

The QuoteServices App Case Study: Overview of the API Gateway Pattern

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

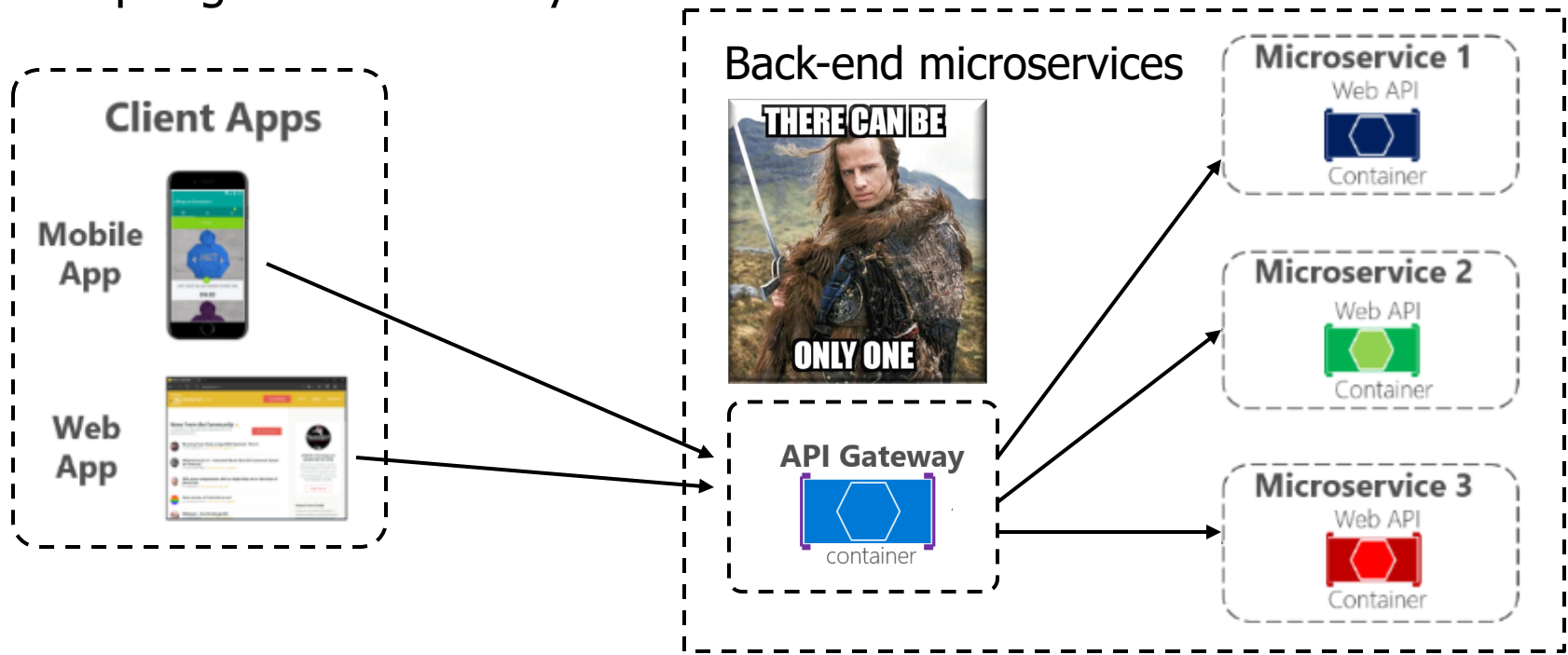
**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

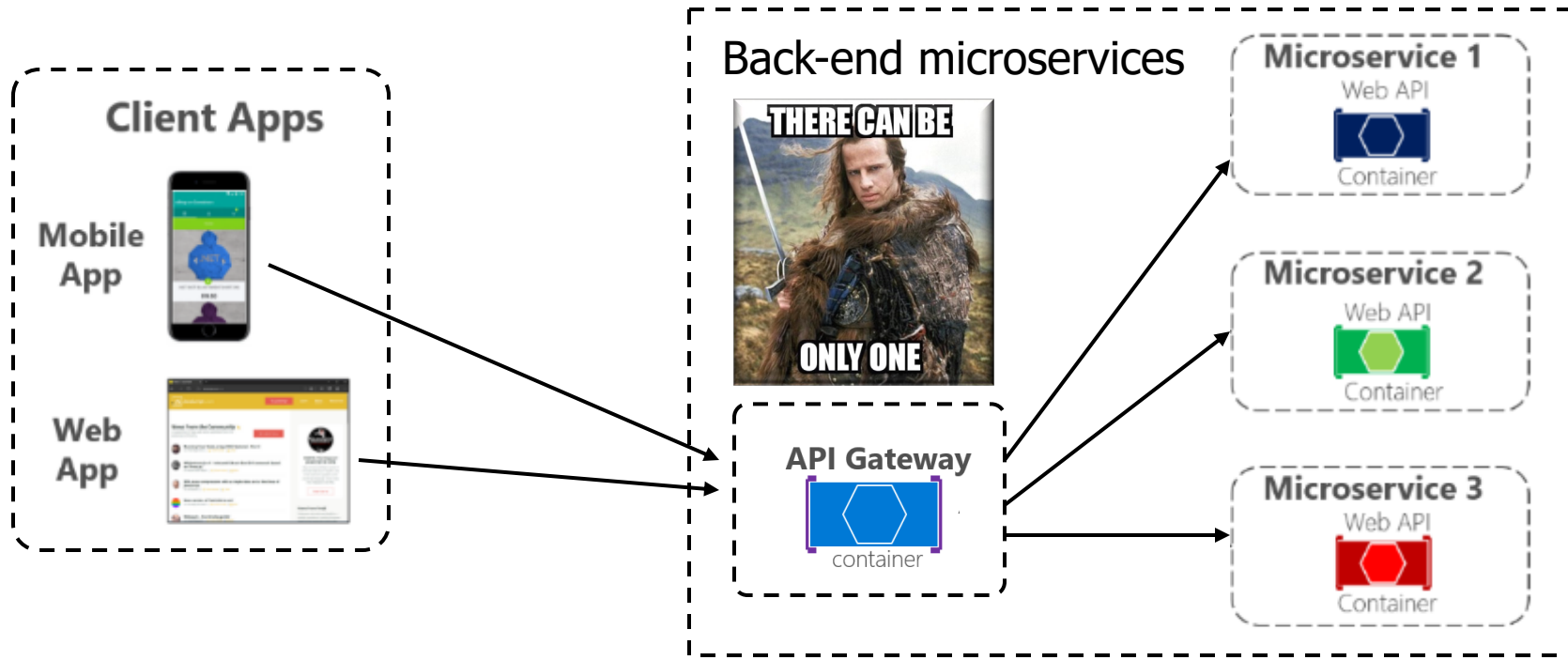
- Recognize the API gateway pattern & common realizations of this pattern in the Spring Cloud Gateway



Overview of the API Gateway Pattern

Overview of the API Gateway Pattern

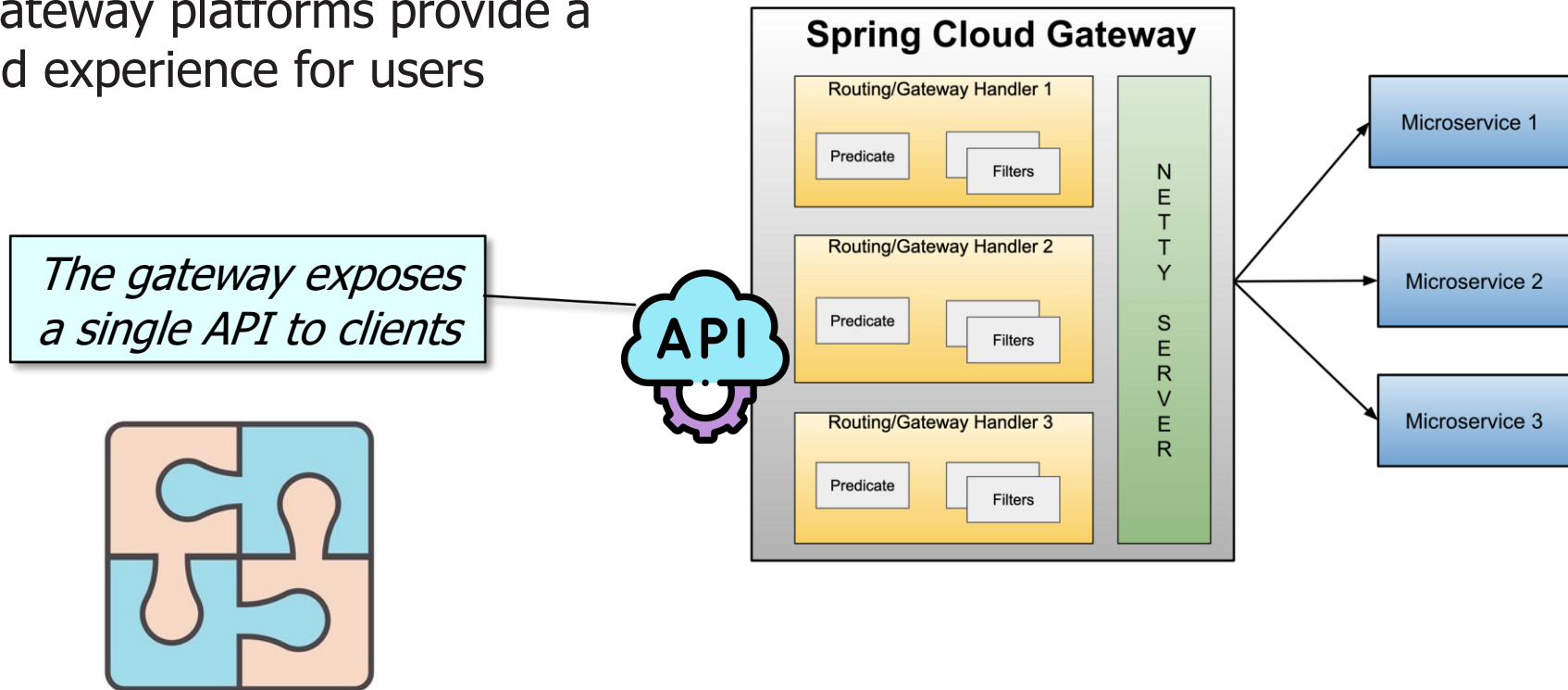
- The API gateway pattern provides a single entry point for back-end APIs & their microservice implementations



See microservices.io/patterns/apigateway.html

Overview of the API Gateway Pattern

- API gateway platforms provide a unified experience for users



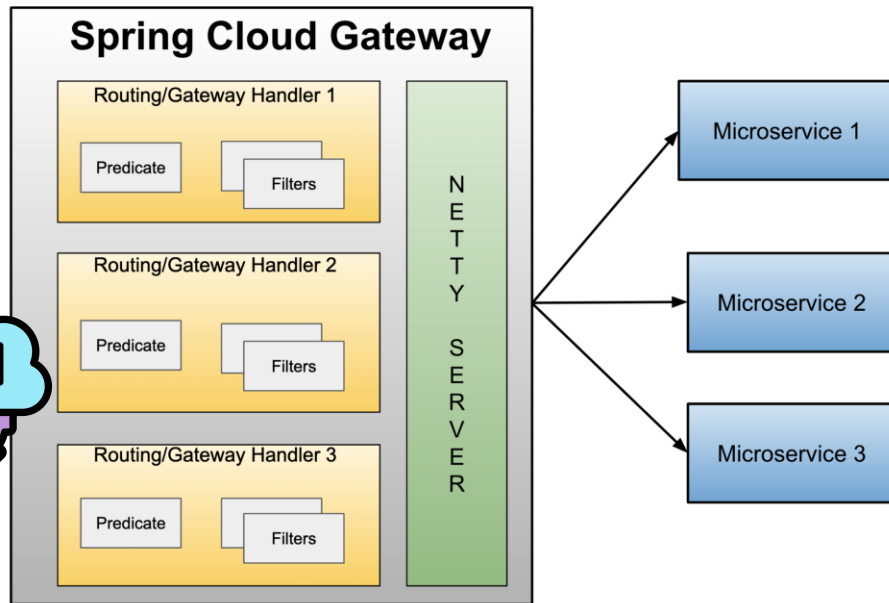
See blog.knoldus.com/spring-cloud-api-gateway

Overview of the API Gateway Pattern

- API gateway platforms provide a unified experience for users, i.e.
 - Takes all client API requests

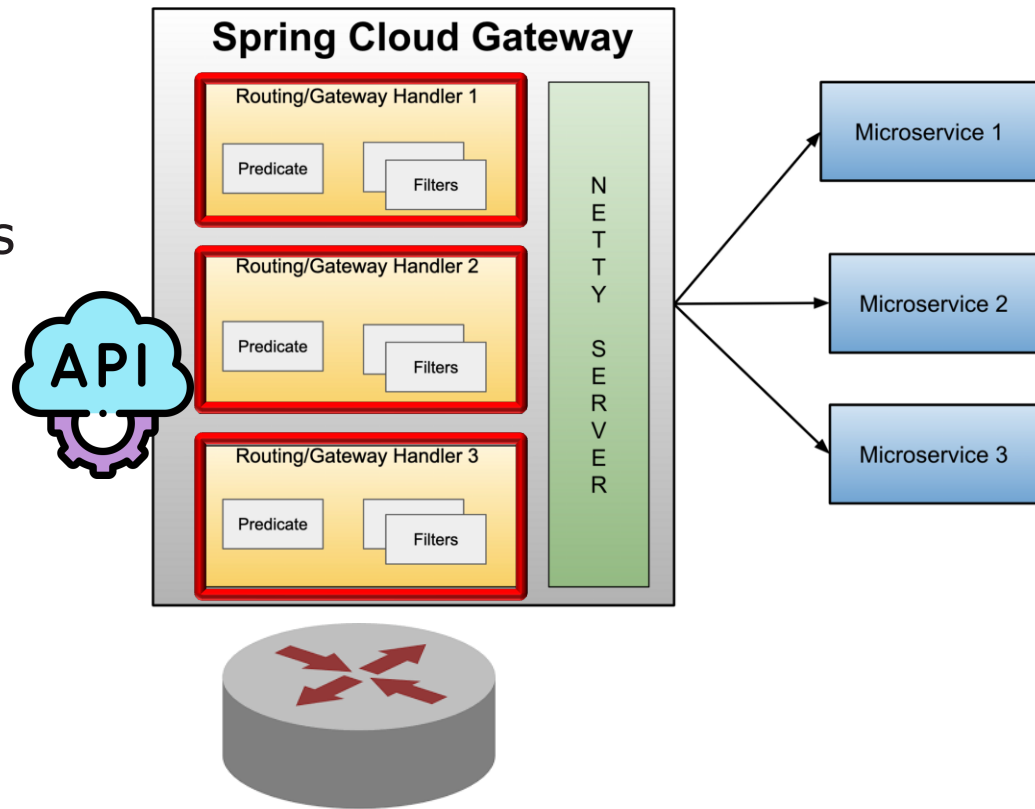


Customer accessing different urls



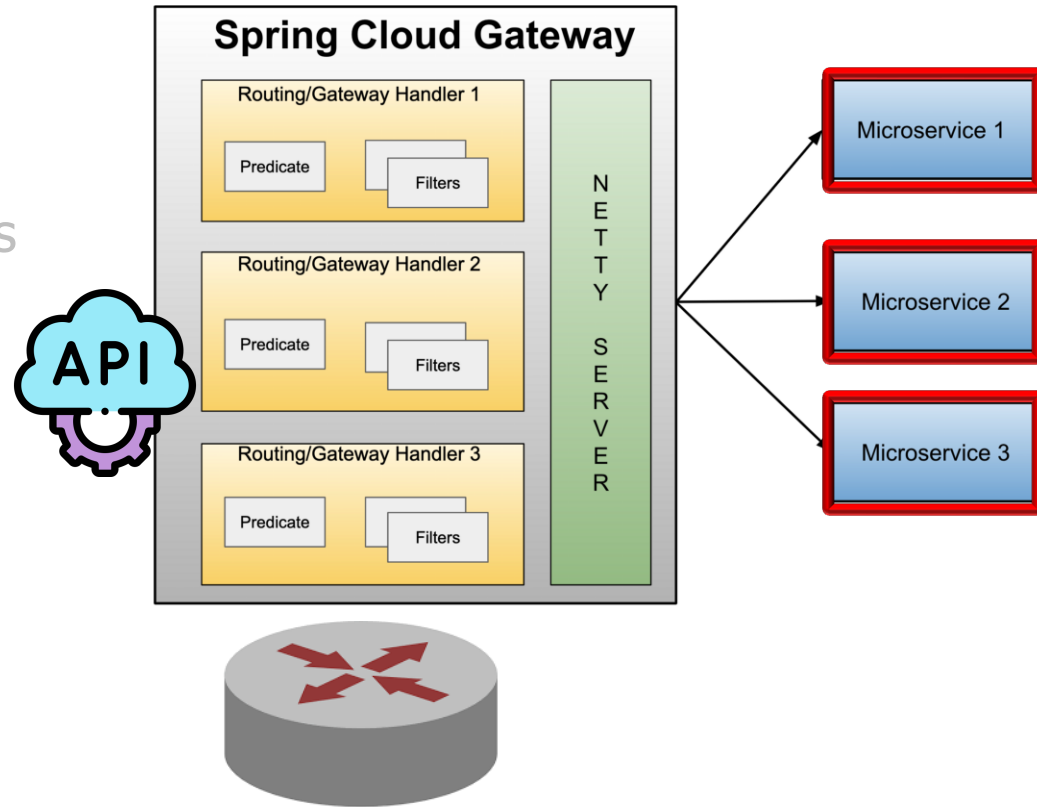
Overview of the API Gateway Pattern

- API gateway platforms provide a unified experience for users, i.e.
 - Takes all client API requests
 - Determines which microservices are designated by the requests



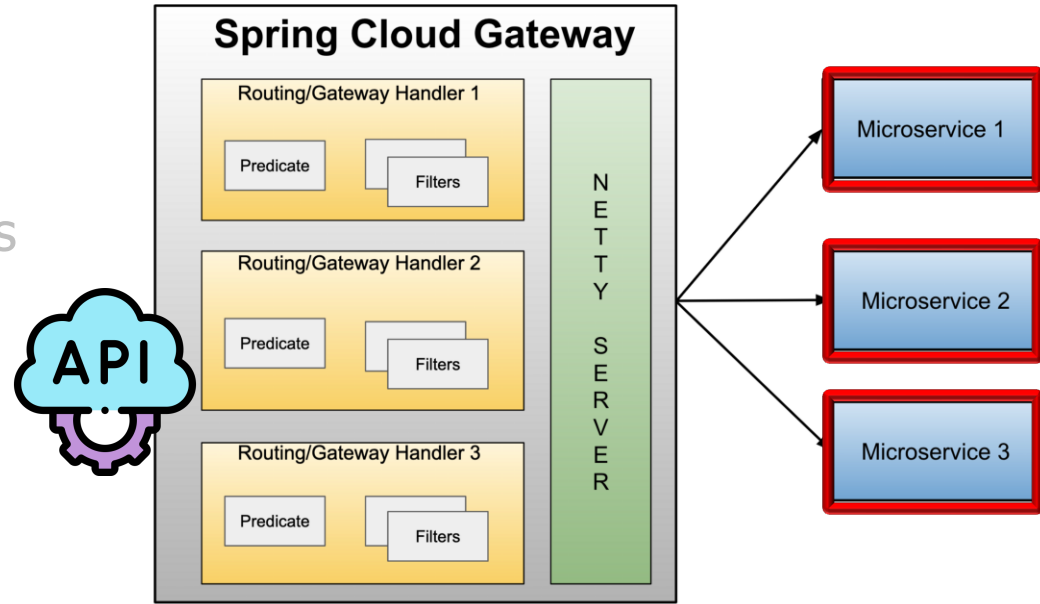
Overview of the API Gateway Pattern

- API gateway platforms provide a unified experience for users, i.e.
 - Takes all client API requests
 - Determines which microservices are designated by the requests
 - Routes requests to designated backend microservices



Overview of the API Gateway Pattern

- API gateway platforms provide a unified experience for users, i.e.
 - Takes all client API requests
 - Determines which microservices are designated by the requests
- Routes requests to designated backend microservices
 - Microservices can be either external and/or internal to the system



Pros & Cons of Using an API Gateway

Pros & Cons of Using an API Gateway

- Pros of an API gateway

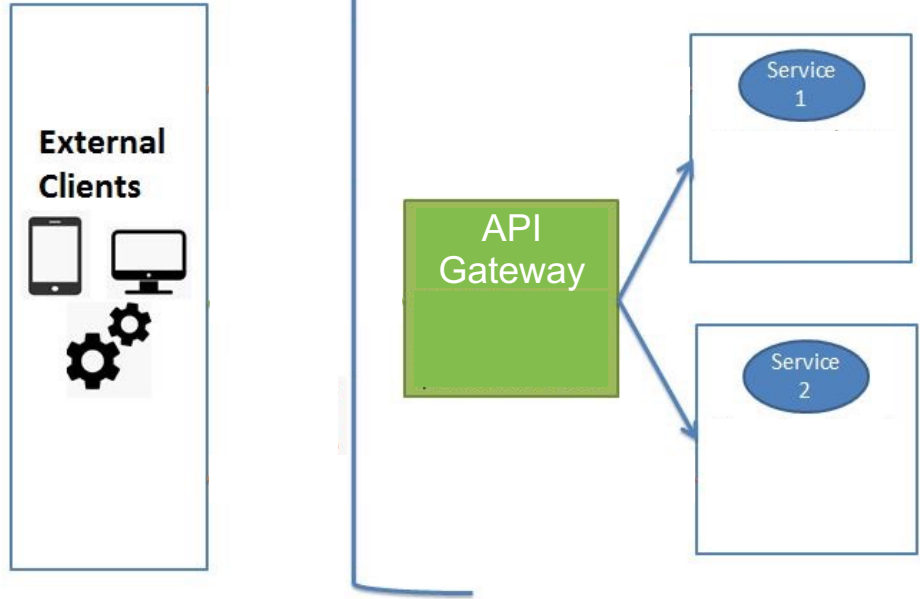


See dashbird.io/knowledge-base/api-gateway/pros-and-cons-of-using-an-api-gateway

Pros & Cons of Using an API Gateway

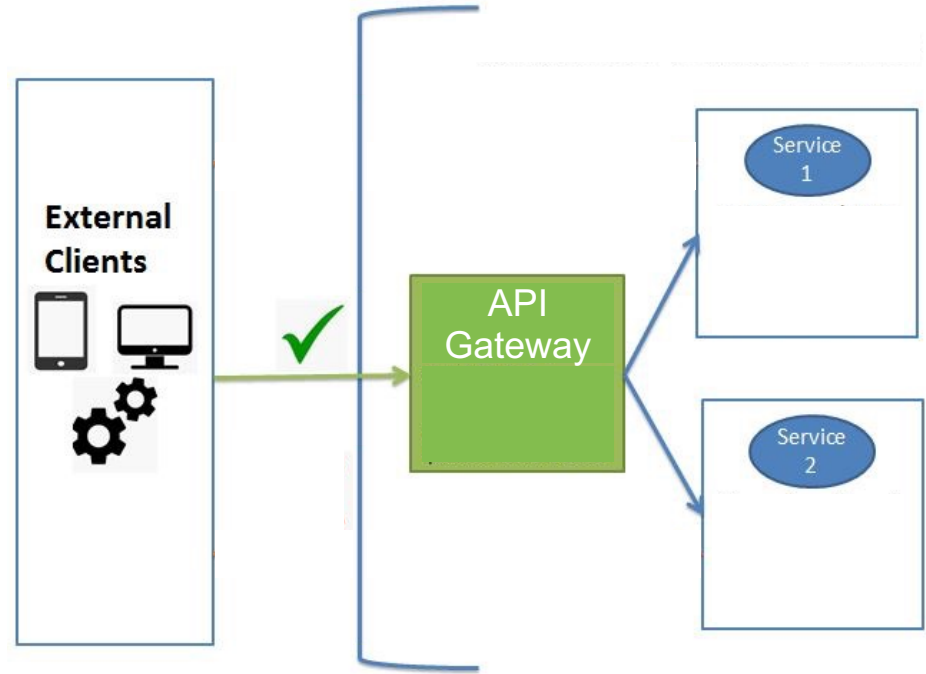
- Pros of an API gateway
 - Simplifies client interaction

SIMPLIFY



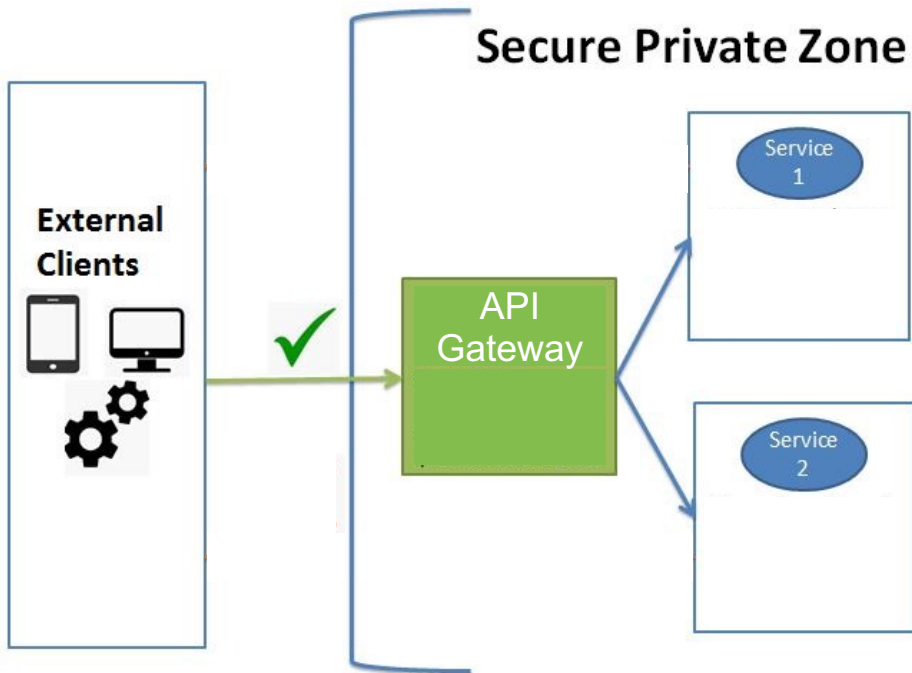
Pros & Cons of Using an API Gateway

- Pros of an API gateway
 - Simplifies client interaction
 - i.e., only a single service API is needed to access all system capabilities



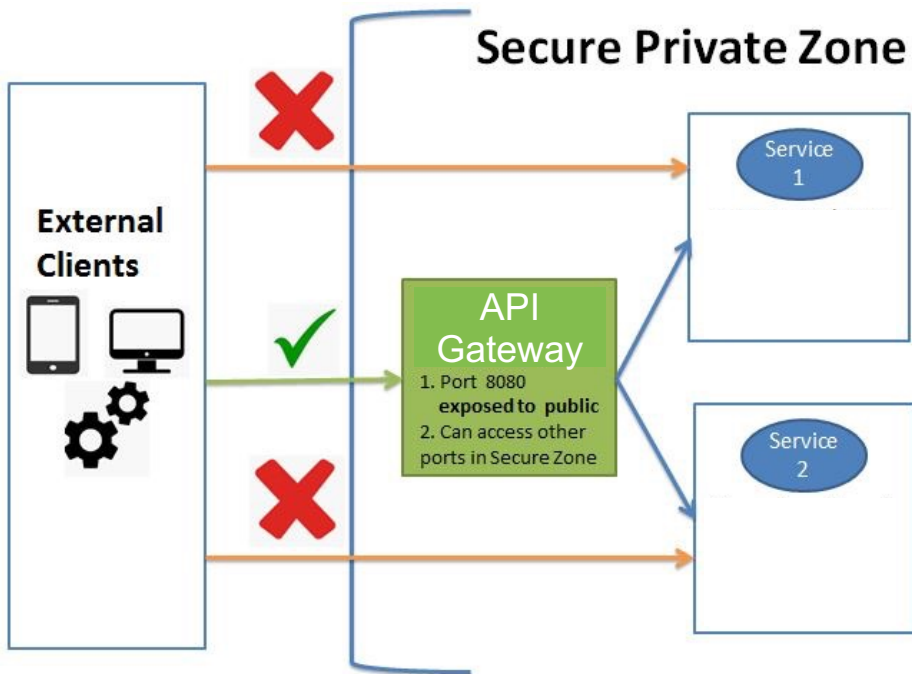
Pros & Cons of Using an API Gateway

- Pros of an API gateway
 - Simplifies client interaction
 - It improves microservice security



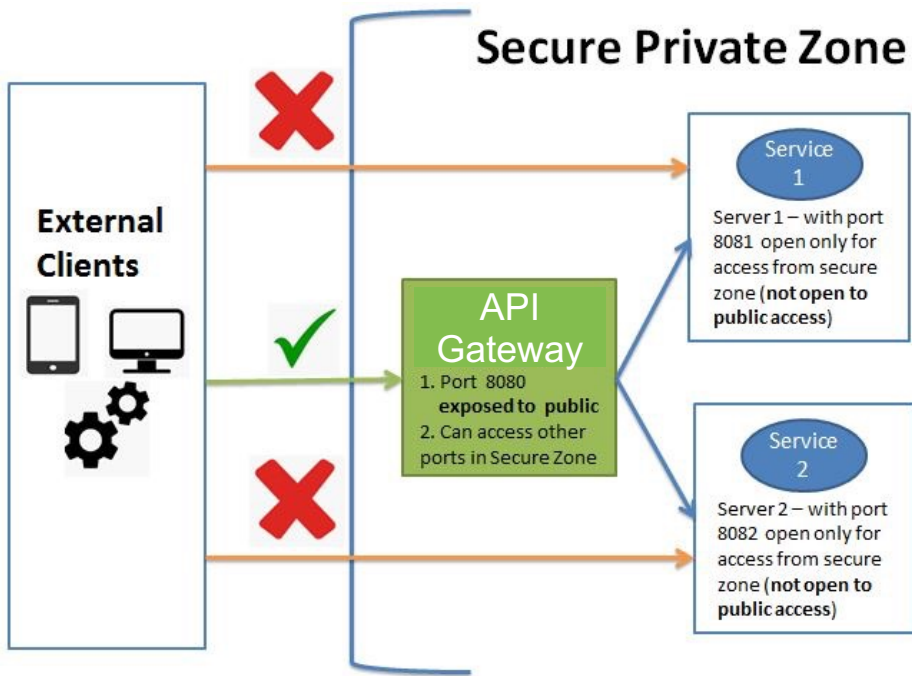
Pros & Cons of Using an API Gateway

- Pros of an API gateway
 - Simplifies client interaction
 - It improves microservice security
 - Limits access of external client calls via one public port number



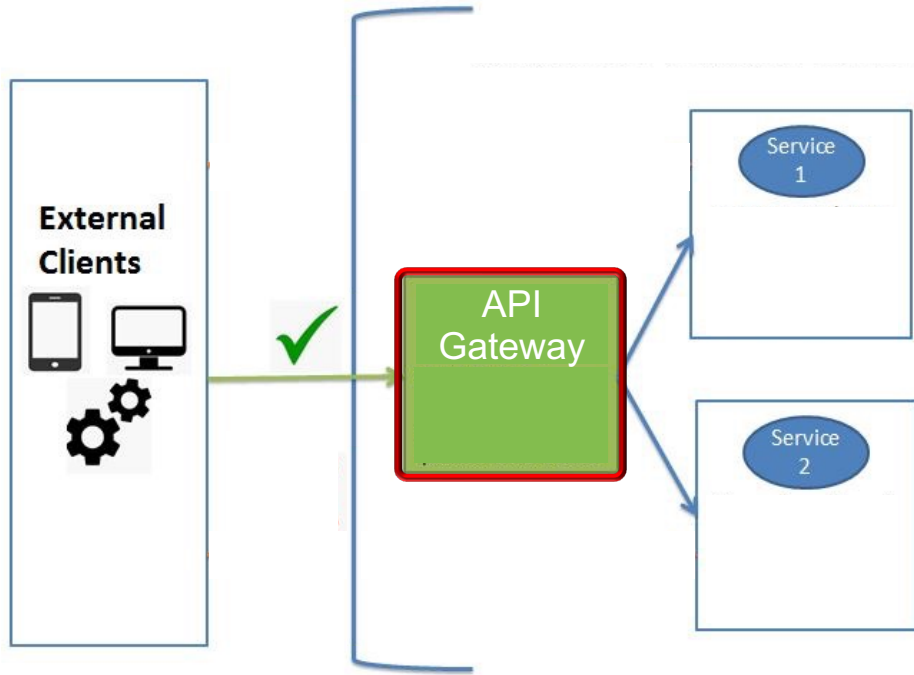
Pros & Cons of Using an API Gateway

- Pros of an API gateway
 - Simplifies client interaction
 - It improves microservice security
 - Limits access of external client calls via one public port number
 - Client needn't know internal system architecture
 - e.g., they can't determine the location of microservice instances



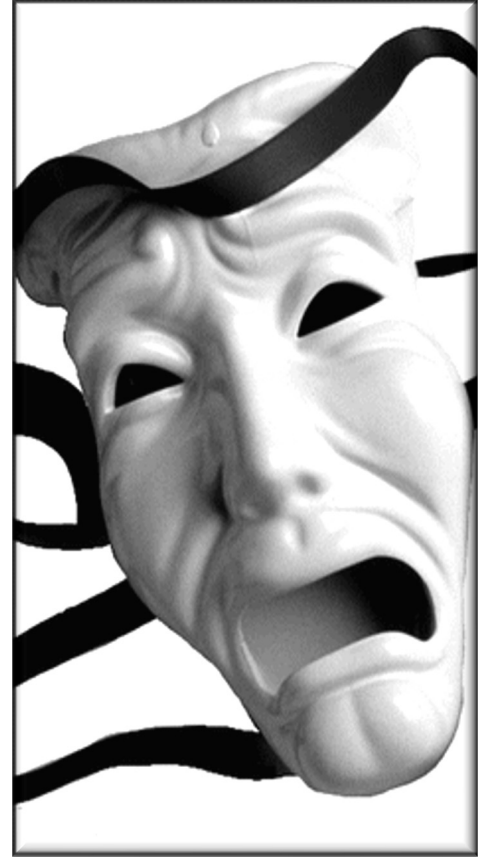
Pros & Cons of Using an API Gateway

- Pros of an API gateway
 - Simplifies client interaction
 - It improves microservice security
 - Cross-cutting concerns only need be implemented once in the API gateway since all calls will be routed through it
 - e.g., authentication, monitoring, load-balancing, & resiliency



Pros & Cons of Using an API Gateway

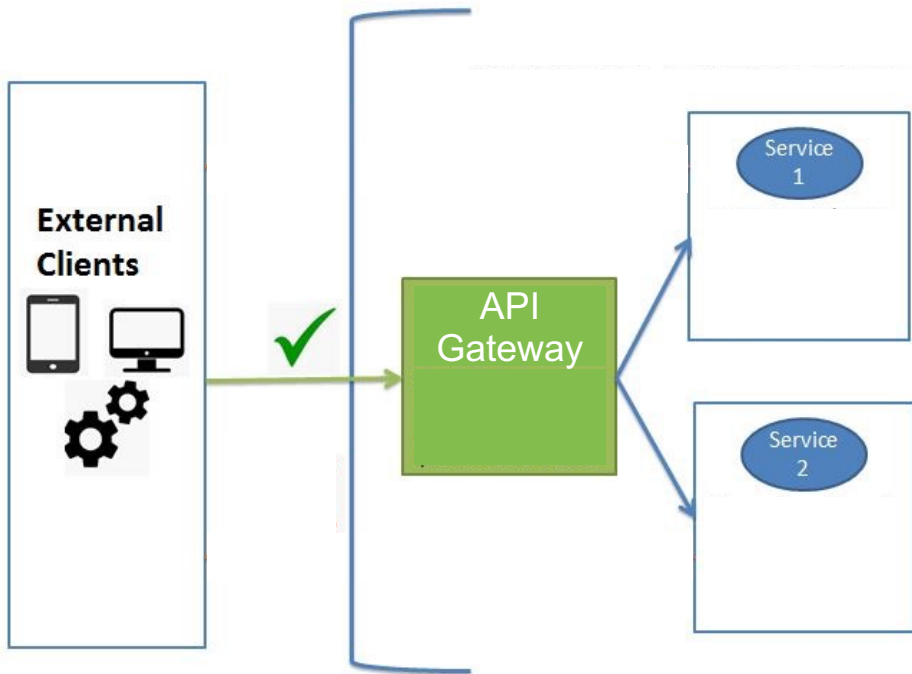
- Cons of an API gateway



See dashbird.io/knowledge-base/api-gateway/pros-and-cons-of-using-an-api-gateway

Pros & Cons of Using an API Gateway

- Cons of an API gateway
 - Involves more “moving parts” & associated learning curve



Pros & Cons of Using an API Gateway

- Cons of an API gateway
 - Involves more "moving parts" & associated learning curve
 - The application*.yml files can be very persnickety..

routes:

- **id:** handey

predicates:

- Path= /handey/**

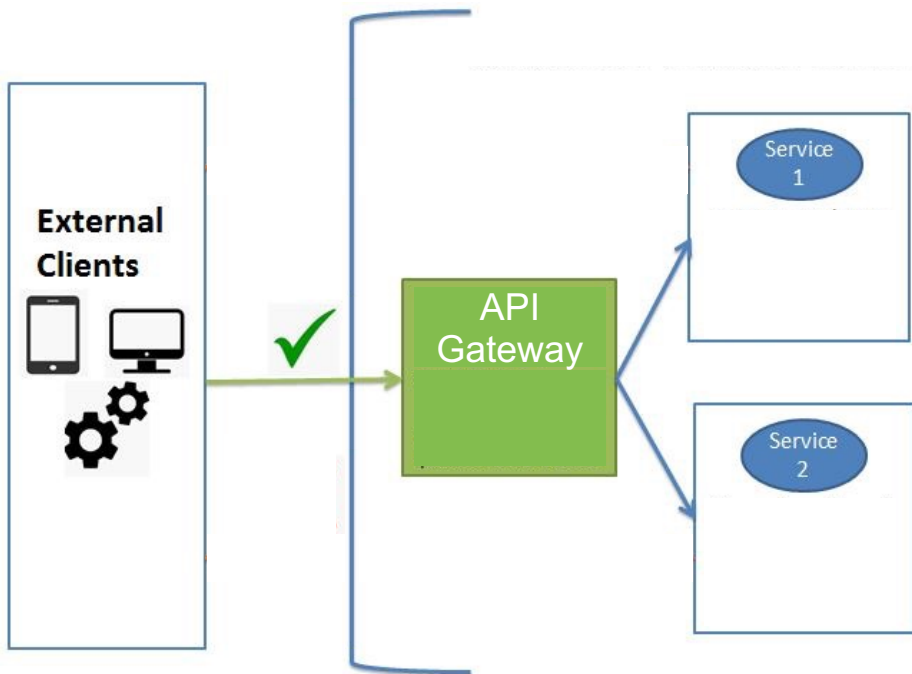
filters:

- StripPrefix=1 # ...

uri: http://localhost:9100

-

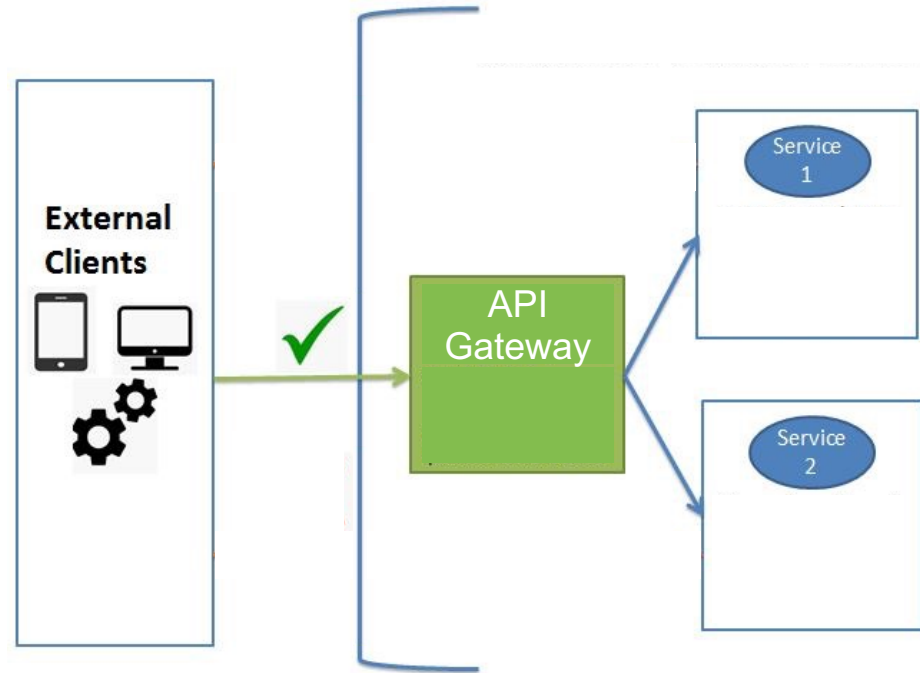
A subtle syntax error..



See next part of the lesson on "*Gateway Microservice Structure & Functionality*"

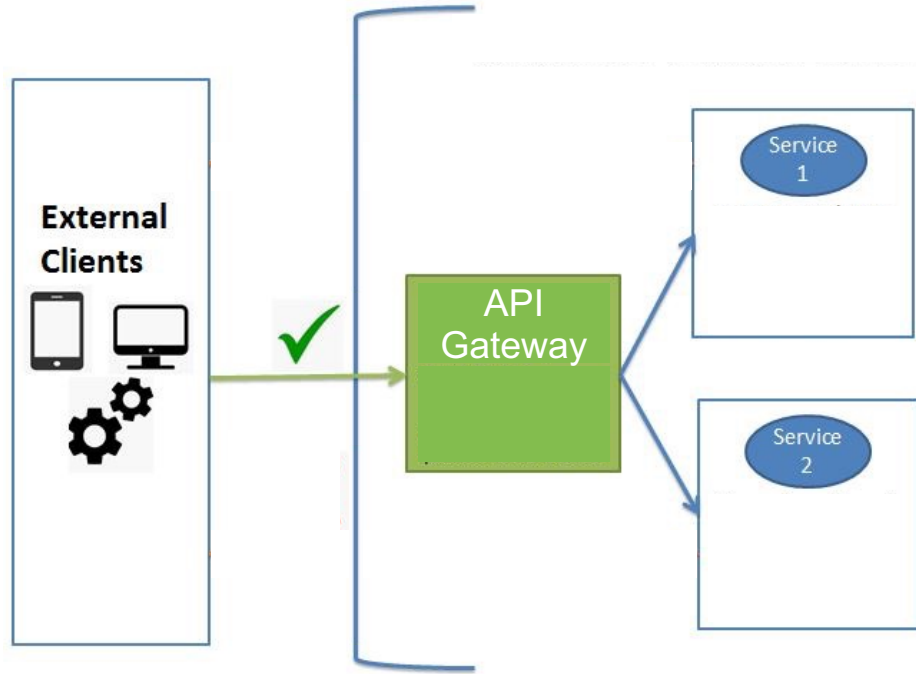
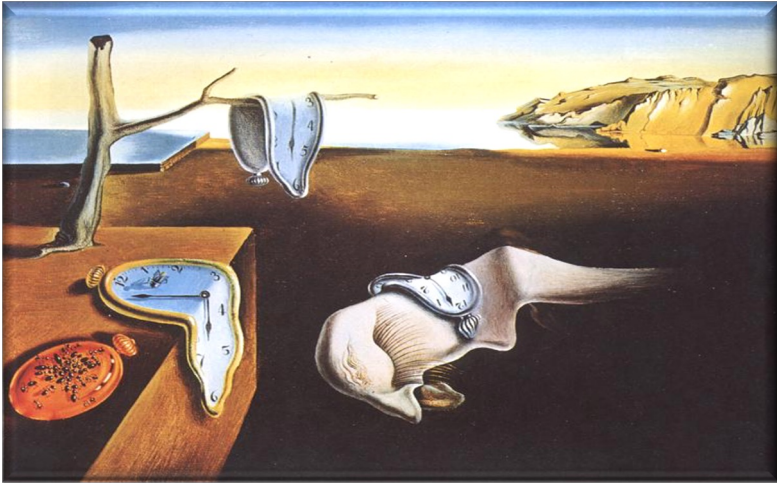
Pros & Cons of Using an API Gateway

- Cons of an API gateway
 - Involves more “moving parts” & associated learning curve
 - The application*.yml files can be very persnickety..
 - Requires additional deployment & orchestration mechanisms



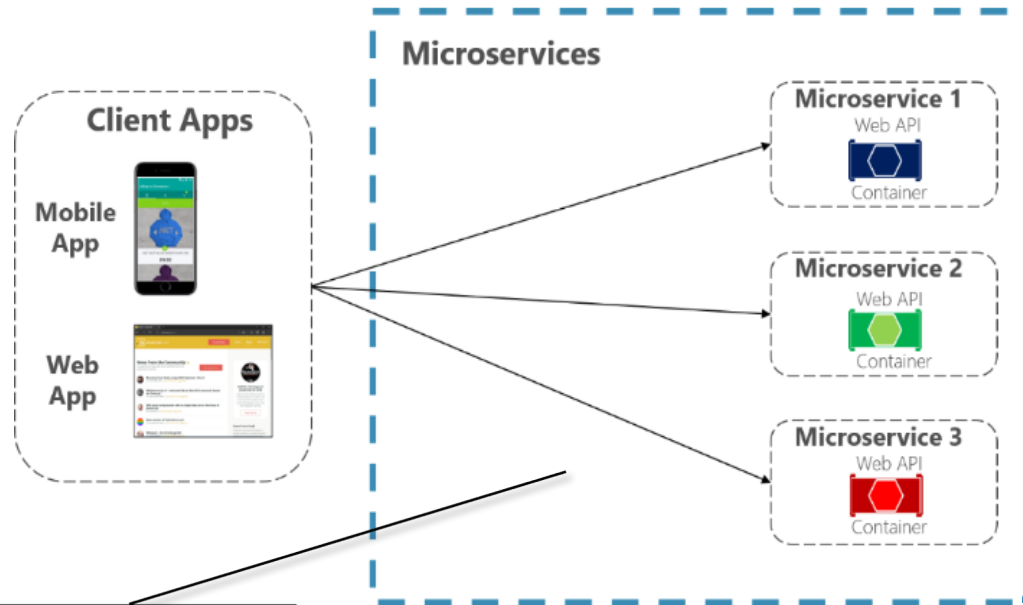
Pros & Cons of Using an API Gateway

- Cons of an API gateway
 - Involves more “moving parts” & associated learning curve
 - Extra levels of indirection can degrade performance



Pros & Cons of Using an API Gateway

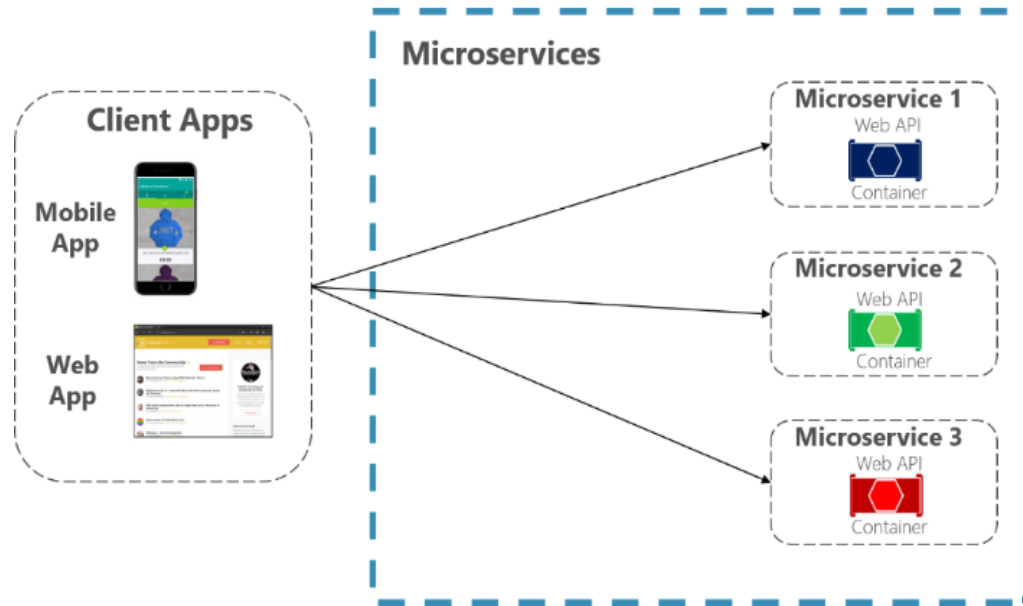
- Cons of an API gateway
 - Involves more “moving parts” & associated learning curve
 - Extra levels of indirection can degrade performance
 - Compared with a direct-client-to-microservice model



Each microservice has a public endpoint, e.g., with a different TCP port & host address

Pros & Cons of Using an API Gateway

- Cons of an API gateway
 - Involves more “moving parts” & associated learning curve
 - Extra levels of indirection can degrade performance
 - Compared with a direct-client-to-microservice model
 - However, only viable for simple/small deployments



IT'S SIMPLE

End of the QuoteServices App Case Study: Overview of the API Gateway Pattern