# Microservices

Authored by Chris Bohnet

# Agenda

- What are microservices?
- What is an API gateway?
- Advantages of microservices
- Disadvantages of microservices
- How microservices are deployed and managed in production
- How microservices are scaled

#### What are microservices?

- \* A suite of small applications that each performs a single function.
- \* Able to be developed, tested and deployed independent of other services & by multiple development groups.

- ★ Combination of services structured to be highly maintainable and scalable.
- \* Services communicate with each other through simple, universally accessible application programming interfaces, APIs.
- Designed for rapid delivery of complex applications broken down in smaller services.
- ★ Individual services that are built around business capabilities.

# What is an API Gateway?

An API Gateway is a server that is a single entry point into the system.

It's a management tool that sits between a client and a collection of backend services.

It's responsible for request routing, composition and protocol translations.

It handles
any of the
common tasks
that are used
across a system
of API
services.

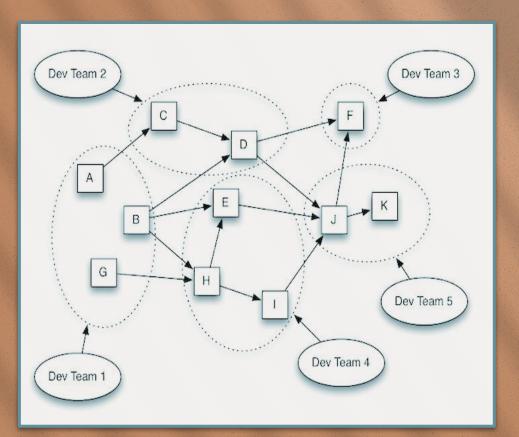
All requests from clients first go through the API gateway and then are routed to the appropriate microservice.

## Advantages

Efficient, productive development with each microservice having a clear team owner.

Collisions avoided with each team having a clear, non-overlapping responsibility.

Applications can grow as the company & business requirements grow.



Continuous delivery possible by independently deploying services also aiding to rapid delivery.

Minimal code of usually no more than 2000 lines performing a single function making it easy to develop, test, deploy, scale and maintain.

Services are loosely coupled driving ease of development, scalability & maintenance.

#### Disadvantages

- ★ Challenging to maintain multiple languages and frameworks & existing tools are likely incompatible with new services.
- \* Each service has its own database & transaction management system making data consistency challenging.
- ★ Each service requires individualized monitoring and testing so additional cost of automation tools and skills needed.
- The number of processing can grow quite large when messaging services and load balancing are factored in.

- Each service has its own database & transaction management system making data consistency challenging.
- Security challenges due to the volume of services and data exchanged among them being exposed on the network.

How microservices are deployed & managed in production

Download & run the docker image on all appropriate server(s). Choose a source code repository to store your microservices such as github.

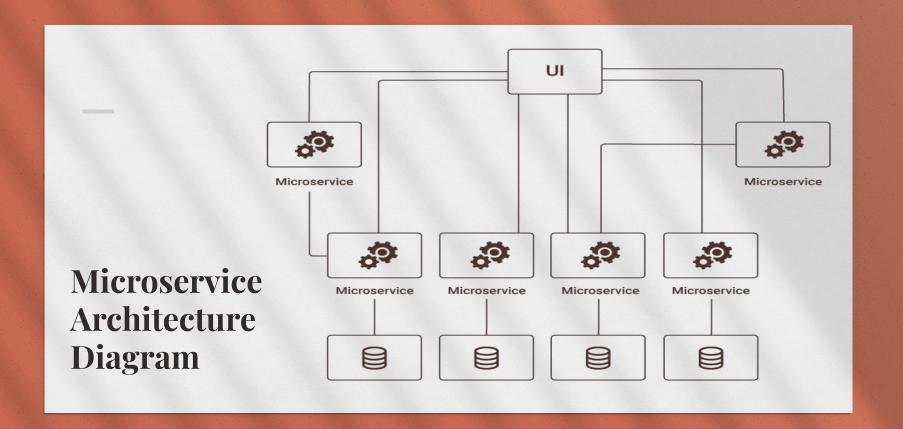
Create continuous integration with building a runtime environment.

Deploy
dependencies
& code with a
lightweight deployment
such as docker images
to build apps independently
from the host
environment.

Build code & Run tests. Communicate configurations to the runtime environment, such as database or instances / ports configurations

### How are microservices scaled?

- Scalability is determined by how efficiently tasks are divided and broken down.
  - Each microservice needs to both scale individually and as part of a larger system.
  - ♦ Dependencies of the microservice should scale with it.
- Performance measures how efficiently the system is able to perform these tasks.
  - Concurrency is the process by which each individual task is broken down into smaller pieces or services.
  - Partitioning determines the efficiency by which these services run in parallel to each other.



## Resources

Newrelic, "Microservices: What they are and how to use them", as referenced at <a href="https://blog.newrelic.com/technology/microservices-what-they-are-why-to-use-them/">https://blog.newrelic.com/technology/microservices-what-they-are-why-to-use-them/</a>

https://microservices.io/

Tiempo development, "Top Microservices Disadvantages", as referenced at <a href="https://www.tiempodev.com/blog/disadvantages-of-a-microservices-architecture/">https://www.tiempodev.com/blog/disadvantages-of-a-microservices-architecture/</a>

Programmable Web, "Understanding the Role of APIs in microservices architecture", as referenced at <a href="https://www.programmableweb.com/news/understanding-role-apis-microservice-architectures/analysis/2">https://www.programmableweb.com/news/understanding-role-apis-microservice-architectures/analysis/2</a> 016/05/05

Datawire, "Nine questions to ask when continuously deploying microservices", as reference at <a href="https://www.datawire.io/guide/deployment/nine-questions-ask-continuously-deploying-microservices/">https://www.datawire.io/guide/deployment/nine-questions-ask-continuously-deploying-microservices/</a>
DZone, "Scaling Microservices: The Challenges and Solutions", as referenced at <a href="https://dzone.com/articles/scaling-microservices-the-challenges-and-solutions">https://dzone.com/articles/scaling-microservices-the-challenges-and-solutions</a>