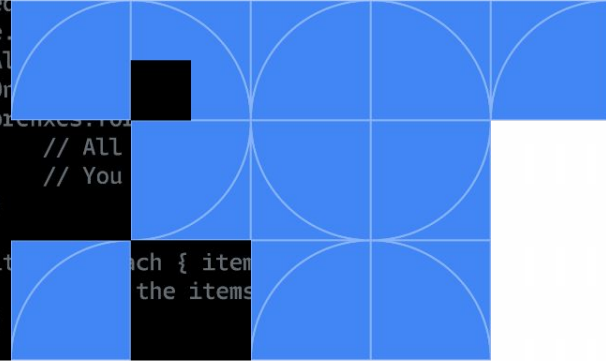
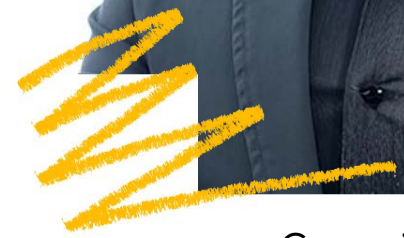


```
// You'll need
// com.google
listRef.listAll
.addOr
prefixes.fo
// ALL
// You
}
it
ach { item
the items
}
}
```



K8s Gateway API 101




Gregorio Palamà

Gregorio Palamà

DevOps & Cloud Engineer @ **Finwave**

Community Manager @ **GDG Pescara**



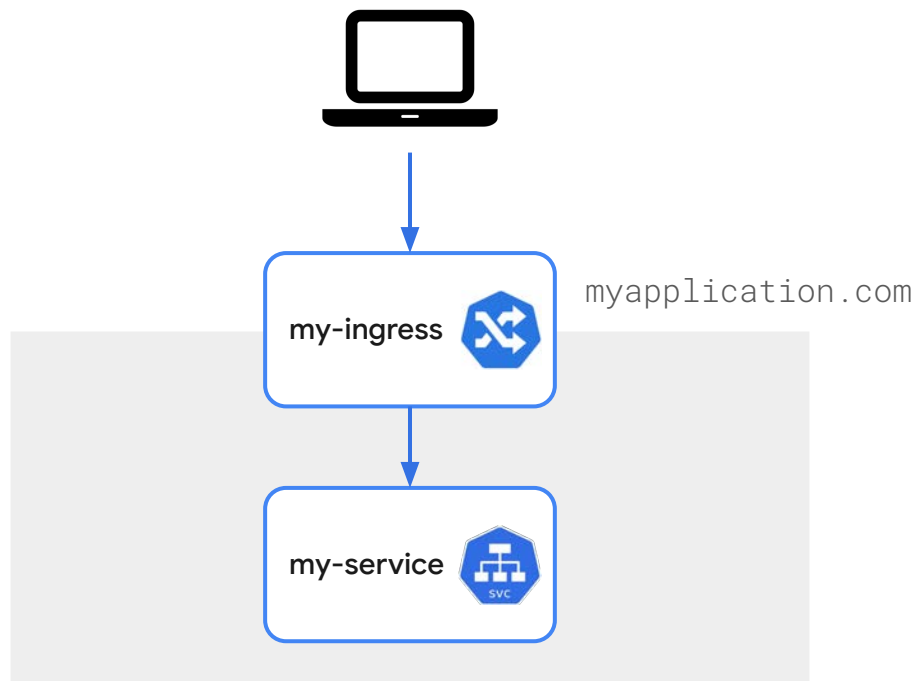


Exposing our Services outside the cluster

**Ingress API is
GA since K8s
1.19 and is
widely used**



Ingress (north/south)



Ingress example

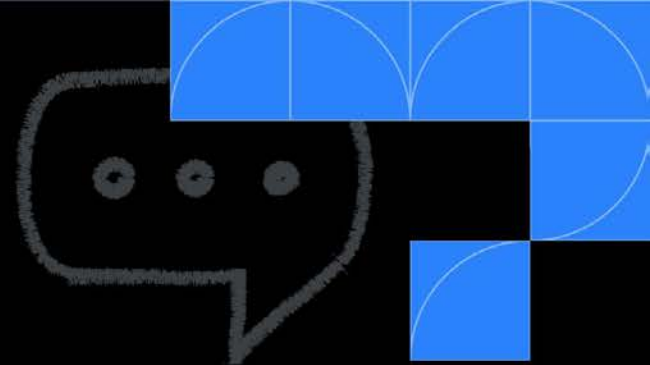
```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
spec:
  ingressClassName: nginx-example
  rules:
  - http:
    paths:
    - path: /my-path
      pathType: Prefix
      backend:
        service:
          name: my-service
          port:
            number: 80
```



```
Text('Simple Statement or URL',  
  style: TextStyle(  
    color: Colors.blue[200],  
  ),  
),  
),
```

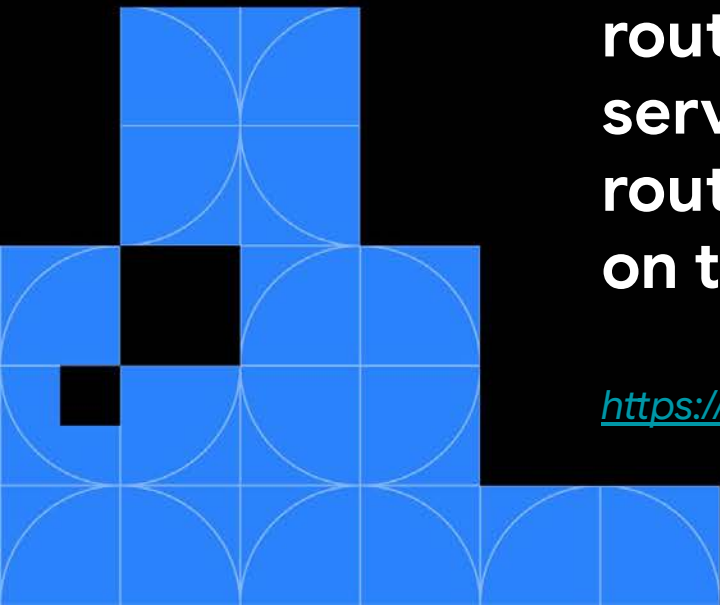
```
s.star,  
r: Colors.blue[500],
```

```
Text('23'),
```



Ingress exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. Traffic routing is controlled by rules defined on the Ingress resource.

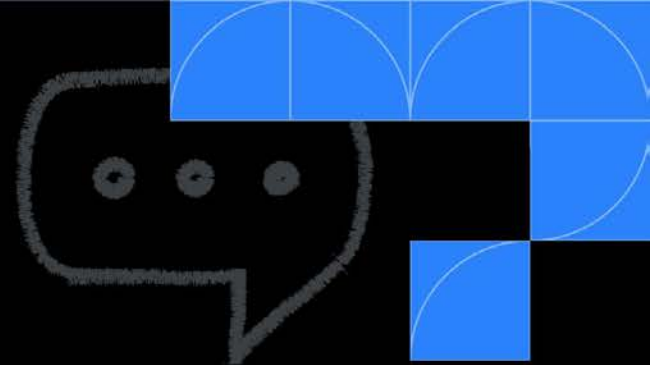
<https://kubernetes.io/docs/concepts/services-networking/ingress/>



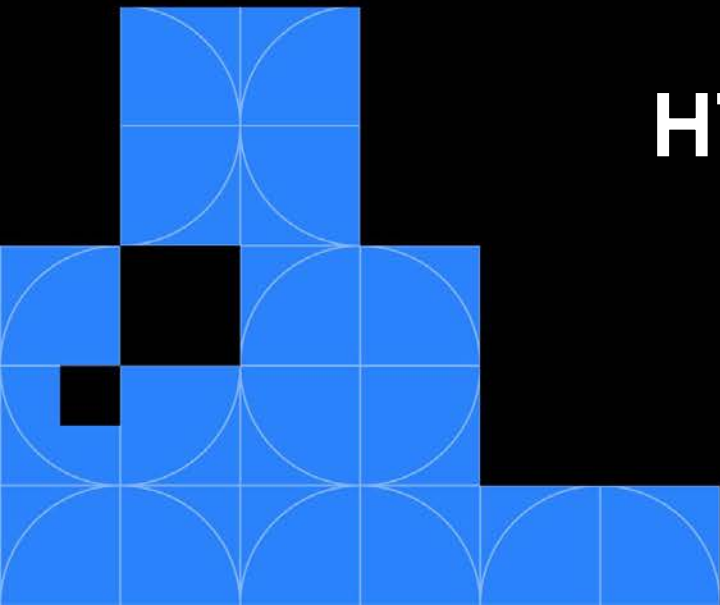
```
Text('Simple Statement or URL',  
  style: TextStyle(  
    color: Colors.blue[200],  
  ),  
),  
),
```

```
s.star,  
r: Colors.blue[500],
```

```
Text('23'),
```



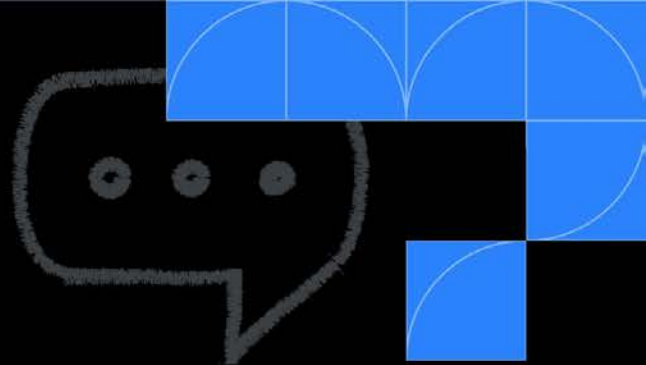
HTTP and HTTPS, no other protocols



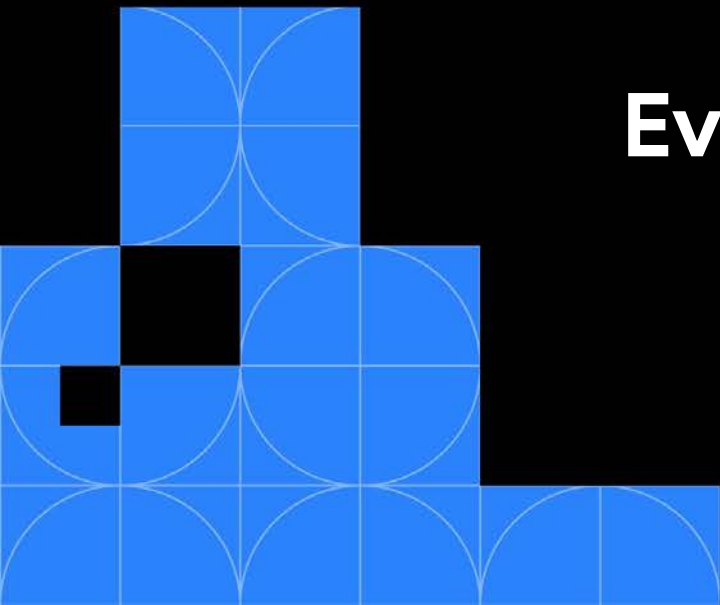

```
Text('Simple Statement or URL',  
  style: TextStyle(  
    color: Colors.blue[200],  
  ),  
),  
),
```

```
  s.star,  
  r: Colors.blue[500],
```

```
Text('23'),
```



**Everything is defined in the
same resource**



Ingress Personas



User
Ingress owner

Ingress Personas

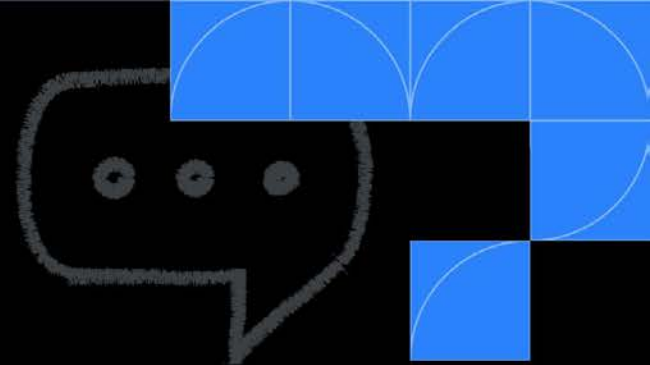
User
Ingress owner

- Infrastructure configuration
- TLS configuration
- Routing configuration

```
Text('Simple Statement or URL',  
  style: TextStyle(  
    color: Colors.blue[200],  
  ),  
),  
),
```

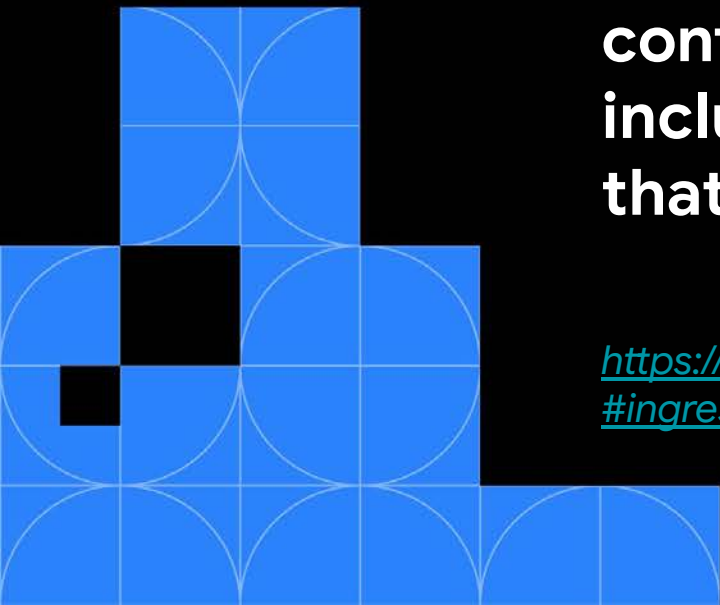
```
s.star,  
r: Colors.blue[500],
```

```
Text('23'),
```



IngressClass are resources that contain additional configuration including the name of the controller that should implement the class.

<https://kubernetes.io/docs/concepts/services-networking/ingress/#ingress-class>



Ingress Personas

User

→ Routing configuration

Infrastructure Provider
Cluster Operator

→ Infrastructure configuration

→ TLS configuration

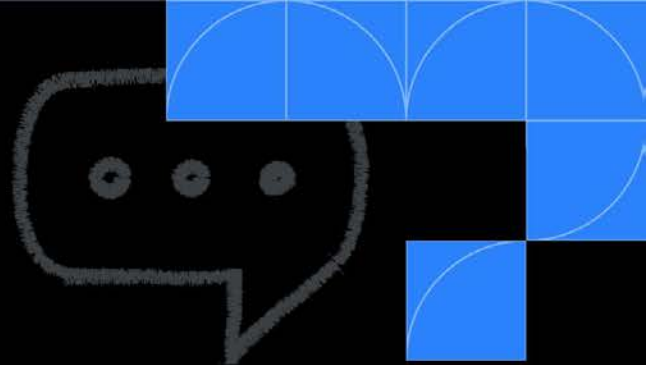


**Extensibility
relies on
annotations**

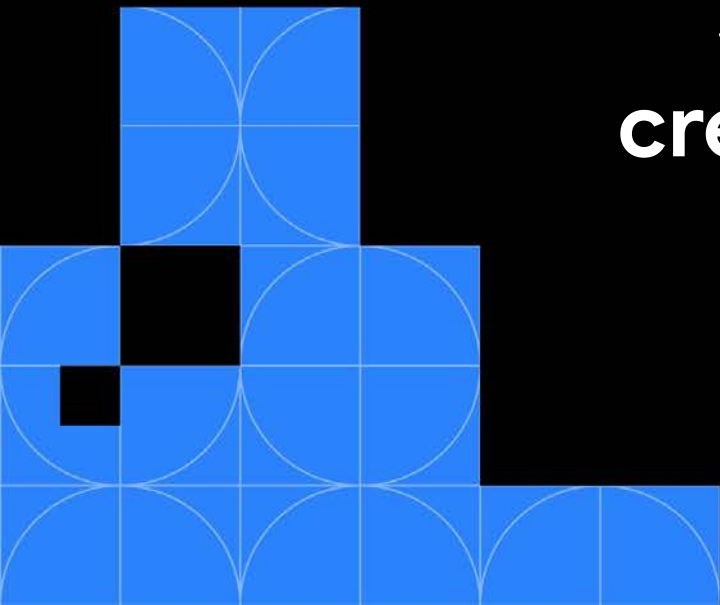
```
Text('Simple Statement or URL',  
  style: TextStyle(  
    color: Colors.blue[200],  
  ),  
),  
),
```

```
s.star,  
r: Colors.blue[500],
```

```
Text('23'),
```



**Some implementations
created custom extensions,
valid for the single
implementation only**




```
Text(
  'Section Title',
  style: TextStyle(
    color: Colors.green[200],
  ),
),
),
```

```
s.star,
r: Colors.green[500],
```

```
Text('23'),
```

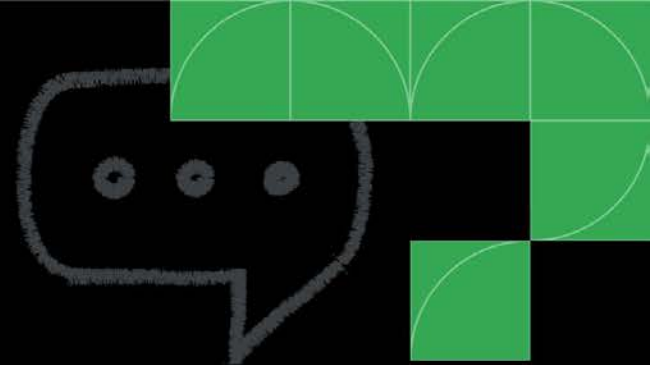


Gateway API



**GA since K8s
1.25 (October
31, 2023)**

```
Text('Simple Statement or URL',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
,  
,  
s.star,  
r: Colors.green[500],  
Text('23'),
```



API Gateway describes anything that exposes capabilities of a backend service, while providing extra capabilities for traffic routing and manipulation and sometimes more advanced features.

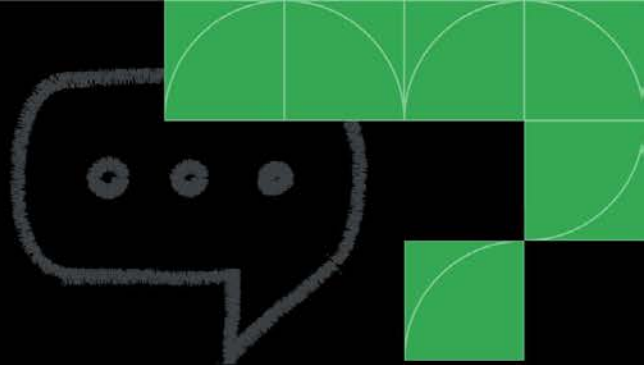
<https://gateway-api.sigs.k8s.io/#whats-the-difference-between-gateway-api-and-an-api-gateway>



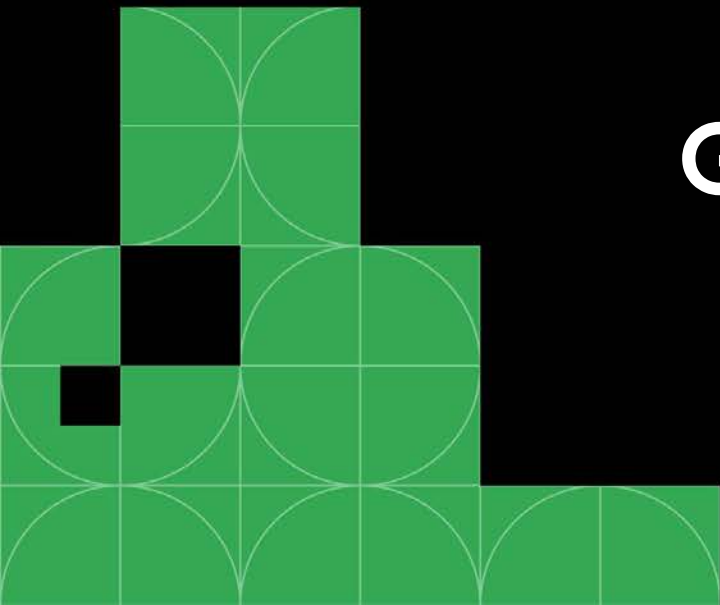

```
Text('Simple Statement or URL',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
),  
),
```

```
s.star,  
r: Colors.green[500],
```

```
Text('23'),
```



Gateway API Personas



Gateway API Personas



Ian

Infrastructure Provider



Chihiro

Cluster Operator



Ana

Application Developer

Gateway API Personas

Ian

Infrastructure Provider

- Infrastructure configuration
- Can work on multiple clusters

Chihiro

Cluster Operator

- Entry points configuration
- TLS configuration

Ana

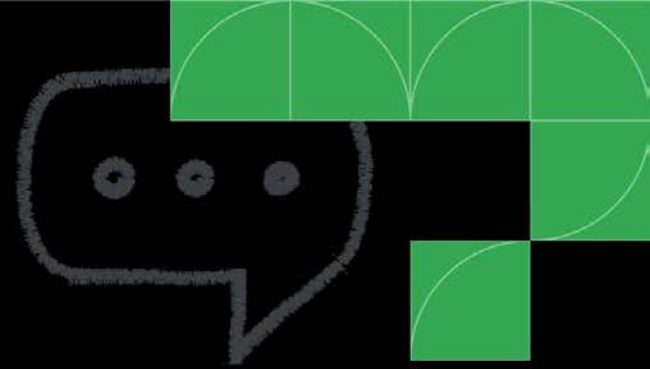
Application Developer

- Routing configuration

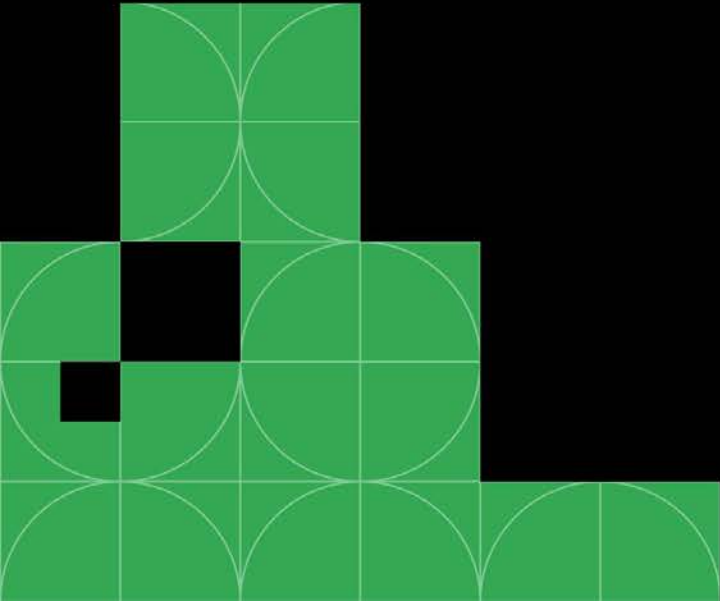

```
Text('Simple Statement or URL',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
),  
),
```

```
s.star,  
r: Colors.green[500],
```

```
Text('23'),
```



Resources



GatewayClass

- Defines a set of Gateways that share a common configuration and behaviour
- Each GatewayClass will be handled by a single controller
- Cluster-scoped resource
- Similar to IngressClass for Ingress

Gateway

- Describes how traffic can be translated to Services within the cluster
- Defines a request for a specific load balancer config that implements the GatewayClass' configuration and behaviour contract
- May be attached to one or more Route references
- It is possible to limit the Routes that can attach to a Gateway

Routes

- Define protocol-specific rules for mapping requests from a Gateway to Kubernetes Services
- In v1alpha2 are included 4 Route resources types
- Can attach to one or more Gateways
- Filters and advanced rules can be applied

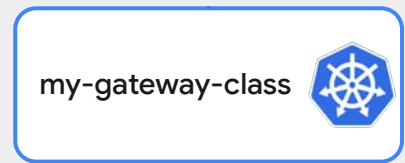
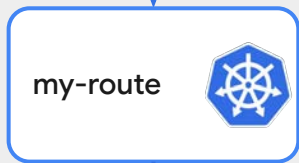
Routes

- HTTPRoute
- TLSRoute
- TCPRoute / UDPRoute
- GRPCRoute

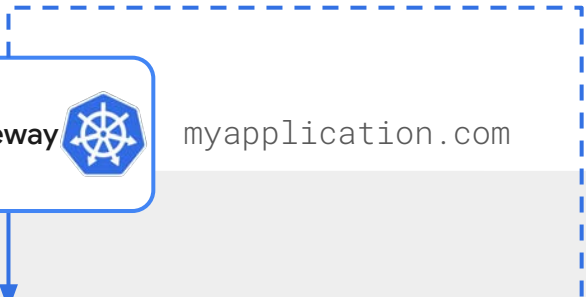
Gateway API (north/south)



myapplication.com



implements



Gateway API example

```
apiVersion: gateway.networking.k8s.io/v1beta1
kind: Gateway
metadata:
  name: my-gateway
  namespace: gateway-api-ns1
spec:
  gatewayClassName: my-gateway-class
  listeners:
  - name: myapplication
    port: 80
    protocol: HTTP
    allowedRoutes:
      kinds:
      - kind: HTTPRoute
      namespaces:
        from: Selector
        selector:
          matchLabels:
            kubernetes.io/metadata.name: gateway-api-ns2
```


Gateway API example

```
apiVersion: gateway.networking.k8s.io/v1beta1
kind: HTTPRoute
metadata:
  name: my-route
  namespace: gateway-api-ns2
spec:
  parentRefs:
  - kind: Gateway
    name: my-gateway
    namespace: gateway-api-ns1
  rules:
  - backendRefs:
    - name: my-service
      port: 80
```

Gateway API example

```
apiVersion: gateway.networking.k8s.io/v1beta1
kind: HTTPRoute
metadata:
  name: my-route
  namespace: gateway-api-ns2
spec:
  parentRefs:
  - kind: Gateway
    name: my-gateway
    namespace: gateway-api-ns1
  rules:
  - backendRefs:
    - name: my-service
      port: 80
      weight: 90
    - name: my-second-service
      port: 80
      weight: 10
```

Gateway API example


```
apiVersion: gateway.networking.k8s.io/v1beta1
kind: HTTPRoute
metadata:
  name: my-route
  namespace: gateway-api-ns2
spec:
  parentRefs:
  - kind: Gateway
    name: my-gateway
    namespace: gateway-api-ns1
  rules:
  - matches:
    - headers:
      - type: Exact
        name: conference
        value: conf42
      path:
        type: PathPrefix
        value: /some/thing
      method: GET
    backendRefs:
    - name: my-service
      port: 80
```

Gateway API example

```
apiVersion: gateway.networking.k8s.io/v1beta1
kind: HTTPRoute
metadata:
  name: my-route
  namespace: gateway-api-ns2
spec:
  parentRefs:
  - kind: Gateway
    name: my-gateway
    namespace: gateway-api-ns1
  rules:
  - filters:
    - type: RequestHeaderModifier
      requestHeaderModifier:
        add:
        - name: my-header
          value: conf42
    backendRefs:
    - name: my-service
      port: 80
```



LIVE DEMO

The background of the slide is a vibrant, futuristic cityscape at sunset. The sky is a mix of orange, pink, and purple, with soft clouds. The city buildings are illuminated with various colors, including blue, purple, and pink, creating a high-tech, cyberpunk aesthetic. A large, semi-transparent black text box is positioned on the left side of the image, containing the title. In the bottom left corner, there is a decorative pattern of green squares, some of which contain white circular or semi-circular shapes.

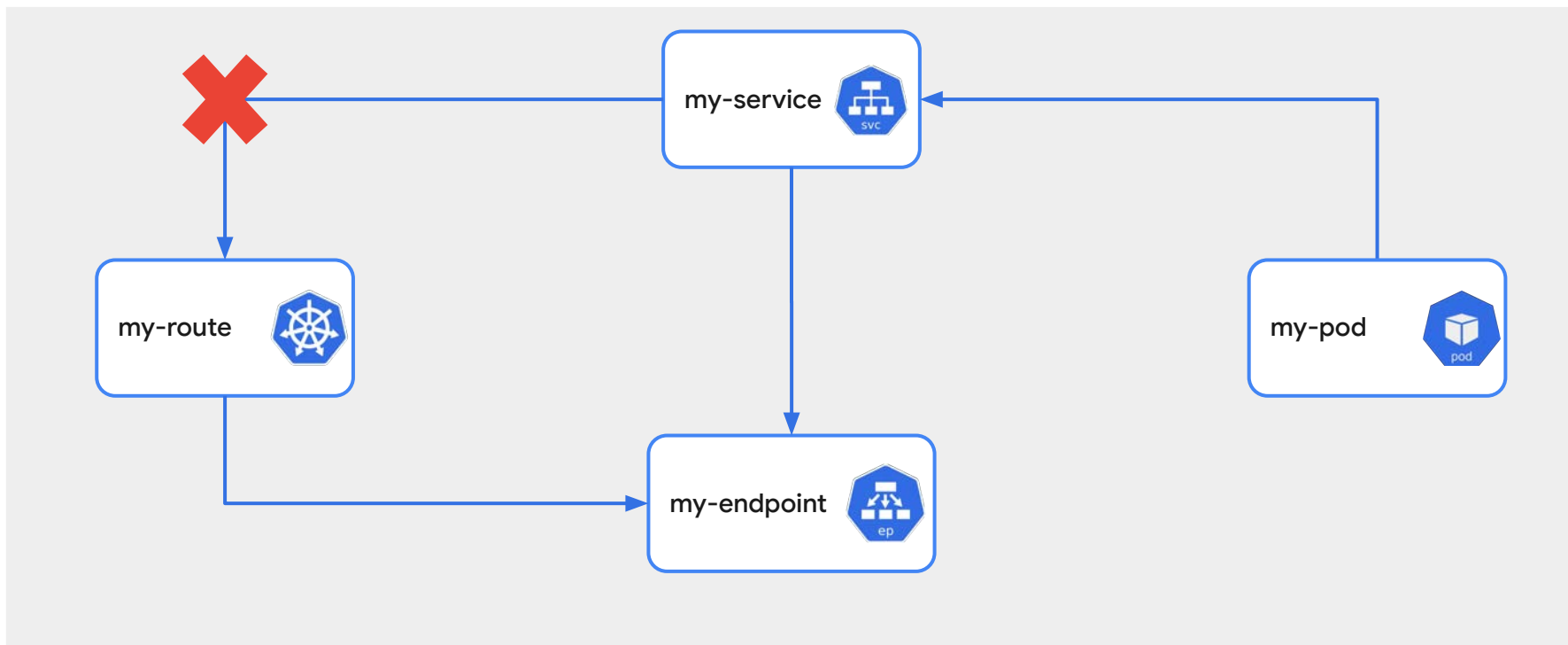
Gateway API for Mesh Management and Administration (GAMMA) Initiative

GAMMA introduced changes

- A Route can be associated directly with a Service
- A Service has 2 facets:
 - ◆ A **frontend**, combination of the ClusterIP and the DNS
 - ◆ A **backend**, the collection of endpoint IPs

Gateway API GAMMA (east/west)

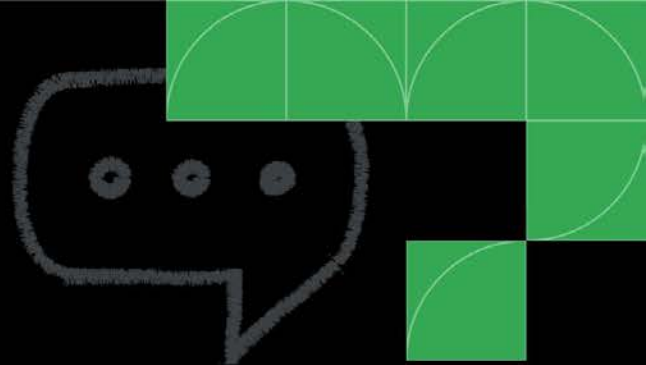
3



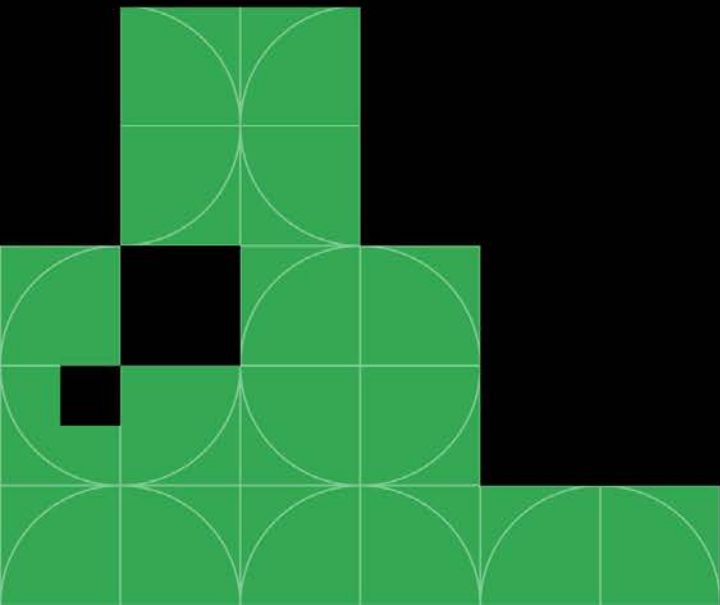

```
Text('Simple Statement or URL',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
),  
),
```

```
s.star,  
r: Colors.green[500],
```

```
Text('23'),
```



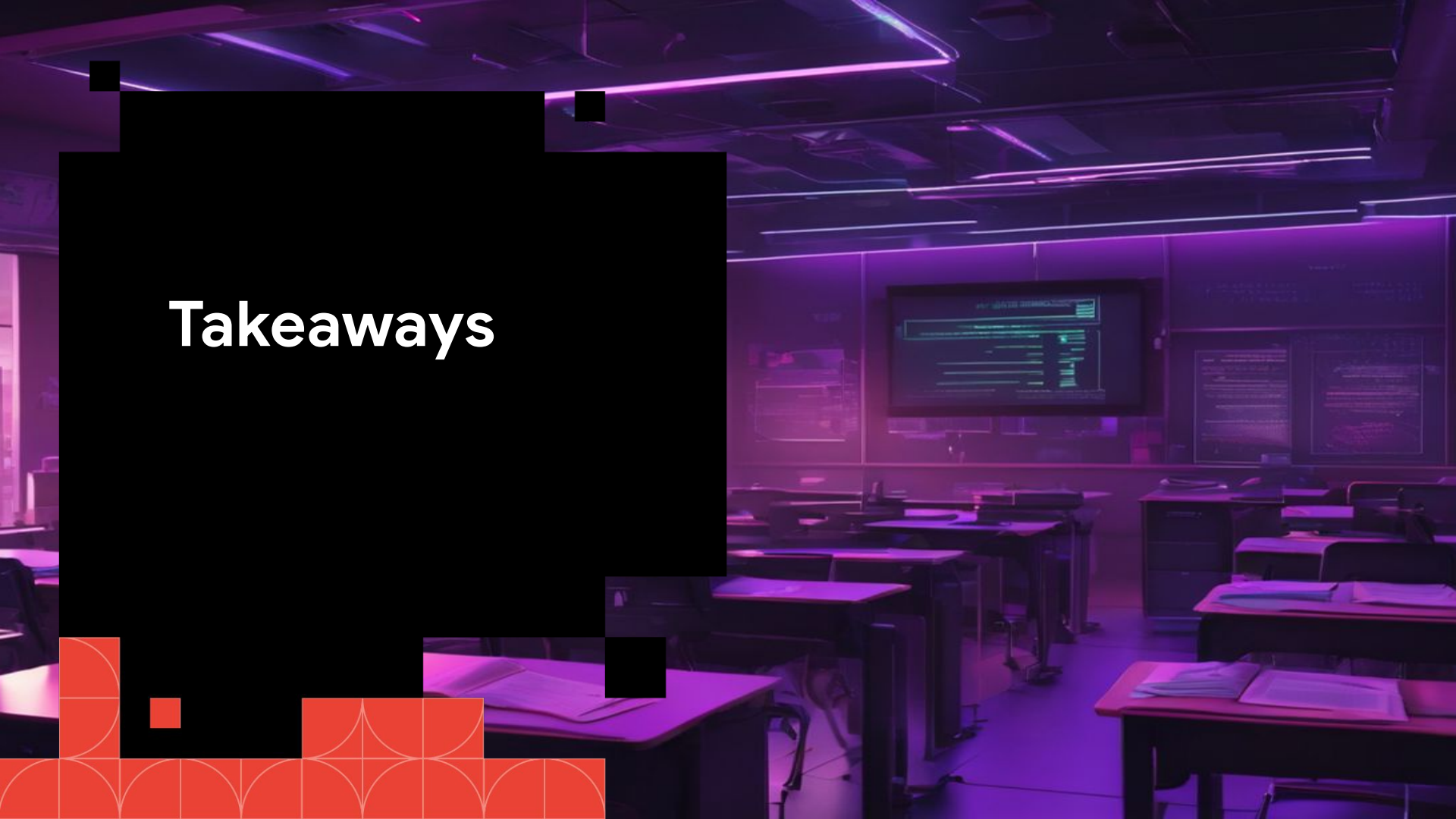
Implementations



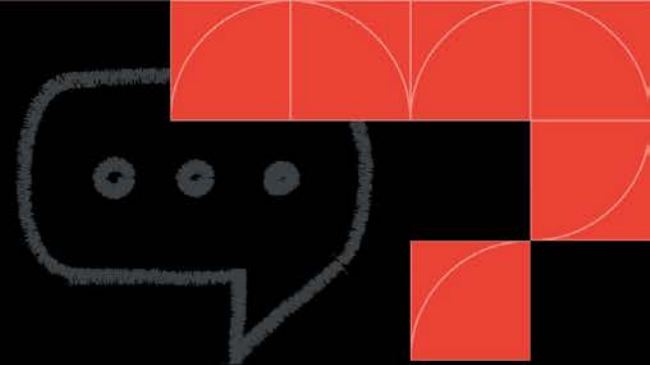
Implementations

- Google Kubernetes Engine (GA)
- Kong (GA)
- Apache APISIX (beta)
- Istio (beta)
- ...and more: <https://gateway-api.sigs.k8s.io/implementations/>

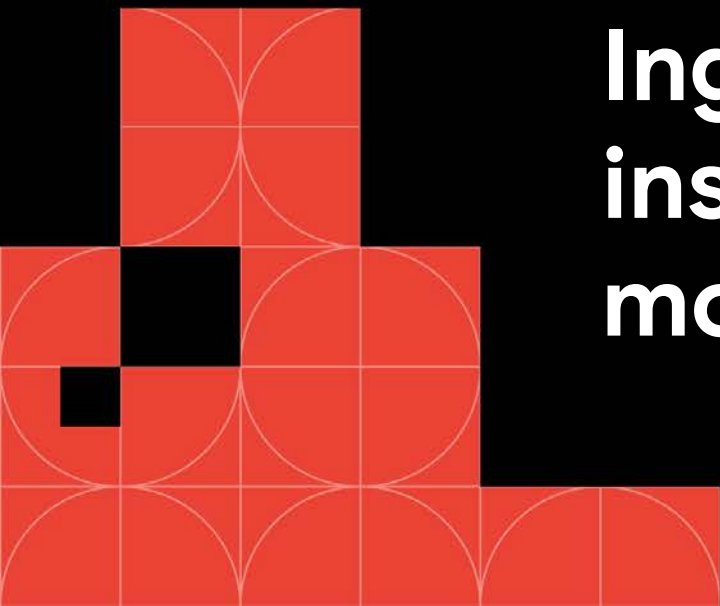
Takeaways



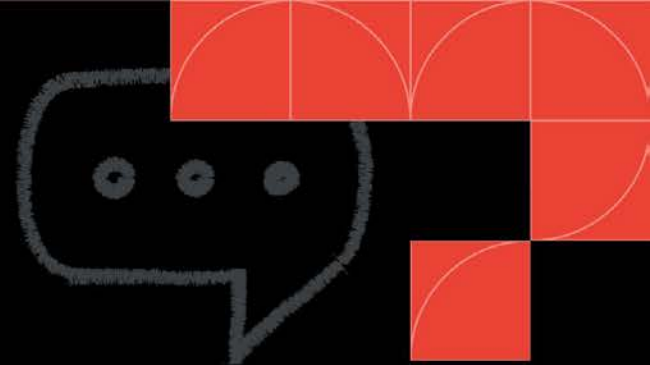

```
Text('Simple Statement or URL',  
  style: TextStyle(  
    color: Colors.red[200],  
  ),  
,  
,  
s.star,  
r: Colors.red[500],  
Text('23'),
```



Ingress API has an insufficient permission model




```
Text('Simple Statement or URL',  
  style: TextStyle(  
    color: Colors.red[200],  
  ),  
,  
,  
s.star,  
r: Colors.red[500],  
Text('23'),
```



**Gateway API has different
good implementations, and
all of them adhere to the
standard**



Useful links

- <https://gateway-api.sigs.k8s.io/>
- <https://github.com/kubernetes-sigs/gateway-api>
- <https://kubernetes.io/docs/concepts/services-networking/ingress/>
- <https://www.youtube.com/@GoogleOpenSource/search?query=gateway%20api>

Feedbacks

