Spring Gateway as the perfect gateway solution

Dan Erez

May 2024

Who am I?

- Lead Architect @ZIM
- +20 years of software development
  - Native language: Java
  - Enterprise Architecture
  - Cloud Architecture & Cost management
- Speaker
- Trying to innovate…See in medium (dan.erez)

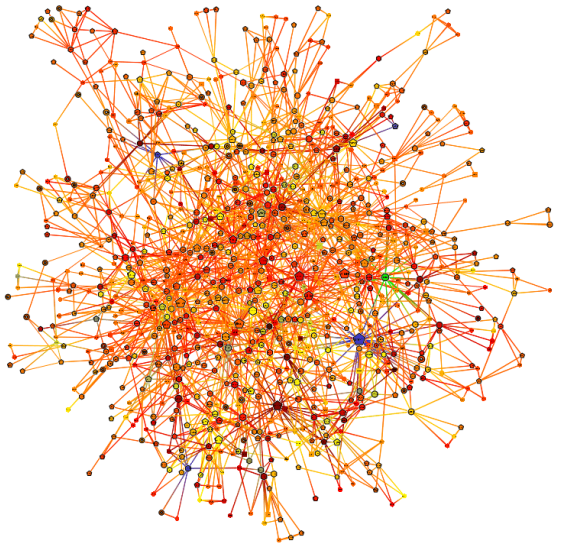
Agenda

- About API Gateways
- What are your options?
- Why & When Spring Gateway?
- Demo
- Q&A

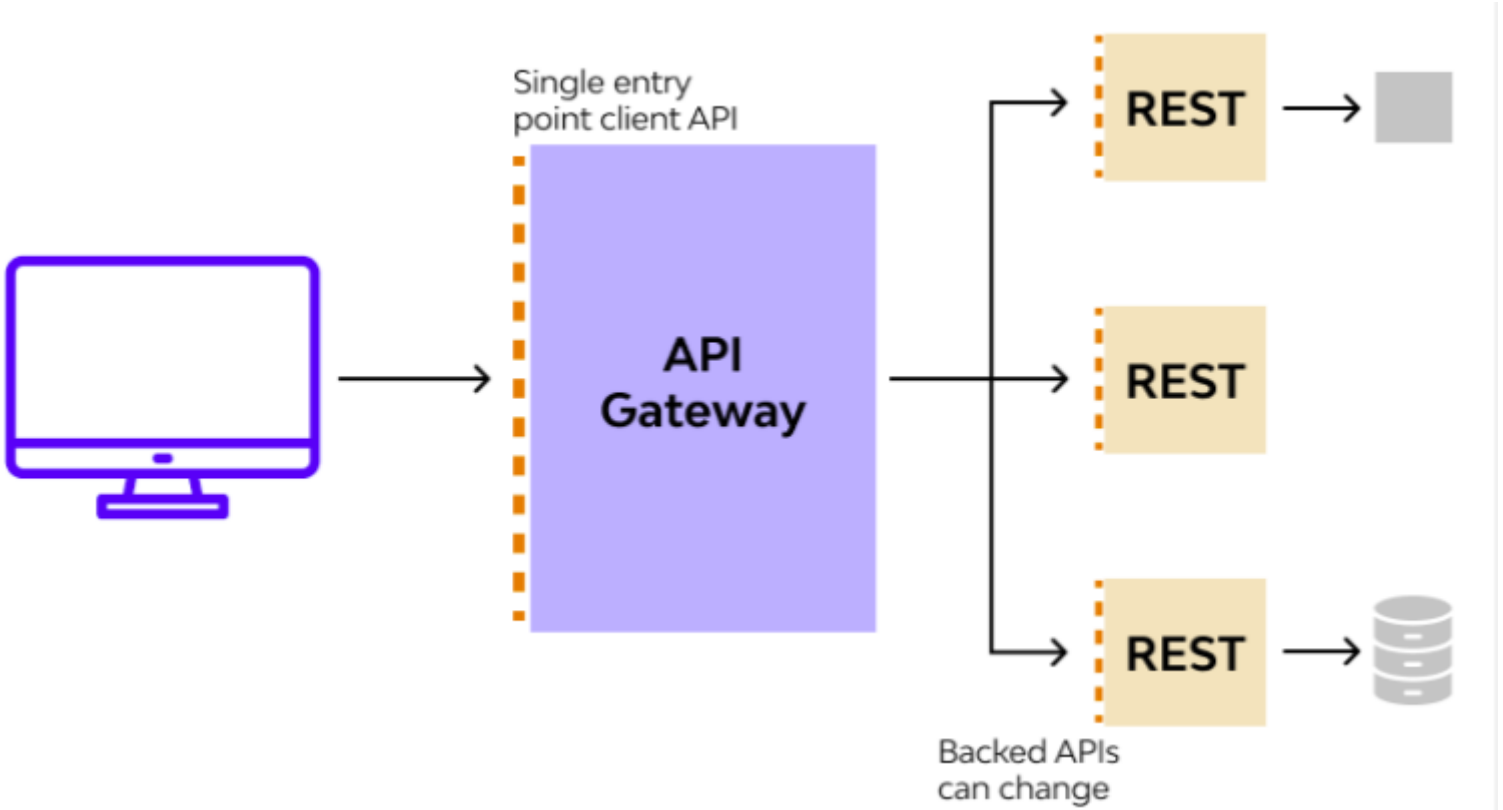

THERE IS NO PLACE LIKE
127.0.0.1

About API Gateways

- Once there was a monolith…
- Enter micro services architecture
- Too many?
- One gateway to rule them all!



API Gateway

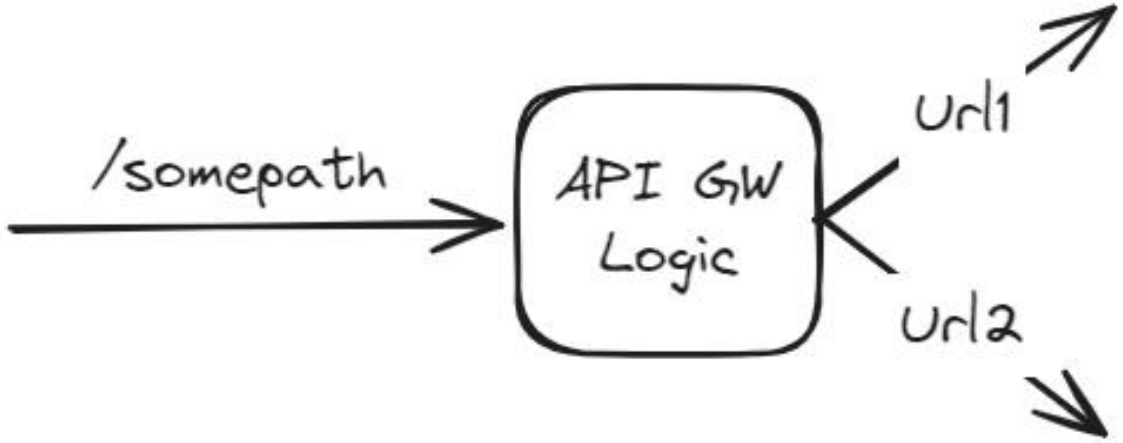

Single entry
point client API
API
Gateway
REST
REST
REST
Backed APIs
can change

What can it do for you?

- Routing
- Enriching headers & data
- Authentication & Authorization
- Caching
- Rate limiting, throttling, circuit breakers
- Load Balancing
- Versioning
- Business flow control
- Much more…

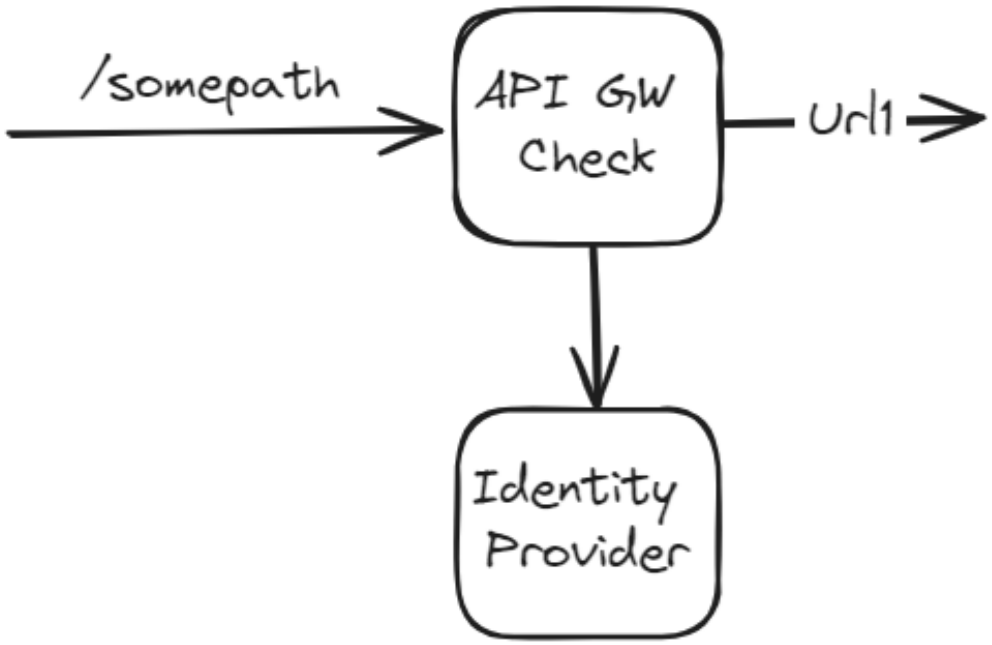
Routing

- Simple routing
- Filter by:
  - Path
  - Header
  - Data
- External routing



Authentication & Authorization

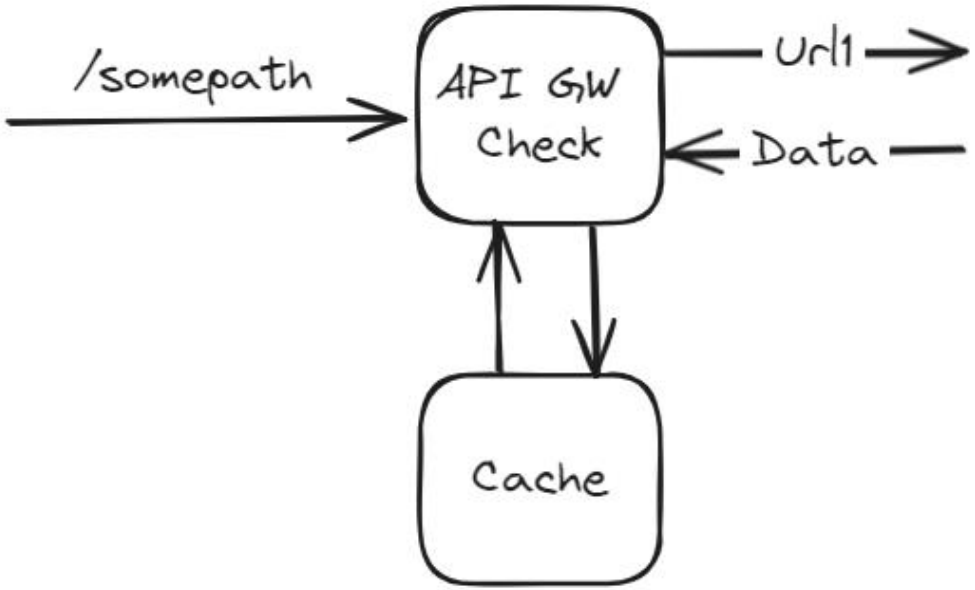
- Prove who you are:
  - OAuth2
  - Kerberos
  - Whatever you want
- Know your roles



- Tokens (JWT, Opaque Tokens)

Caching

- Redis based simple caching
- Smart caching
- Read through, read aside

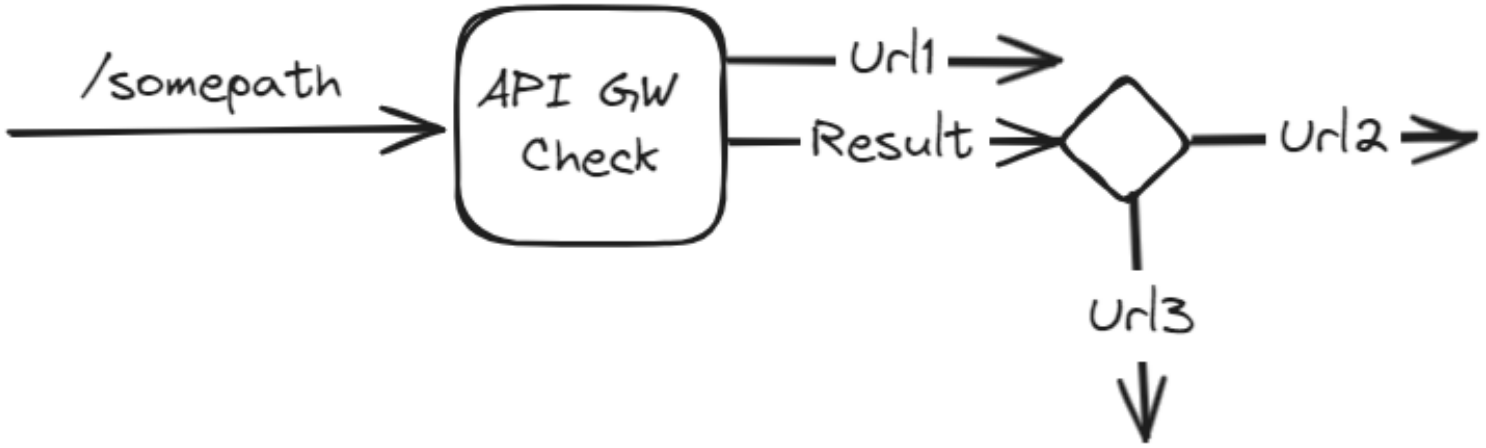


Requests Flow control

- Rate limiting
- Throttling, smart throttling
- Circuit breakers

Business Flow Control

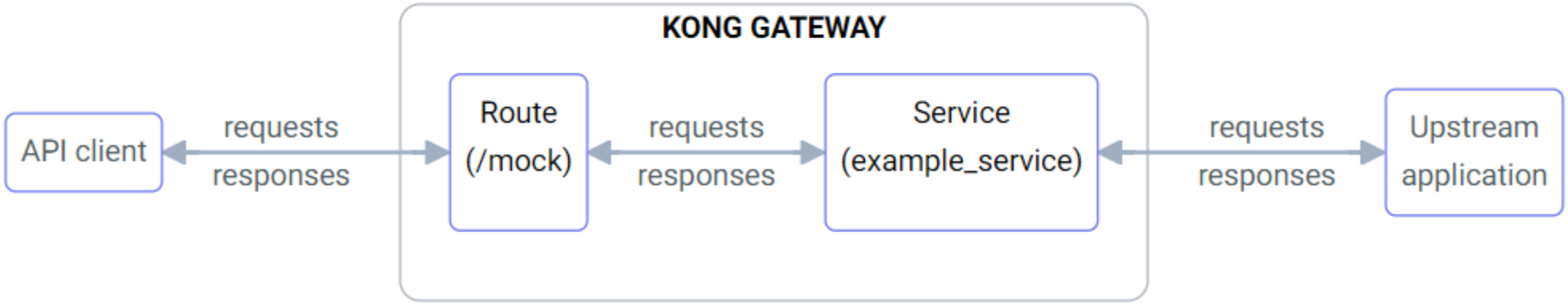
- Small building blocks
- Aggregate or chain them together
- Apply logic



API Gateway Options – on prem

- WSO2
- Kong
- Nginx Plus
- IBM API Connect
- Mulesoft Anypoint

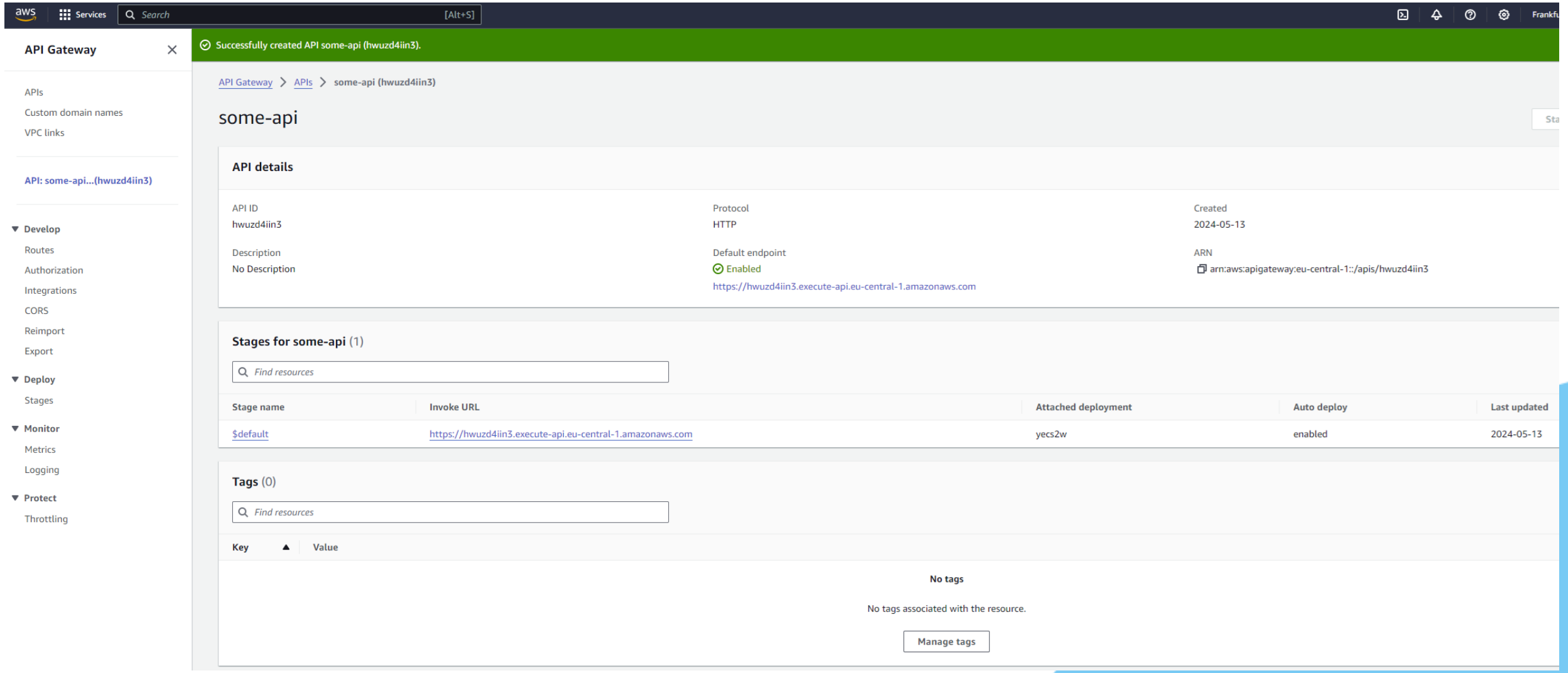
Kong API Path



API Gateway Options - cloud

- Azure Gateway
- AWS Gateway
- Google cloud Apigee
- Cloudflare API Gateway

AWS Gateway


aws
Services
Search
[Alt+S]
API Gateway
Successfully created API some-api (hwuzd4iin3).
APIs
Custom domain names
VPC links
API: some-api...(hwuzd4iin3)
Develop
Routes
Authorization
Integrations
CORS
Reimport
Export
Deploy
Stages
Monitor
Metrics
Logging
Protect
Throttling
API Gateway > APIs > some-api (hwuzd4iin3)
some-api
API details
API ID
hwuzd4iin3
Protocol
HTTP
Created
2024-05-13
Description
No Description
Default endpoint
Enabled
https://hwuzd4iin3.execute-api.eu-central-1.amazonaws.com
ARN
arn:aws:apigateway:eu-central-1::/apis/hwuzd4iin3
Stages for some-api (1)
Find resources
Stage name
Invoke URL
Attached deployment
Auto deploy
Last updated
$default
https://hwuzd4iin3.execute-api.eu-central-1.amazonaws.com
yecs2w
enabled
2024-05-13
Tags (0)
Find resources
Key
Value
No tags
No tags associated with the resource.
Manage tags

Spring Cloud Gateway

- Bean around for a while…remember Zuul?
- Built on Spring Webflux
- Routing
- Enriching (headers, body)
- Circuit breakers
- Filter chains
- etc.

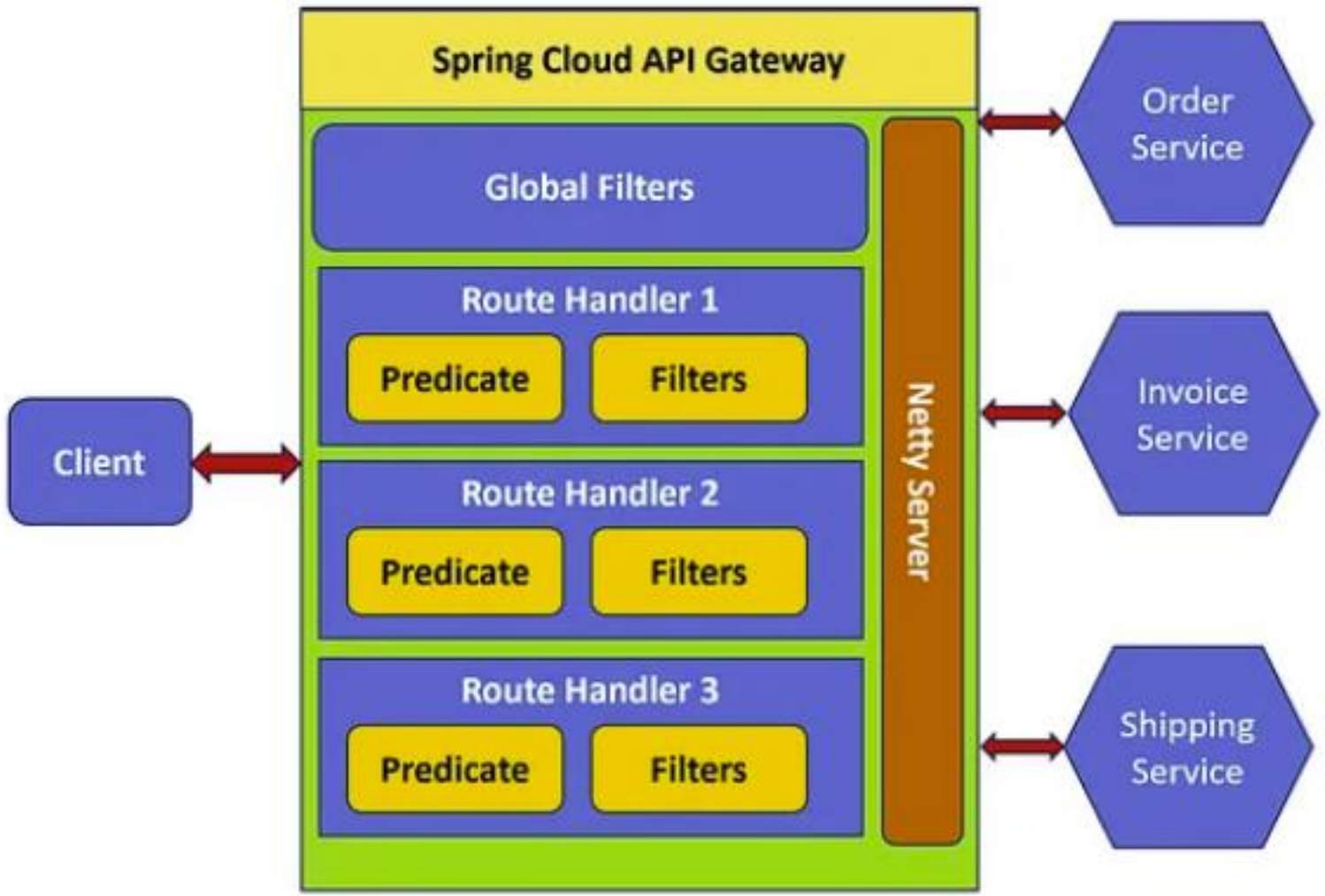
Pros and Cons of SCG

- Pros:
  - Spring Ecosystem Integration
  - Reactive programming support
  - Simple configuration
  - Easily moved between environments
  - Flexible & customizable
  - It’s on the Dev (not DevOps) field

Pros and Cons of SCG

- Cons:
  - If you're new to Spring Boot…
  - Or Reactive Programming…
  - No GUI

Spring Cloud Gateway



Declarative Configuration

- Simple
- Familiar syntax
- Does not require compilation

Declarative Configuration

```
spring:
  cloud:
    gateway:
      routes:
        - id: user-service
          uri: lb://user-service
          predicates:
            - Path=/users/**
        - id: order-service
          uri: lb://order-service
          predicates:
            - Path=/orders/**
```

Code Configuration

- Builder based syntax
- Compilation validates syntax
- Easier re-use
- Easier testing
- More readable?

Code Configuration

```
RouteLocatorBuilder.builder()
 .routes()
 .route("user-service", r -> r.path("/users/**")
 .uri("lb://user-service"))
 .route("order-service", r -> r.path("/orders/**")
 .uri("lb://order-service"))
.build();
```

Many Predicates

- By Cookie
- After/Before/Between Dates
- By Header
- By Host or Remote Address
- By Query Param

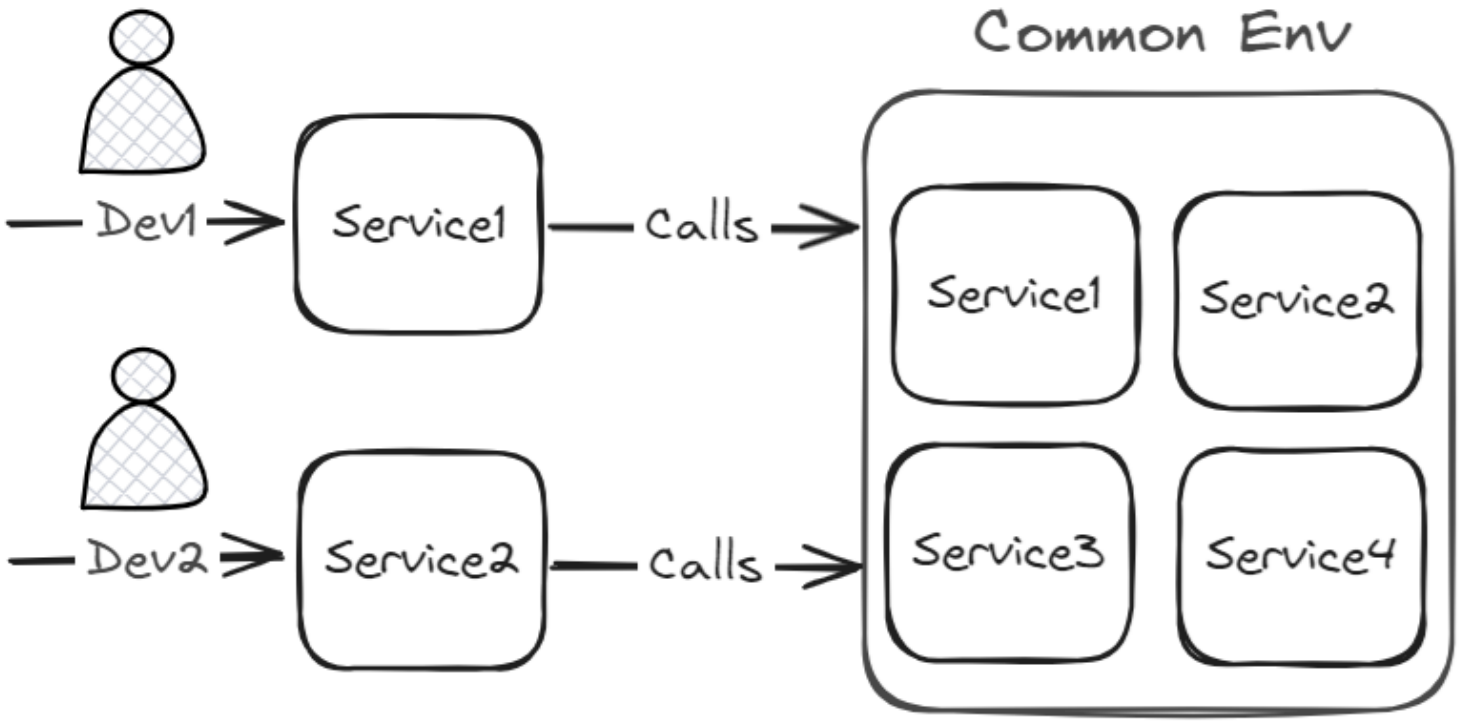
Global Filters

- Applied on all requests
- Can be several, ordered filters
- Can contain pre and post logic
- Use cases:
  - Logging
  - Global context enrichment
  - Common response pattern

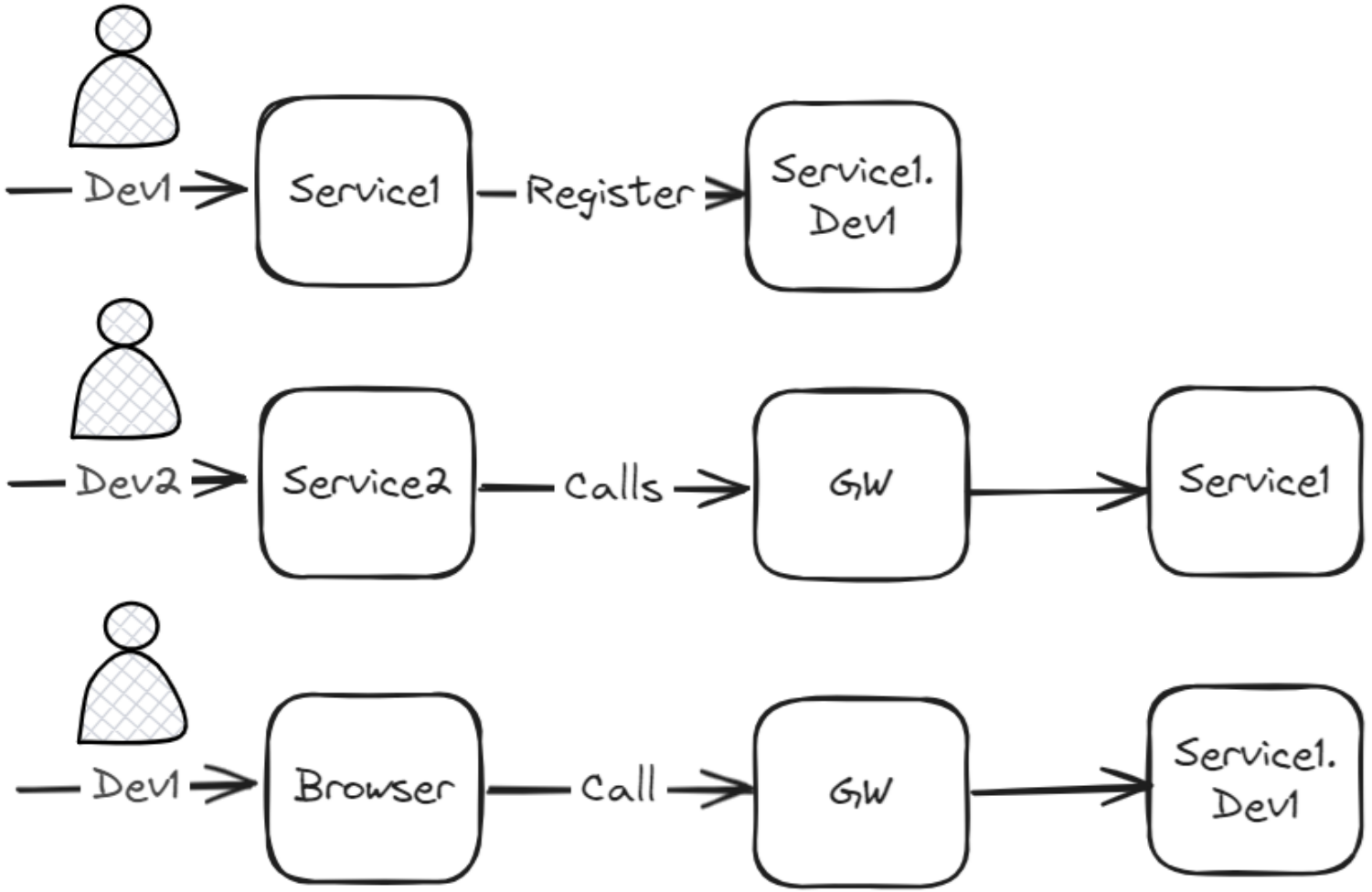
Demos!

- Let’s set up a service and a Gateway
- Some weird routings…
- Let’s write a pre filter
- Let’s write a post filter

The Problem


Common Env
Dev1
Service1
Calls
Service1
Service2
Dev2
Service2
Calls
Service3
Service4

Spring to the rescue



Takeaways

- API Gateway is a must, and it can do great things for you
- For simple needs – use a simple product
- When you need to customize or integrate with Spring Boot apps – Spring Gateway rules!
- Know and use your options!

Q&A & Thank you!

Too shy to ask here?

- dan.erez@gmail.com
- Feel free to consult!