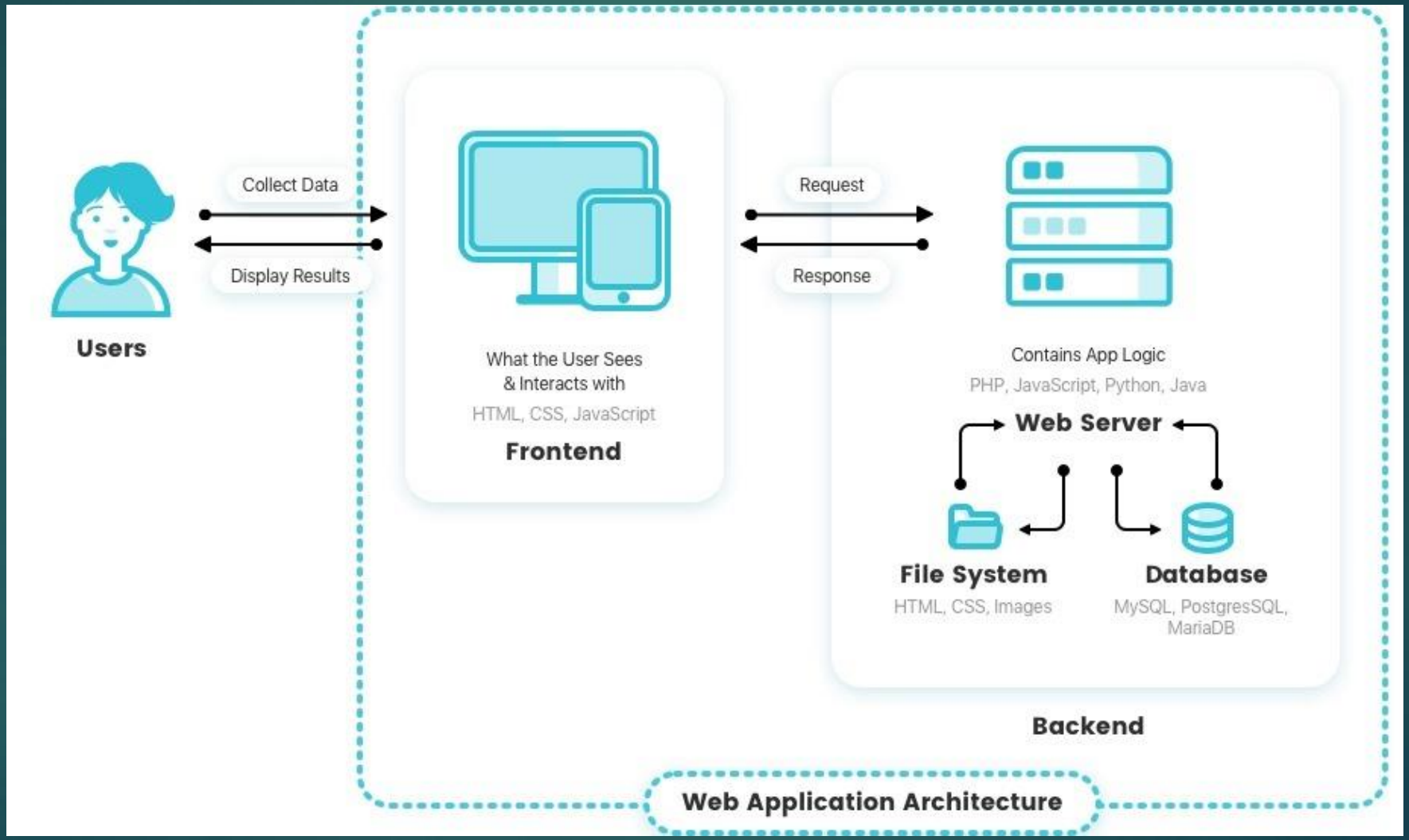
Python Workshop Day 3

MICRO-SERVICE ARCHITECTURE, REST AND CREATING API'S IN PYTHON

How are Apps Built?

How are Apps Built?



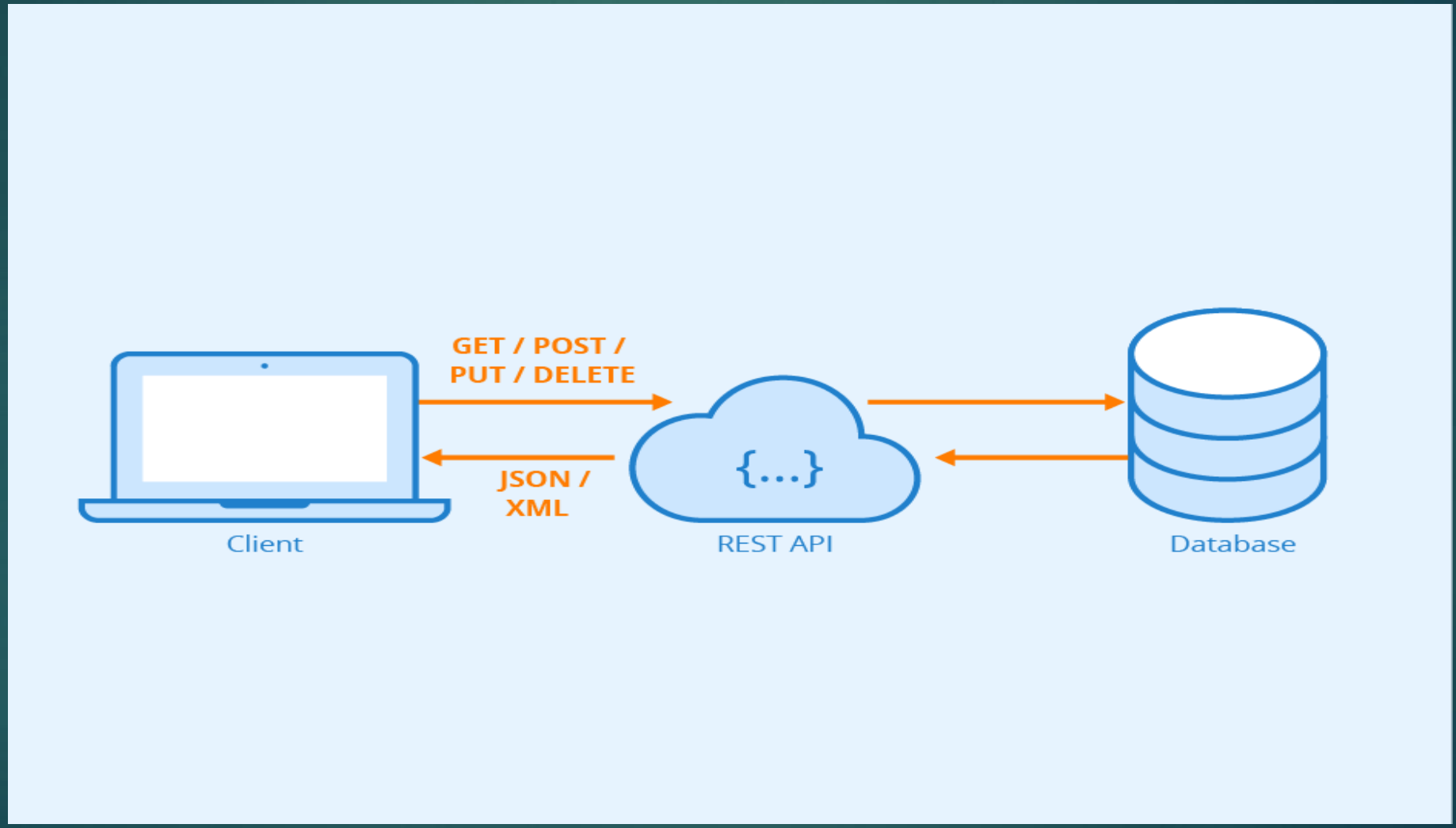
Components of an application?

Components of an application?

To be very precise:

- Frontend
- Backend
- Database

Where comes an API?



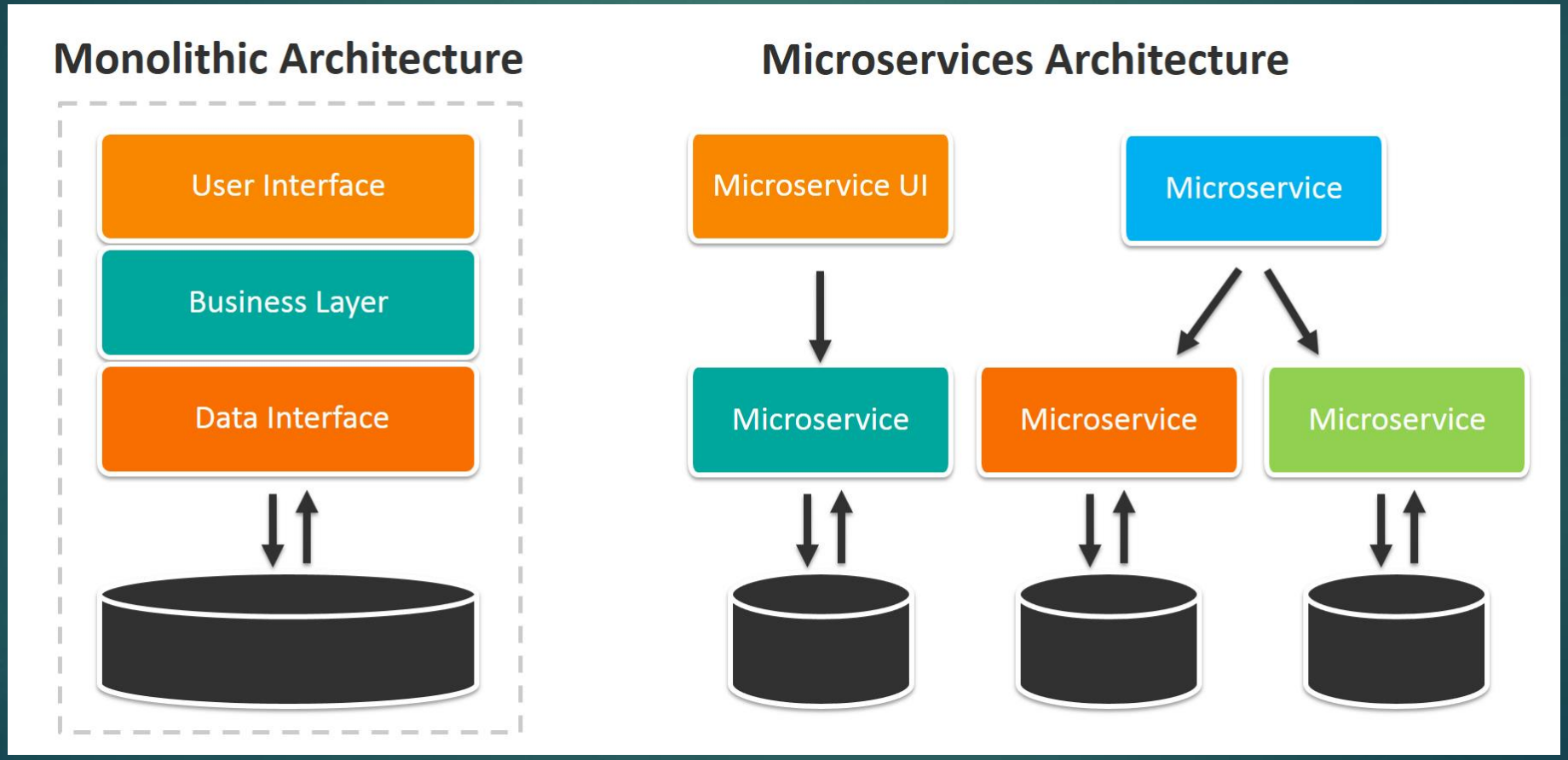
Publically Available API's

- Facebook's Developer Group – Graph API
- Twitter Firehose
- Google's APIGee

Types of Application Architecture?

- Monolithic Architecture
- Micro-services Architecture

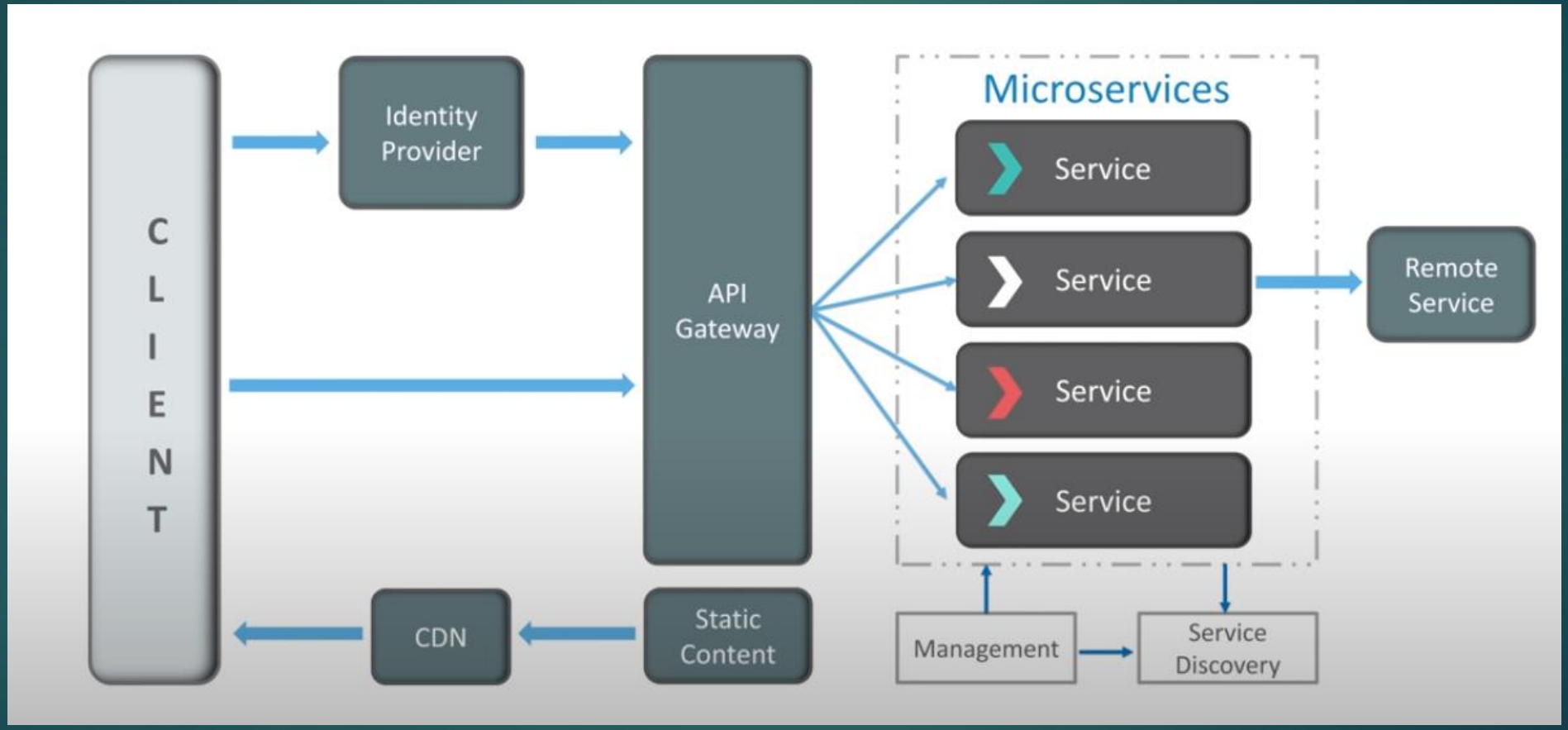
Difference:


Monolithic Architecture
User Interface
Business Layer
Data Interface
Microservices Architecture
Microservice UI
Microservice
Microservice
Microservice
Microservice

Monolithic Architecture Challenge?

- Large and Complex Applications
- Slow Development
- No Continuous Development
- Unscalable
- Unreliable
- Inflexible

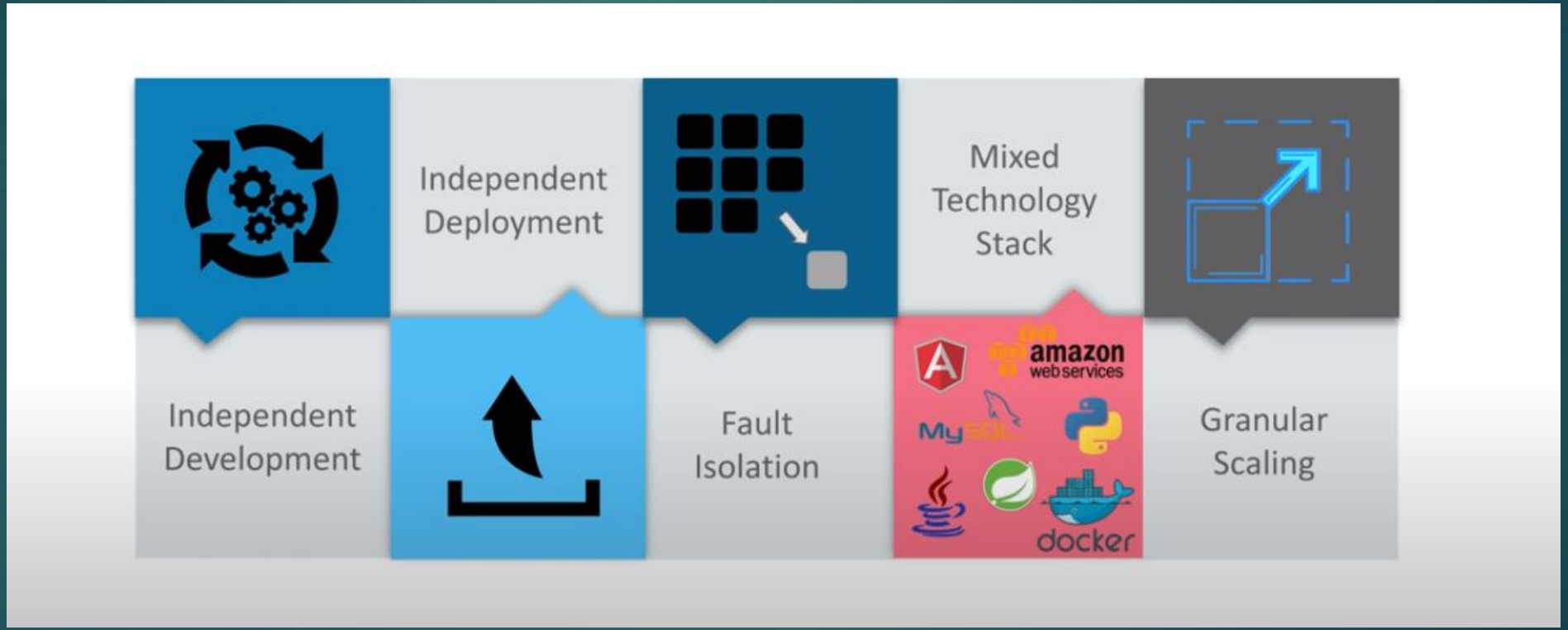
Microservices Application Architecture:



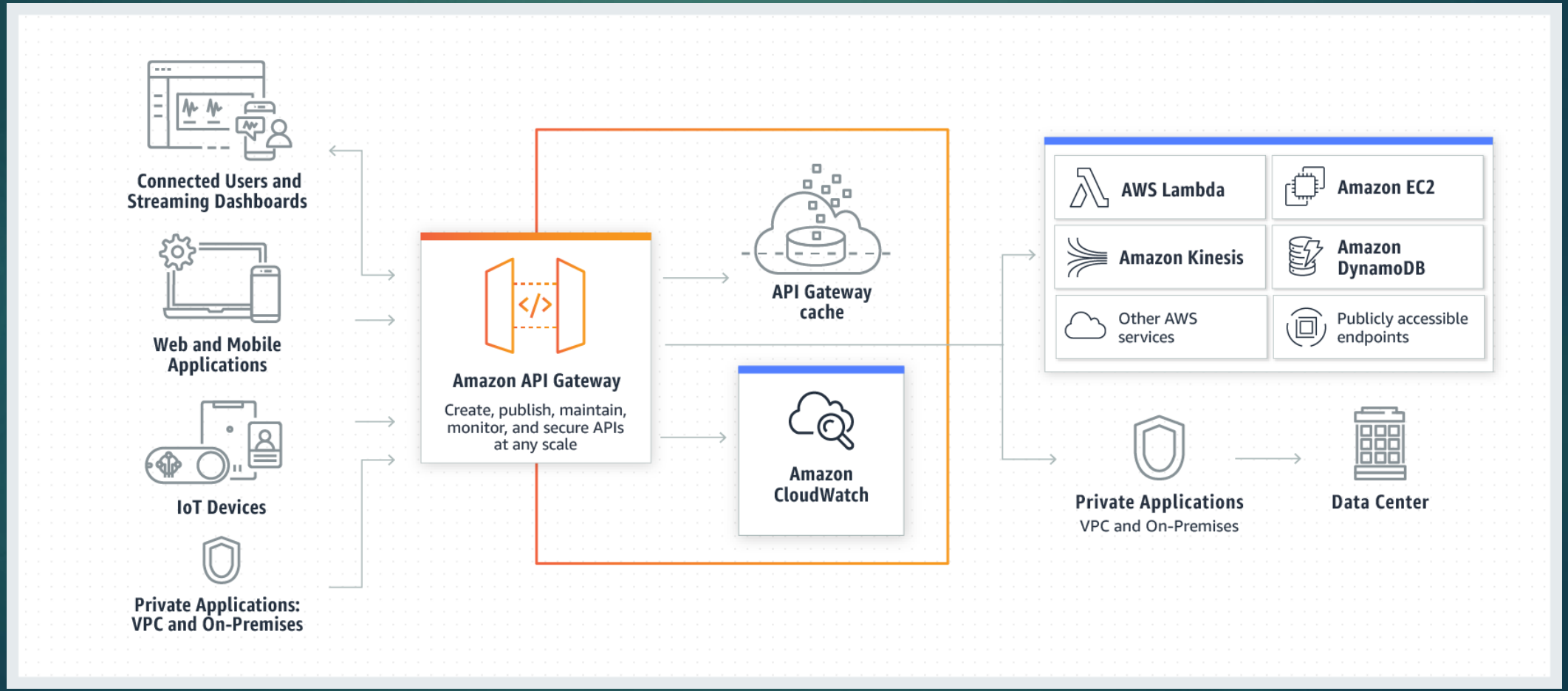
Features of Microservice Architecture

- Small Focused
- Loosely Coupled
- Language Neutral

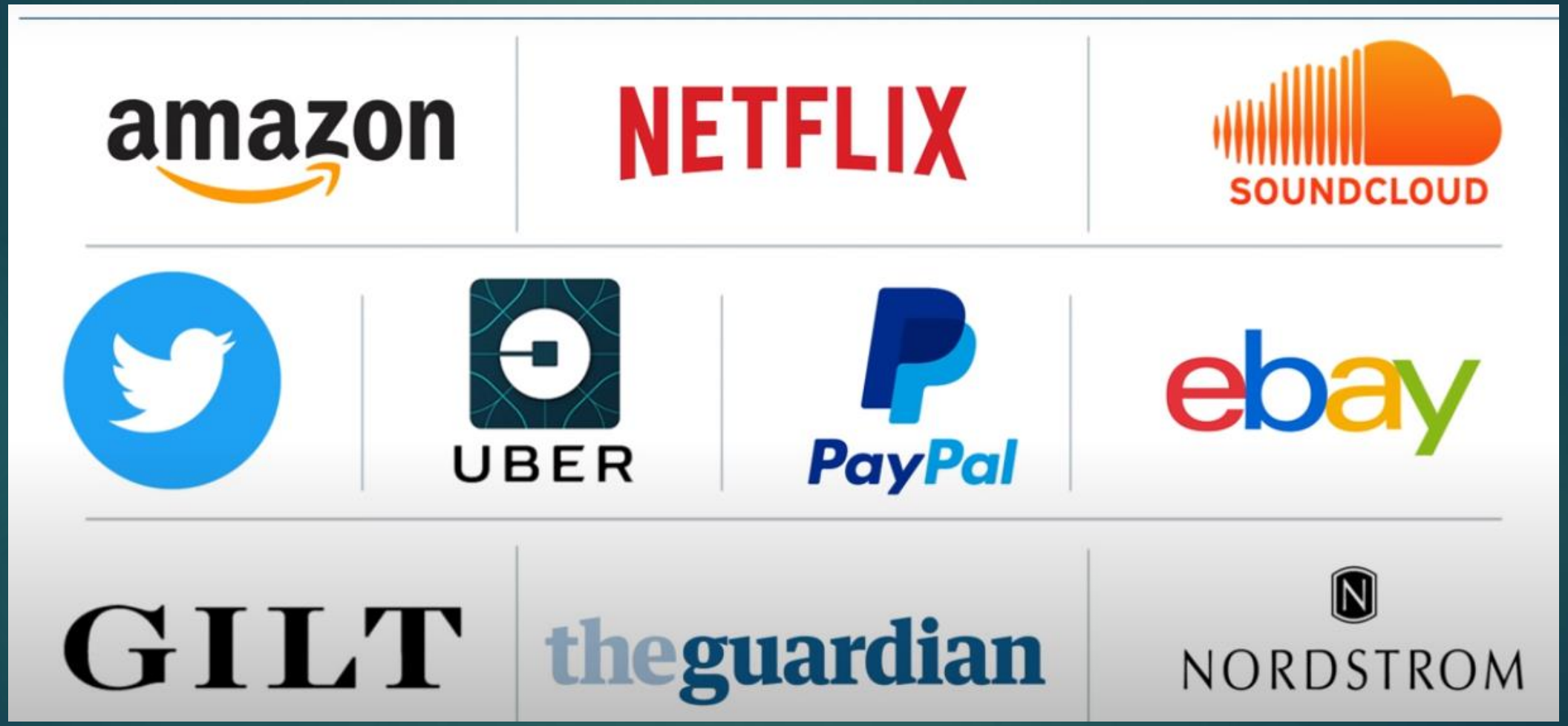
Advantages of Microservices Architecture



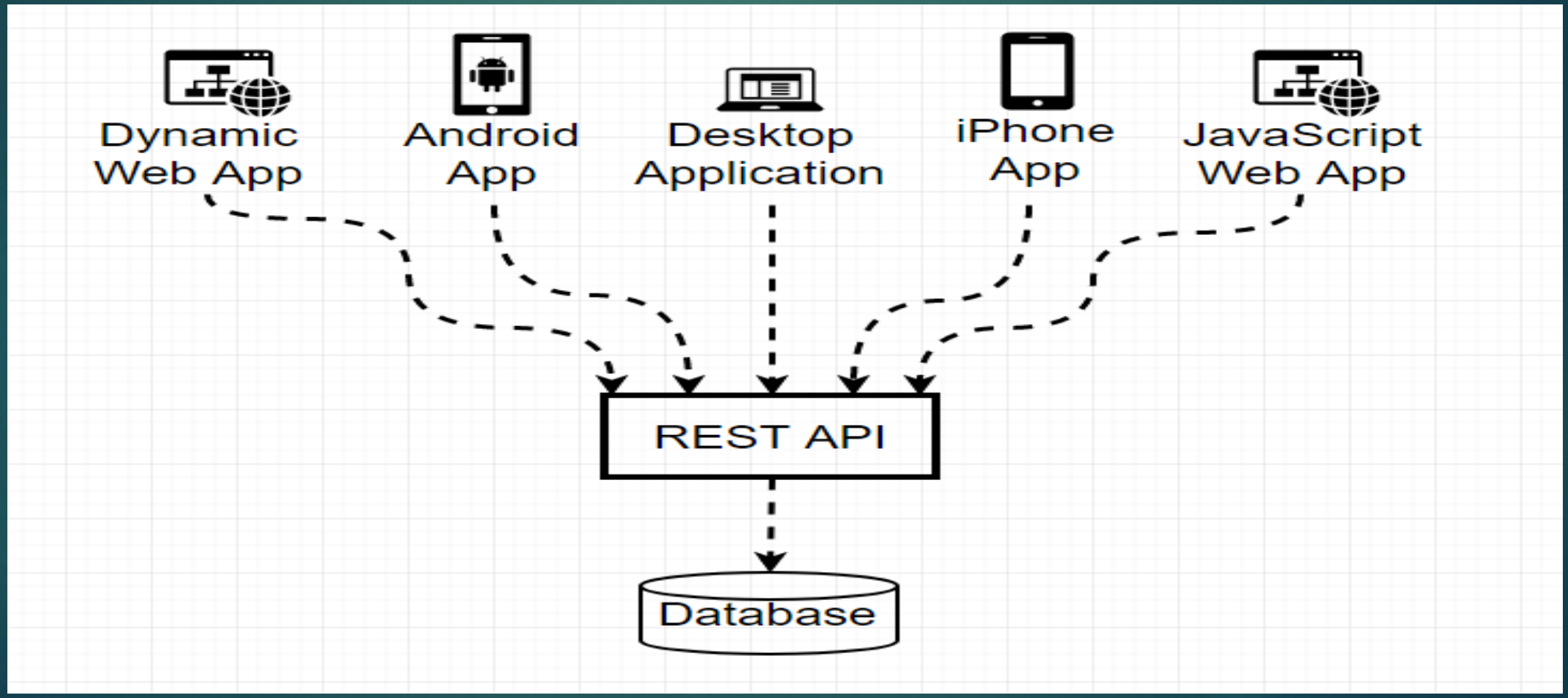
How does Amazon work?



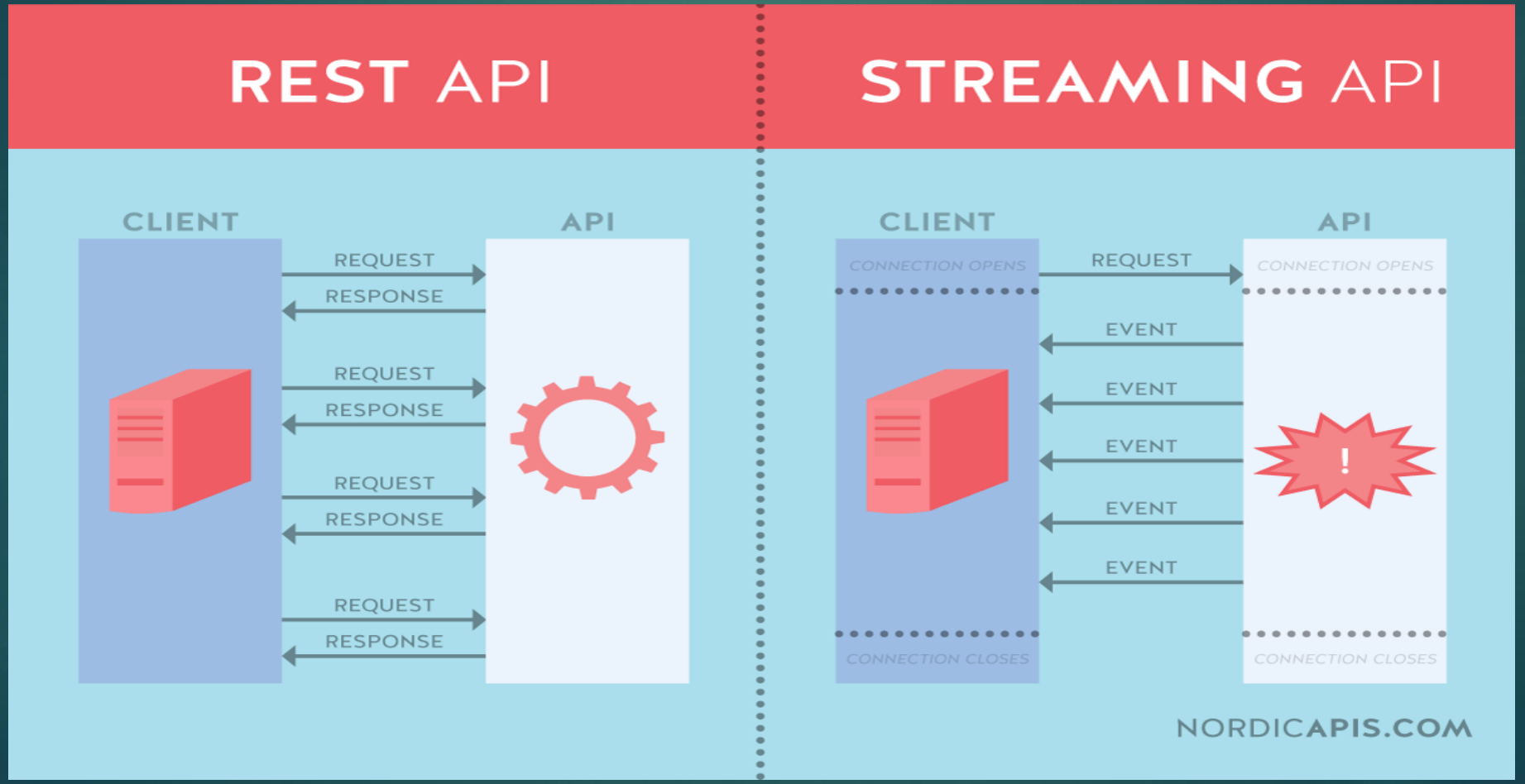
Companies using Microservices



REST API – The REpresentational State Transfer



REST vs Streaming API.



A few things mentioned during discussion:

- Server Side Processing
- Client Side Processing
- Application Programming Interface (API)
- https://developer.twitter.com/en/docs/tweets/post-and-engage/overview
- https://cloud.google.com/apigee
- https://developers.facebook.com/apps/
- https://httpstatuses.com
- https://jsoneditoronline.org/
- https://github.com/syedsaadahmed/PSA-Koblenz-Python-Training
- https://restfulapi.net/resource-naming