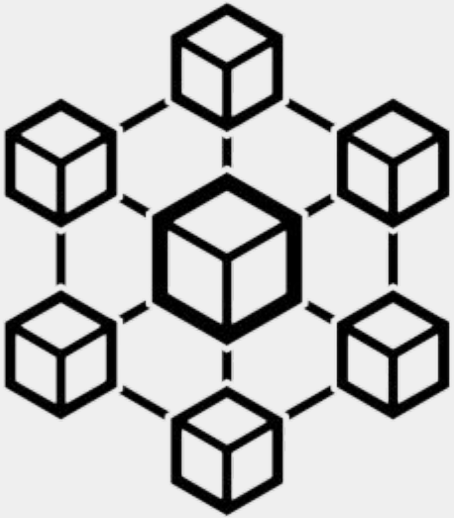
Introduction to Microservices



created by

Rafed Muhammad Yasir

edited by

Toukir Ahammed

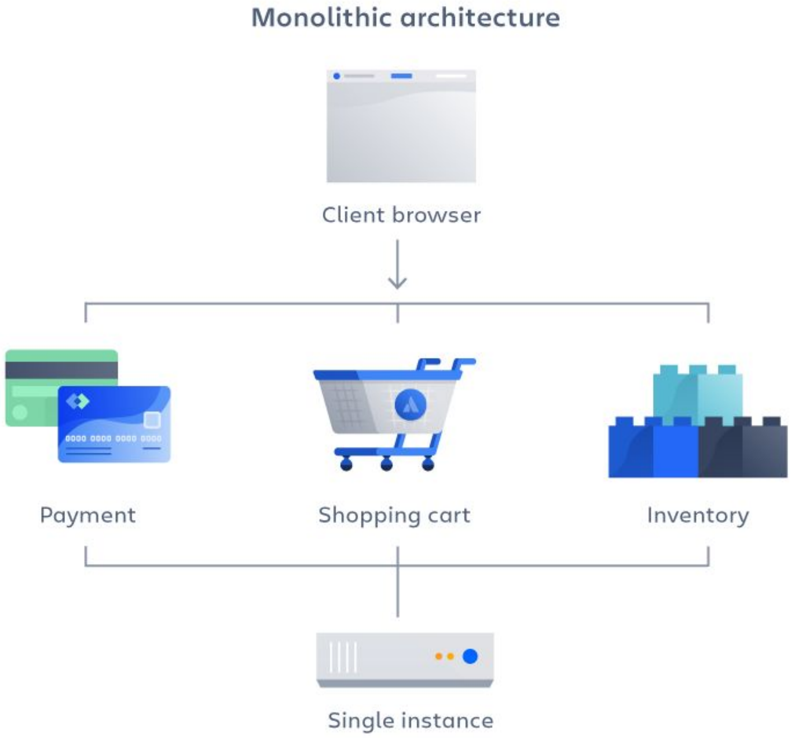
Things We Will Discuss

1. Monolithic Architecture
2. Examples of Monolithic Architectures
3. Challenges of a monolithic architecture
4. Microservices
5. Monoliths vs Microservices (example)
6. Advantages of Microservices
7. Companies Using Microservices
8. Best practices to design a monolithic architecture
9. Problems with Microservices
10. What to know next?

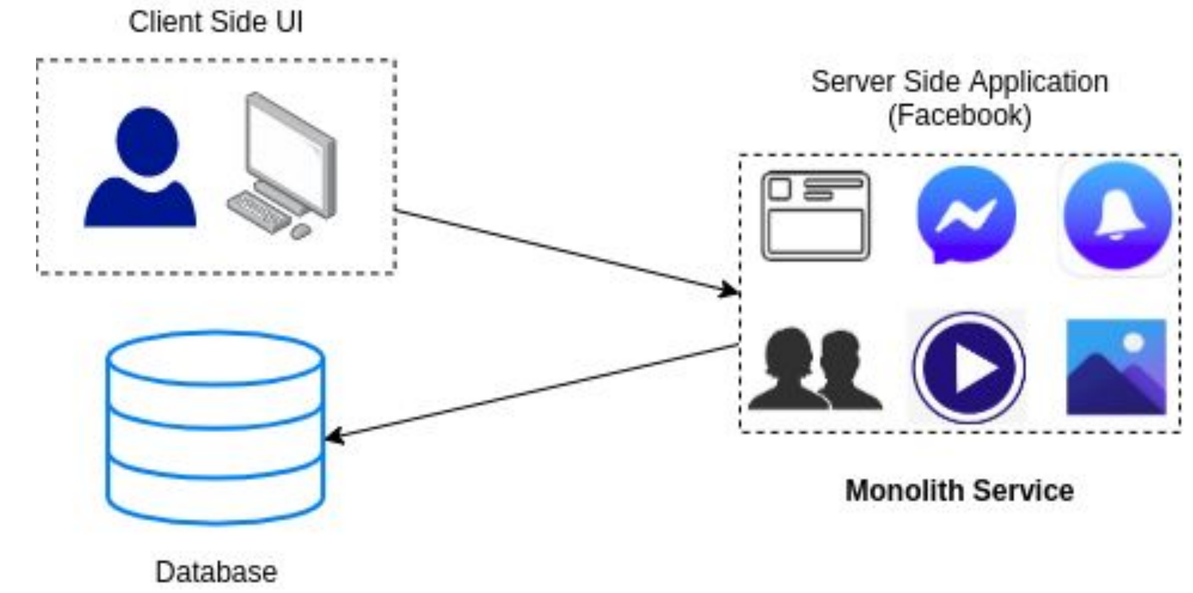
Monolithic Architecture

- A monolithic architecture is a traditional model of a software program, which is built as a unified unit that is self-contained and independent from other applications.
- A monolithic architecture is a singular, large computing network with one code base that couples all of the business concerns together.
- To make a change to this sort of application requires updating the entire stack by accessing the code base and building and deploying an updated version of the service-side interface.

Monolithic Architecture



Monolithic Architecture



Monolithic architecture is like a big container where all the software components of an application are packaged together.

Examples of Monolithic Architectures

WordPress

Router OS

MVP

Minimum Viable Products

Advantages of a monolithic architecture

- Easy deployment – One executable file or directory makes deployment easier.
- Development – When an application is built with one code base, it is easier to develop.
- Performance – In a centralized code base and repository, one API can often perform the same function that numerous APIs perform with microservices.
- Simplified testing – Since a monolithic application is a single, centralized unit, end-to-end testing can be performed faster than with a distributed application.
- Easy debugging – With all code located in one place, it’s easier to follow a request and find an issue.

Monolithic Architecture - Challenges

Large & Complex Applications

Slow Development

Blocks Continuous Deployment

Unscalable

Unreliable

Inflexible

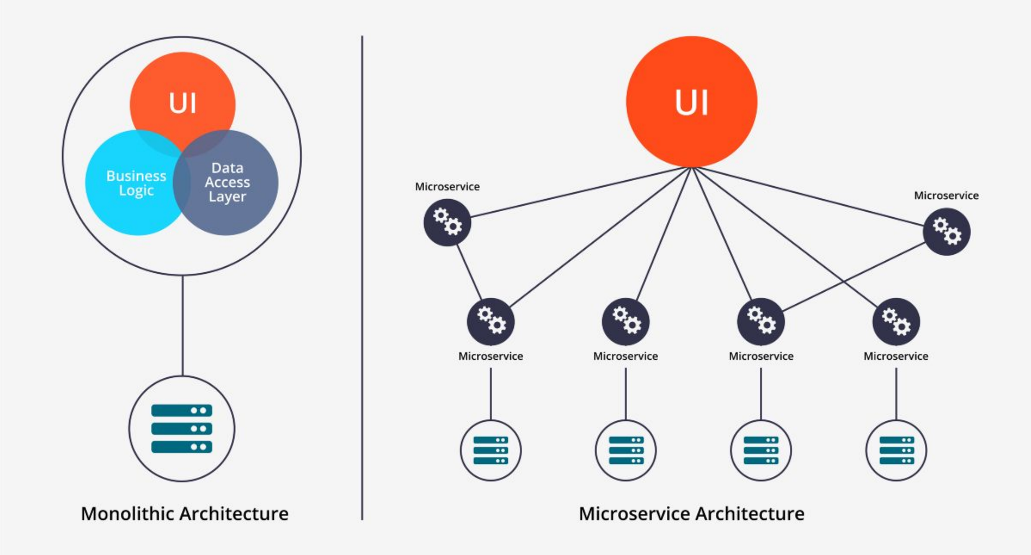
Disadvantages of a monolithic architecture

- Slower development speed – A large, monolithic application makes development more complex and slower.
- Scalability – You can’t scale individual components.
- Reliability – If there’s an error in any module, it could affect the entire application’s availability.
- Barrier to technology adoption – Any changes in the framework or language affects the entire application, making changes often expensive and time-consuming.
- Lack of flexibility – A monolith is constrained by the technologies already used in the monolith.
- Deployment – A small change to a monolithic application requires the redeployment of the entire monolith.

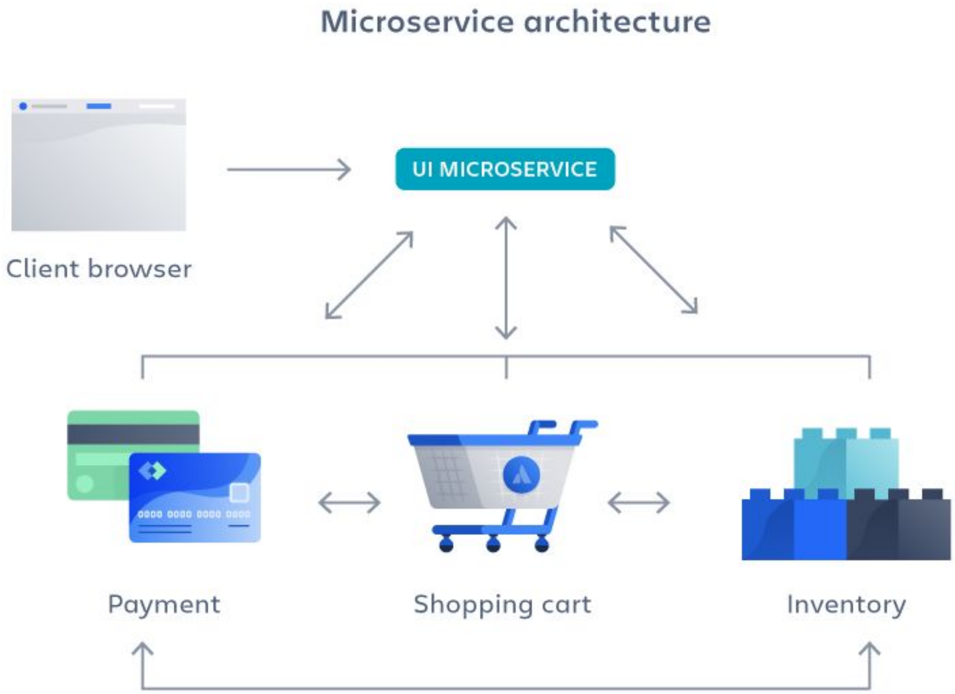
What are Microservices?

Microservices

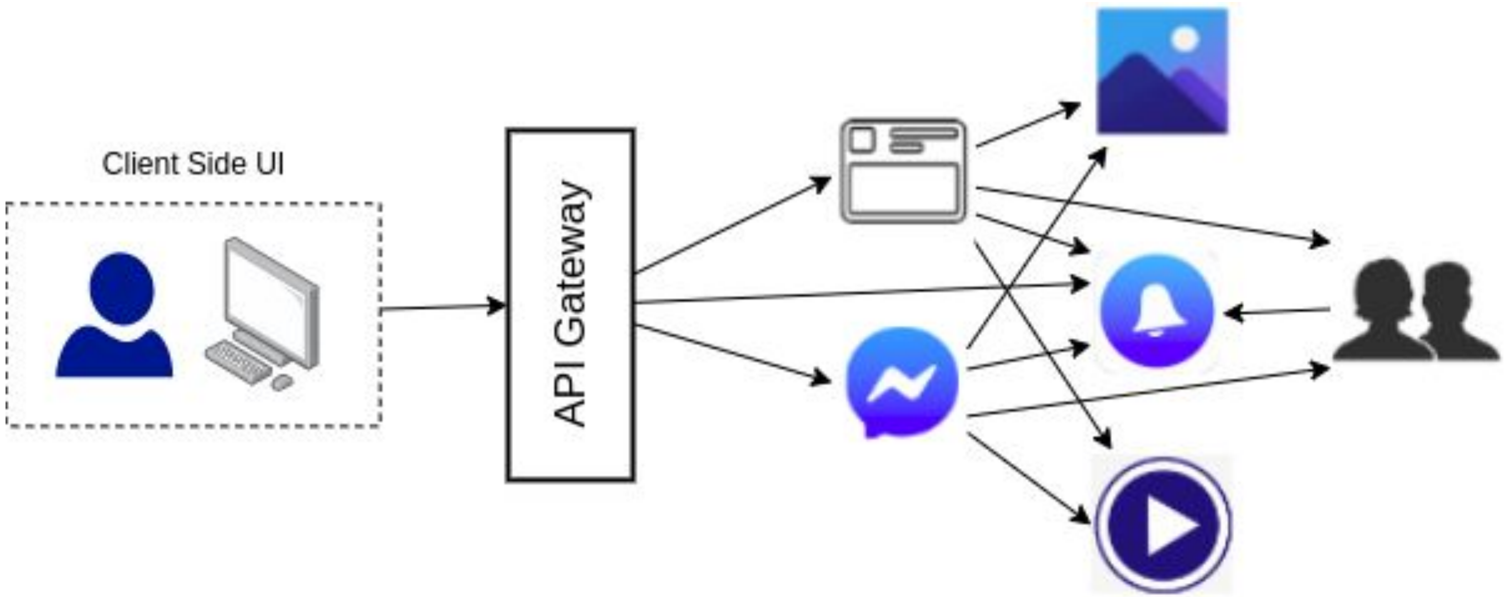
Microservices are basically an architecture where a **monolithic application is decomposed** into **small applications** which are packaged and **deployed independently**



Microservice Architecture

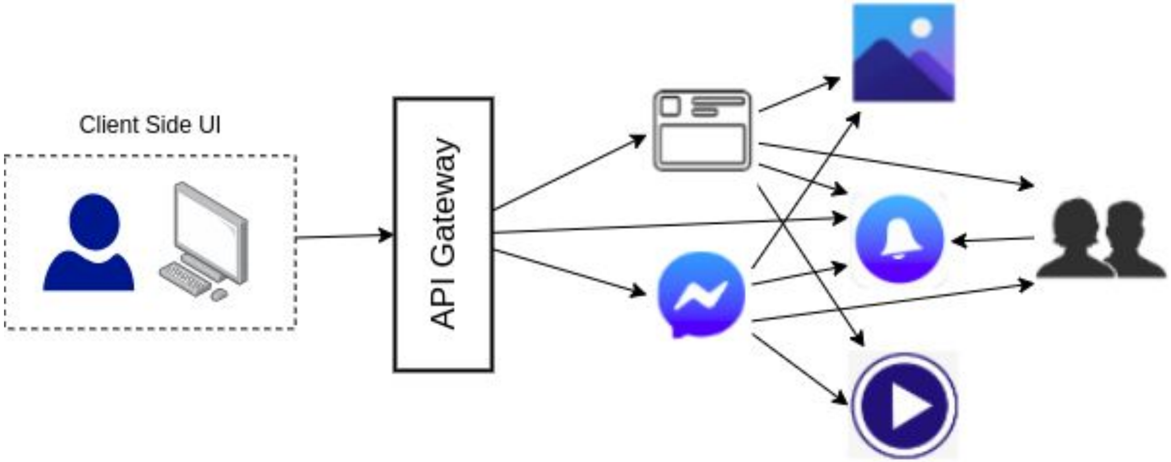


Microservices of Facebook

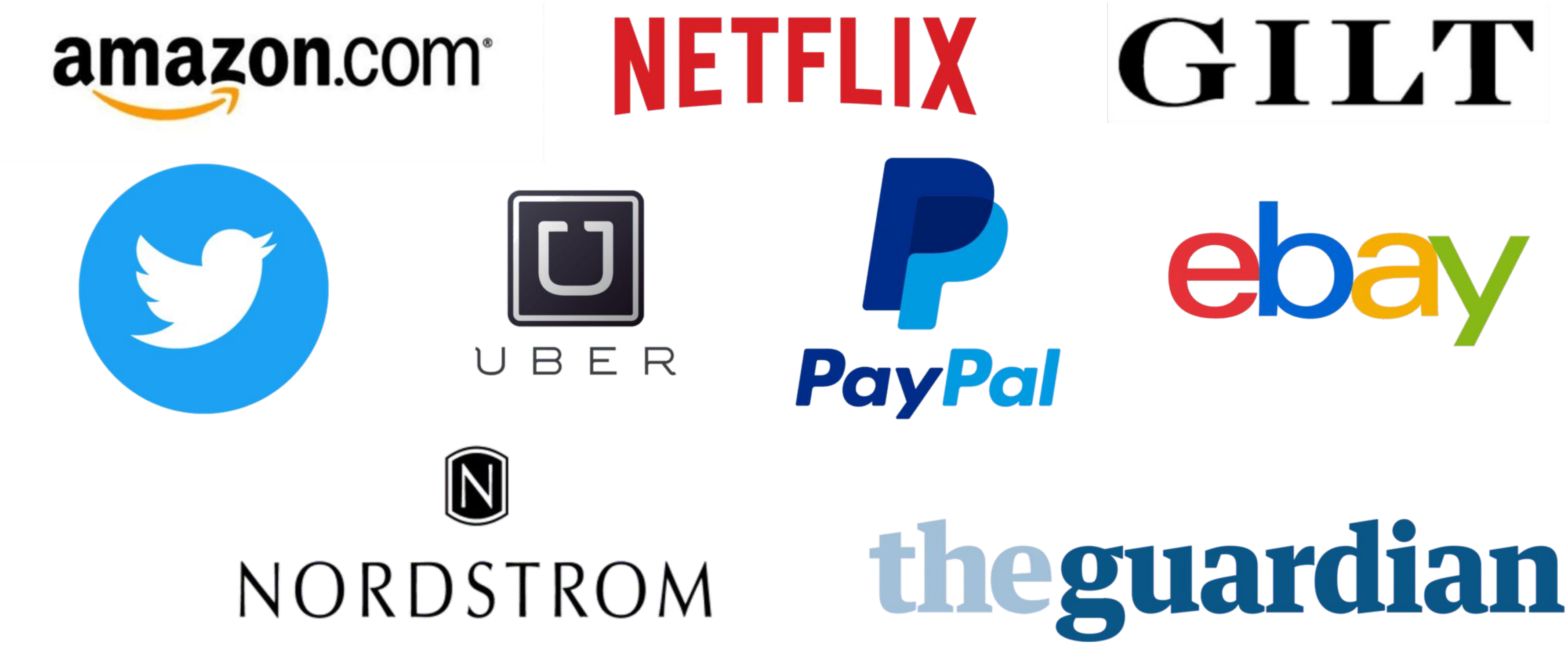


Advantages of Microservice

1. Independent development
2. Independent & continuous deployment
3. Fault isolation
4. Mixed technology stack
5. Granular scaling
6. Agility

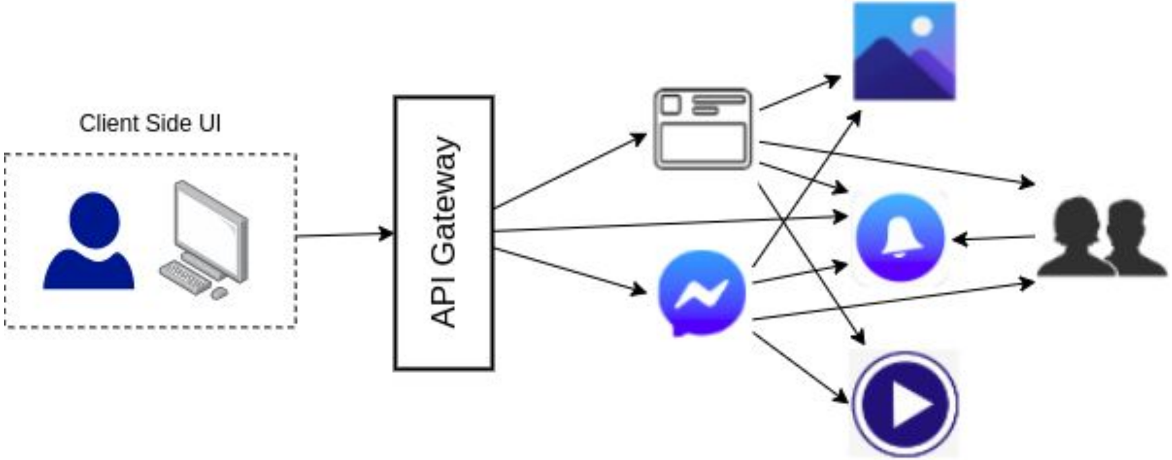


Companies Using Microservices



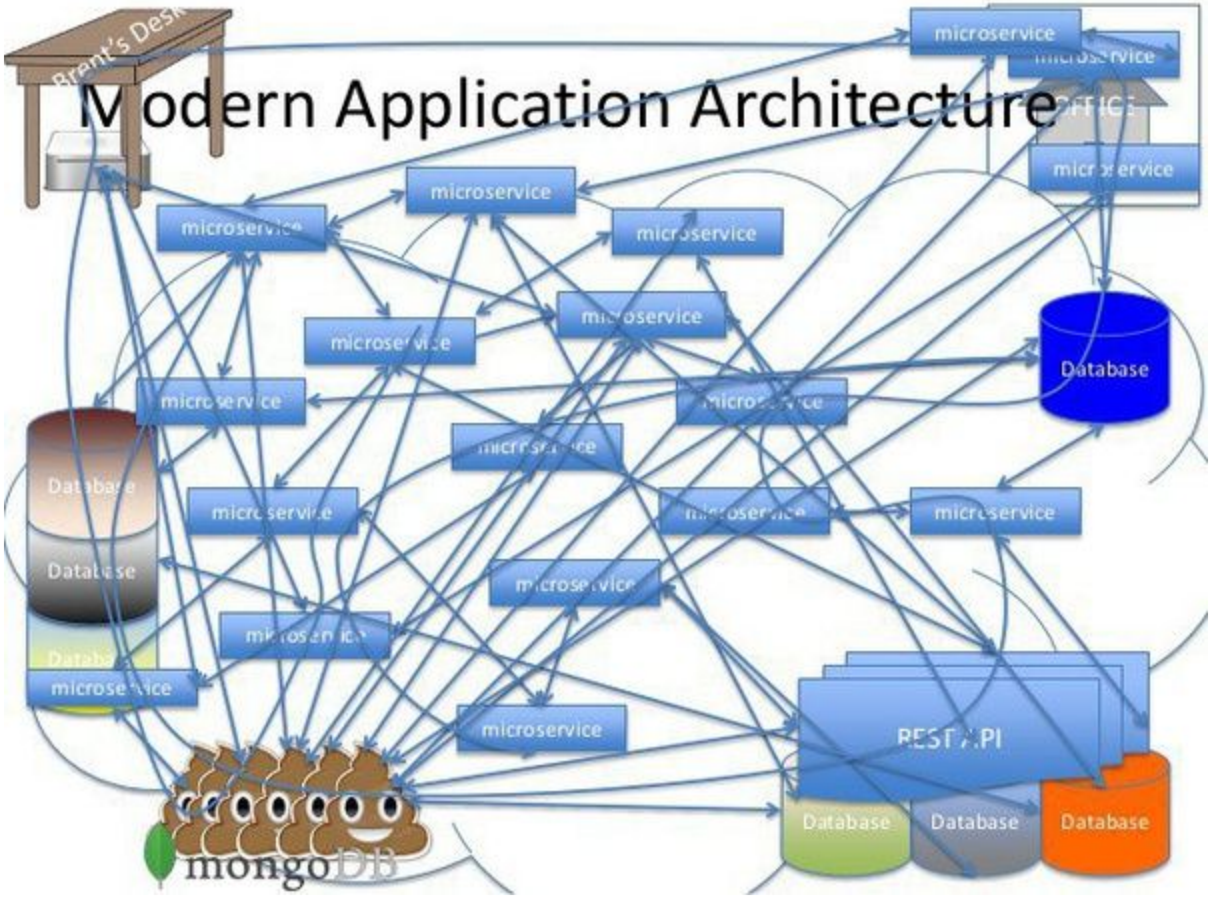
Microservice Architecture - Best Practices

1. Separate data store for each microservice
2. Keep code at a similar level of maturity
3. Separate build for each microservice
4. Deploy in containers
5. Treat servers as stateless



Problems with Microservices

Modern Application Architecture



Q/A