O'Reilly®
Software Architecture

A Hitchhiker’s Guide to
Cloud Native API Gateways

oreillysacon.com
#OReillySACon
Mario-Leander Reimer
Principal Software Architect
QAware GmbH
Monolithic Vintage System

Users → Monolithic Vintage System

domain.example.com
Users

Monolithic Vintage System

service-a.default.example.com

Route

A Shared PasS

A Namespace

Service A

system.example.com

Monolithic Vintage System
A Hitchhiker's Guide to Cloud Native API Gateways

Users

system.example.com

service-a.default.example.com

service-b...

service-c...

Monolithic Vintage System

A Shared PasS

A Namespace

Service A

Service B

Service C
A Hitchhiker’s Guide to Cloud Native API Gateways

Monolithic
Vintage System

A Shared PasS

A Namespace

Service A’

Service B

Service C’

3rd Party Apps

Users

system.example.com

service-a.default.example.com

service-b...

service-c...

8 | #OREillySACCon Berlin 2019 | A Hitchhiker’s Guide to Cloud Native API Gateways | @LeanderReimer cloudnativenerd qaware
Users

Monolithic Vintage System

A Shared PasS

A Namespace

Service A'

Service B

Service C'

B Namespace

Service X

Service Y

Service Z'

3rd Party Apps

Unreliable Legacy Systems

system.example.com

service-a.default.example.com

service-b...

service-c...

3rd Party Apps

Users
Users

Monolithic Vintage System

System.example.com

Service A'

Route

Service B

Route

Service C'

Route

3rd Party Apps

User

Service-a.default.example.com

Service-B...

Service-C...

B Namespace

Service X

Service Y

Service Z'

B Shared PasS

A Namespace

Service A'

Route

Service B

Route

Service C'

Route

Unreliable Legacy Systems

Internal Systems

Monolithic Vintage System

3rd Party Apps

Users

Service-a.default.example.com

Service-B...

Service-C...

System.example.com

service-b...

service-c...
APIs are the center piece of any successful digital product. Proper management of your APIs right from the start is crucial, to not end up in API hell.

https://thenewstack.io/history-service-mesh/
API Gateways are like the Façade Pattern in Cloud Native Application Design and Microservice Architectures.
• **Traffic Management:** Path, Header, Host based Routing, Path Rewrite

• **Rollout and Deployment:** A/B Deployment, Canary Release, et.al.

• **QoS and Resiliency:** Circuit Breaker, Retry, Timeouts, Rate Limiting

• **Security:** AAA, Terminate TLS, Support for JWT and JWKS, Open ID, ...

• **Protocol Translation:** XML to JSON, gRPC to JSON, ...

• **Transformation:** Fan Out / Collect, Backend for Frontend, GraphQL, ...

• **Observability:** Integration into Logging, Monitoring, Tracing Stacks
API Gateways for North-South Communication

Service Meshes for East-West Communication
API Gateways in a Nutshell

• **Benefits**
  - Encapsulates internal structure of application
  - Provides client-specific APIs
  - BFF reduce the number of round trips
  - Simplifies client code

• **Drawbacks**
  - Yet another highly available component that needs to be managed and deployed
  - Risk of becoming a development bottleneck if managed centrally
  - Business logic in API gateway leads to accidental ESB
Possible Criteria for a Comparison

- **Open Source**: no initial $$$ and no vendor lock-in please
- **Maturity**: good and active community, little issues, frequent releases
- **Supported Features**: Traffic Management, Deployment, Security, Translation, Transformation, QoS, Resiliency, Observability
- **DevOps Friendly**: Easy setup and operability, supported platforms, CI/CD
- **Performance**: Small overhead, high throughput, super scalable
- **Observability**: good logging, monitoring, tracing capabilities + integration
Four Categories of API Gateways

A. Build Your Own API Gateway
B. API Management Solutions
C. Service Proxies
D. Cloud Native API Gateways
A. Build Your Own API Gateway

• Several frameworks available: Netflix Zuul 2, Spring Cloud Gateway, Node, Vert.x, Ballerina, Camel (K), Express Gateway, Sentinel

• Provides a lot of flexibility.

• But: your team needs to develop, maintain and operate the gateway!
B. API Management Solutions

- Several full-blown API Management solutions: Kong, Tyk, Mulesoft, 3scale, Apigee, ...

- Makes sense in Enterprise use cases. Generally cost $.

- Provide additional features such as payment, developer portals with API key management, integrations into enterprise infrastructure.

- Usually, centrally deployed and operated for the whole platform and all its applications.
C. Service Proxies

- Many choices: Nginx, OpenResty, Envoy, Traefik, Apache, HA Proxy, ...
- Lightweight and simple to use.
- Provided features vary a lot between products.
D. Cloud Native API Gateways

- Few choices: KrakenD, Ambassador, Gloo
- Usually build upon a service proxy such as Envoy and enhance it.
- Provide tight integration with cloud native platform, like K8s.
Demos

https://github.com/lreimer/hitchhikers-guide-api-gateways

https://speakerdeck.com/lreimer/a-hitchhikers-guide-to-cloud-native-api-gateways
42.

THE ANSWER TO LIFE, UNIVERSE, THE CLOUD AND API GATEWAYS.
Please rate today’s session

Session page on conference website

O’Reilly Events App