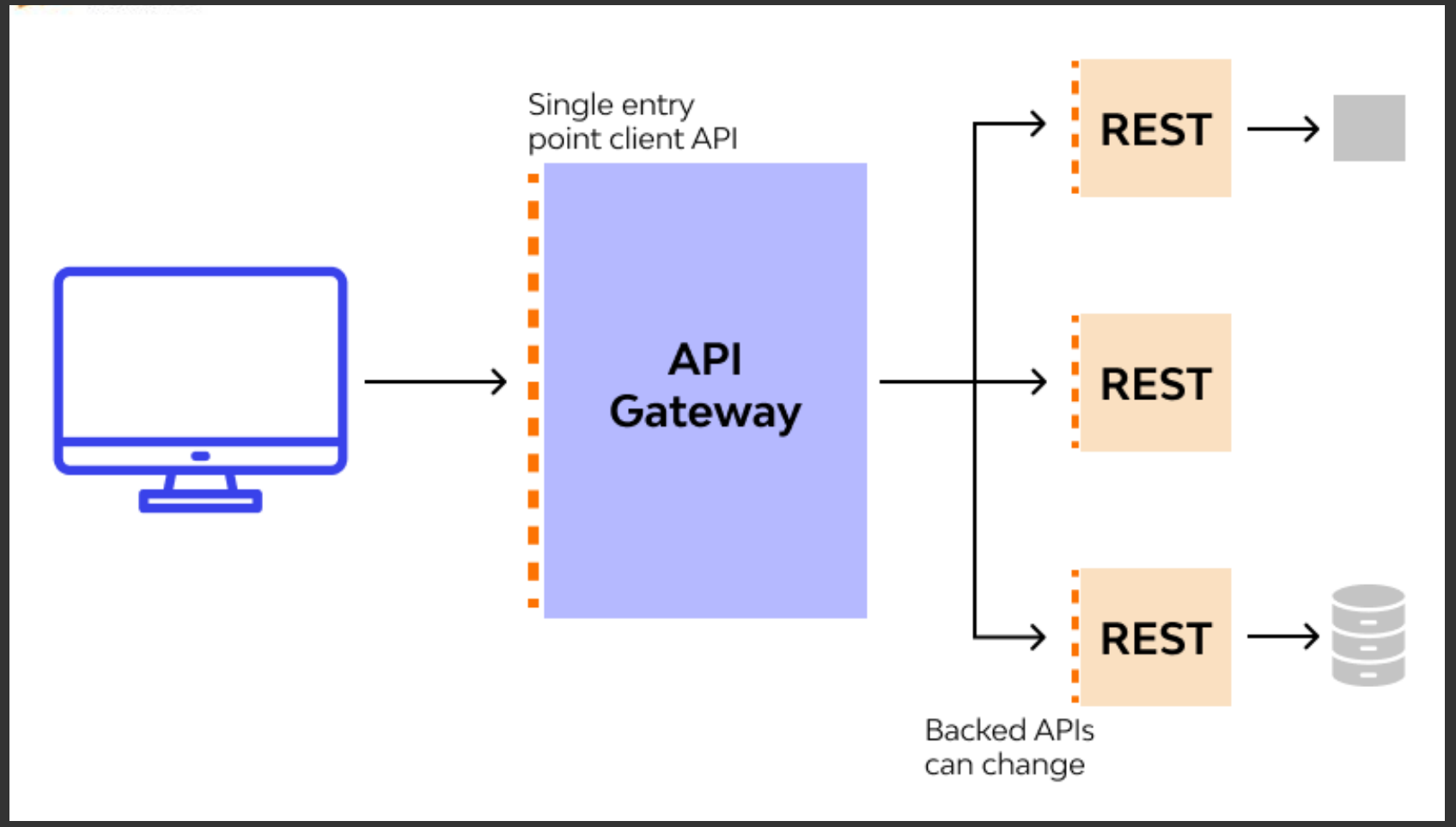
Microservice

Spring Cloud API Gateway



API Gateway

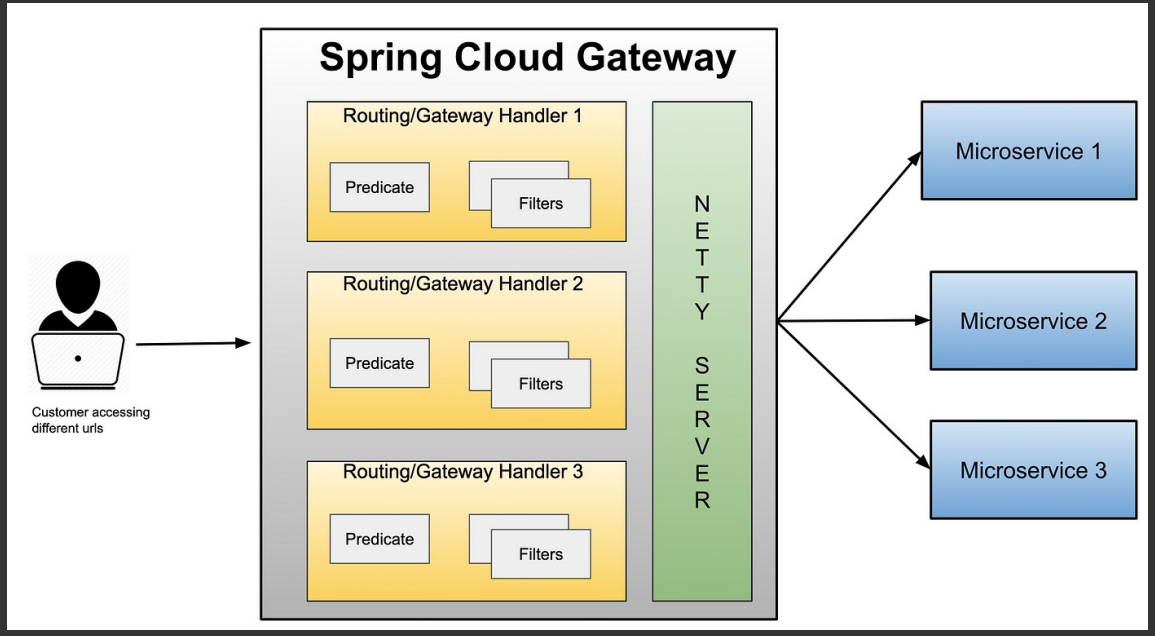
APIs are a common way of communication between applications. In the case of microservice architecture, there will be a number of services and the client has to know the hostnames of all underlying applications to invoke them.

To simplify this communication, we prefer a component between client and server to manage all API requests called API Gateway. Additionally, we can have other features which include:

- Security – Authentication, authorization
- Routing – routing, request/response manipulation, circuit breaker
- Observability – metric aggregation, logging, tracing

Spring Cloud API Gateway

Spring Cloud API Gateway is a powerful, flexible solution for routing and proxying requests to downstream services in a microservices architecture. It handles several important tasks like routing, filtering, authentication, and load balancing.



Spring Cloud Gateway Building Blocks

Spring Cloud Gateway consists of 3 main building blocks:

1. Route
2. Predicate
3. Filters

```yaml
spring:
  cloud:
    gateway:
      routes:
        - id: order-service
          uri: lb://ORDER-SERVICE
          predicates:
            - Path=/api/v1/orders/**
          filters:
            - AddRequestHeader=X-Custom-Header, Anuj
        - id: inventory-service
          uri: lb://INVENTORY-SERVICE
          predicates:
            - Path=/api/v1/inventory/**
        - id: no-service
          uri: lb://NO-SERVICE
          predicates:
            - Path=/api/v1/no-service/**
          filters:
            - AddRequestHeader=X-Custom-Header, Anuj
            - RedirectTo=302, https://codingshuttle.com
```

Spring Cloud Gateway Route

1. Route: Think of this as the destination that we want a particular request to route to. It comprises of destination URI, a condition that has to satisfy – Or in terms of technical terms, Predicates, and one or more filters.

Spring Cloud Gateway Predicate

2. Predicate: This is literally a condition to match. i.e. kind of “if” condition..if requests has something – e.g. path=blah or request header contains foo-bar etc.

Predicates with path: - Path=/api/v1/orders/**

Predicates with Method: - Method=GET

Predicates with Header: - Header=User-Agent, Mozilla/*

Spring Cloud Gateway Filter

3. Filter: These are instances of Spring Framework WebFilter. This is where you can apply your magic of modifying request or response. There are quite a lot of out of box WebFilter that framework provides.

```
filters:
        - AddRequestHeader=X-Request-Id, 12345
        - AddResponseHeader=X-Response-id, abcd
        - RedirectTo=302, https://youtube.com
        - StripPrefix=1
        - RemoveRequestHeader=Cookie
```