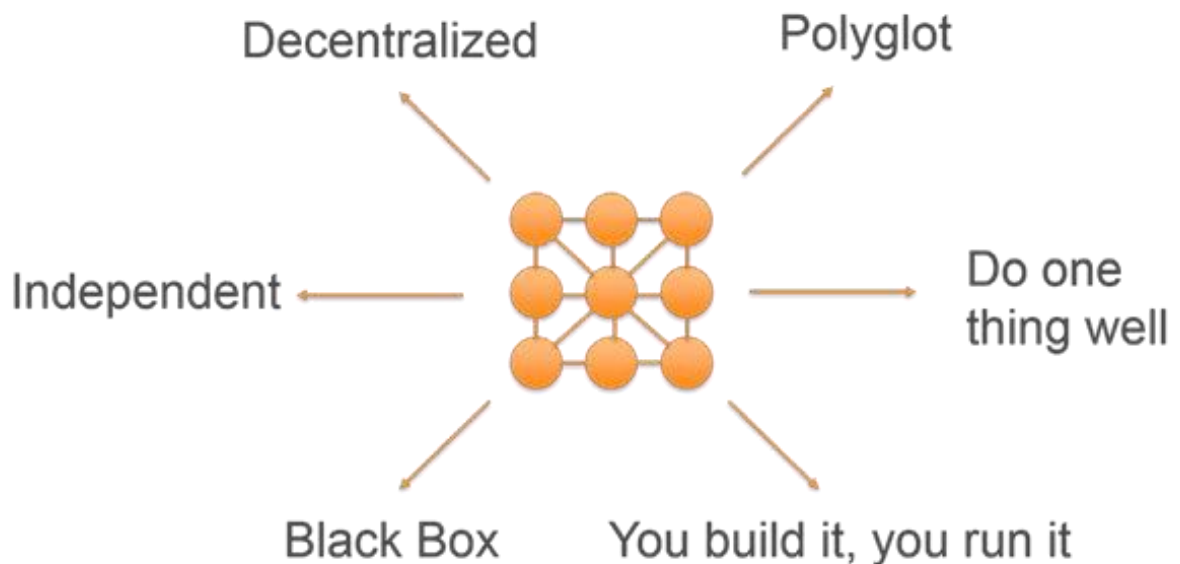
Z-AXIS SCALING

Network name: Layer 7 Load Balancing, Content switching, HTTP Message Steering



Hình 2.5 Trục Z: Mở rộng theo dữ liệu

3. Characteristics



Hình 3.1 Đặc tính của microservices

3.1 Decentralized (phân tán)

Microservices là một hệ thống phân tán về:

- Database – Data model
- Develop
- Deploy
- Manage
- Operate

3.2 Independent

Mỗi service đều có thể thay đổi, nâng cấp hoặc thậm chí thay thế mà không ảnh hưởng tới chức năng các service khác.

3.3 Do one thing well

Mỗi service sẽ được phát triển dựa trên các chức năng của 1 domain nhất định.

Như monolithic có quá nhiều thứ để focus đến khi ứng dụng lớn, phức tạp sẽ bị break => tính hỗn độn, phức tạp, lộn xộn của - 1 trong 3 thách thức phát triển phần mềm (Heterogeneity, Delivery, Security & Trust).

3.4 Polygot

Microservices cho phép các team phát triển tự do chọn tool, ngôn ngữ, data stored.

3.5 Black box

Service bị ẩn đi, mọi giao tiếp đều thông qua API.

3.6 You build it, you run it

Chuỗi các hoạt động phát triển và vận hành ứng dụng do 1 team.



Hình 3.2 DevOps

4. Benefits

- Sử dụng đa ngôn ngữ, mỗi service có thể được phát triển bằng bất kỳ ngôn ngữ lập trình nào.
- Nếu 1 component của hệ thống có vấn đề, component đó sẽ bị cô lập và phần còn lại của hệ thống vẫn hoạt động bình thường.
- Có khả năng mở rộng dễ dàng, chỉ cần mở rộng những service nào cần mở rộng.
- Có thể deploy 1 service độc lập với các phần còn lại của hệ thống. Điều này giúp chúng ta deploy nhanh hơn, cũng đồng nghĩa với việc có thể đưa ra tính năng mới đến khách hàng nhanh hơn.

- Giảm thiểu sự gia tăng phức tạp của một hệ thống lớn.
- Tối ưu cho việc thay thế các service, các service nhỏ nên việc viết lại khi được yêu cầu không là vấn đề vì chi phí thay thế rất thấp.

5. Drawbacks

– Thuật ngữ Microservice tạo ấn tượng cảm giác về kích thước của nó. Trên thực tế nhiều developer ủng hộ chỉ xây dựng service từ dưới 100 dòng code. Service nhỏ là tốt, nhưng nó không phải mục tiêu chính của microservices. Mục tiêu của microservices là phân nhỏ đầy đủ ứng dụng để tạo điều kiện phát triển và triển khai ứng dụng nhanh chóng.

– Ứng dụng microservices là một hệ thống phân tán. Các developer cần phải lựa chọn phát triển mỗi dịch vụ nhỏ giao tiếp với các dịch vụ khác bằng cách nào messaging hay là RPC. Hơn nữa, họ cũng phải viết code để xử lý việc thất bại giữa chừng (partial failure) vì điểm đến của request có thể chậm hoặc không khả dụng. Việc này phức tạp hơn nhiều so với ứng dụng nguyên khối nơi các module gọi nhau thông qua các method/procedure cấp ngôn ngữ.

– Phải đảm bảo giao dịch phân tán (distributed transaction) cập nhật dữ liệu đúng đắn (all or none) vào nhiều dịch vụ nhỏ khác nhau khó hơn rất nhiều.

– Test một ứng dụng microservices cũng phức tạp hơn nhiều. Testing một service trong kiến trúc microservices đôi khi yêu cầu phải chạy cả các dịch vụ nhỏ khác mà nó phụ thuộc. Do đó khi phân rã ứng dụng một khối thành microservices cần luôn kiểm tra mức độ ràng buộc giữa các dịch vụ.

– Việc triển khai microservices phức tạp hơn rất nhiều.