KubeCon | CloudNativeCon

Europe 2023

TV 2 Danmark

KubeCon | CloudNativeCon

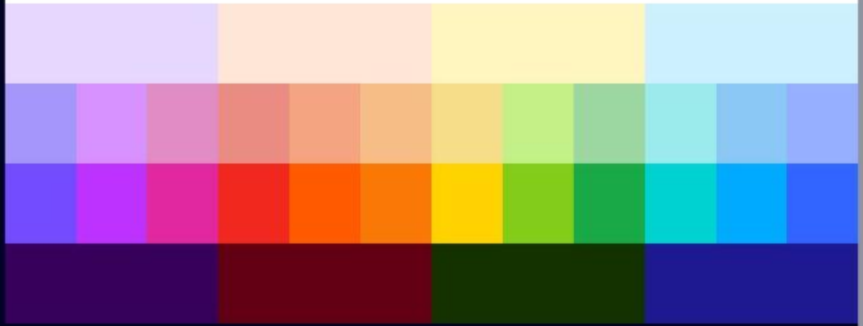
Europe 2023

Beyond Gateway API:

Building a Cloud Agnostic Gateway Controller for Self-Service Network Configuration

Michael Vittrup Larsen

Martin Villumsen



21:30

Danmark

a

g

	Play
News	.dk
Sport	
Sport X	Nyheder
Echo	Sport
Charlie	Vejr
Fri	

Digital Products & Experience

+200
employees

+20
teams

+10
products

TV2 Danmark

Photo by Annie Spratt on Unsplash

Our Journey

Developers have been building their own Kubernetes platforms

- Exposing applications using traditional Ingress
- Istio for more advanced networking
- Increased need for network features external to Kubernetes

We are building a multi-tenant Kubernetes platform

- Reduce developer cognitive load
- Provide a paved path for running apps in the cloud
- Kubernetes API for everything (no Terraform)
- GitOps enable self-service

Our Journey

Challenge: Glueing Things Together

Different Solutions

1. Custom scripts and templates
2. Cartographer (templated supply chains)
3. Custom controller built with MetaController
4. Our own full-blown controller built with KubeBuilder
   - Focus on network configuration implementing Gateway API

References

https://cartographer.sh/
https://github.com/metacontroller/metacontroller
https://github.com/kubernetes-sigs/kubebuilder

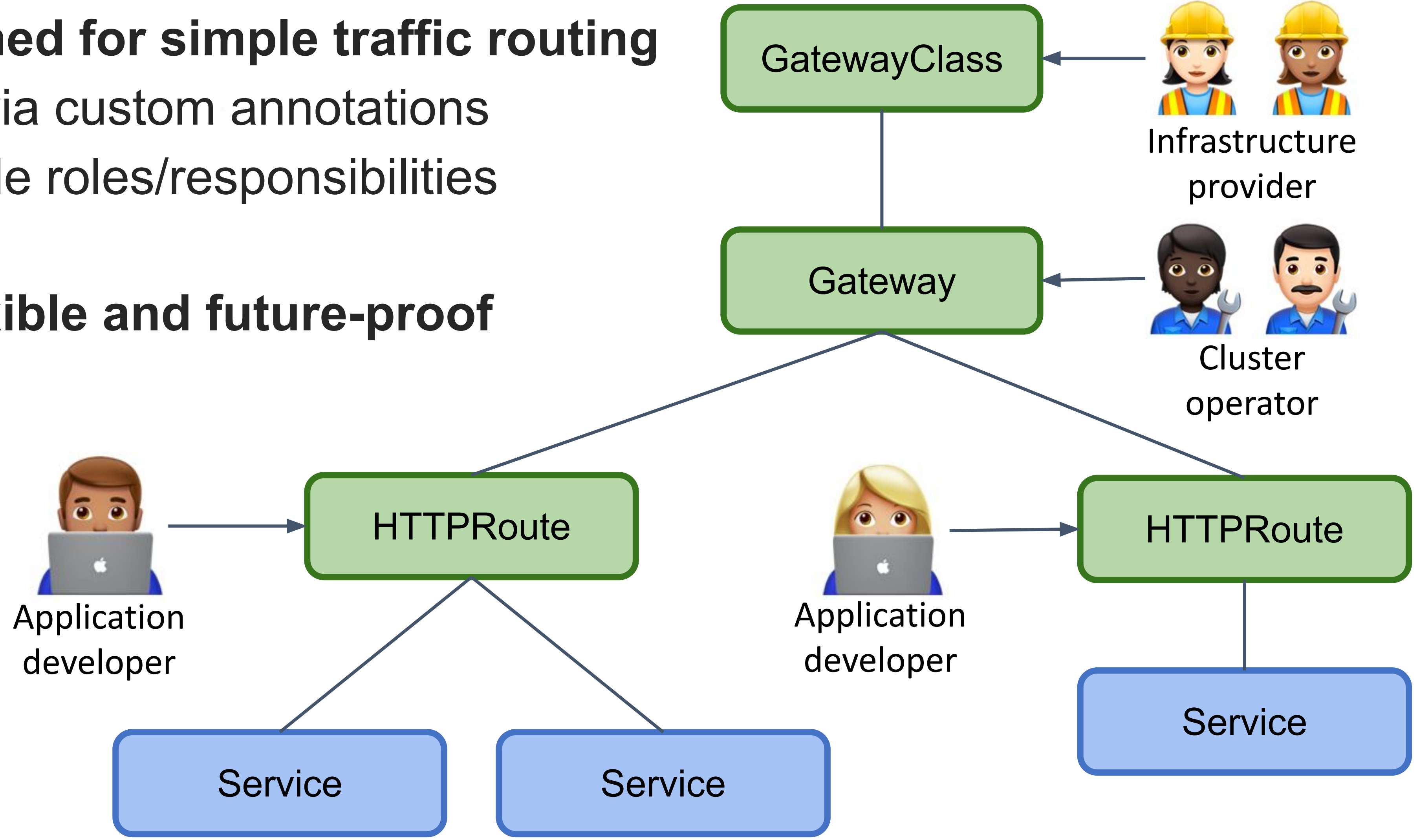
Why Gateway API?

Ingress was designed for simple traffic routing

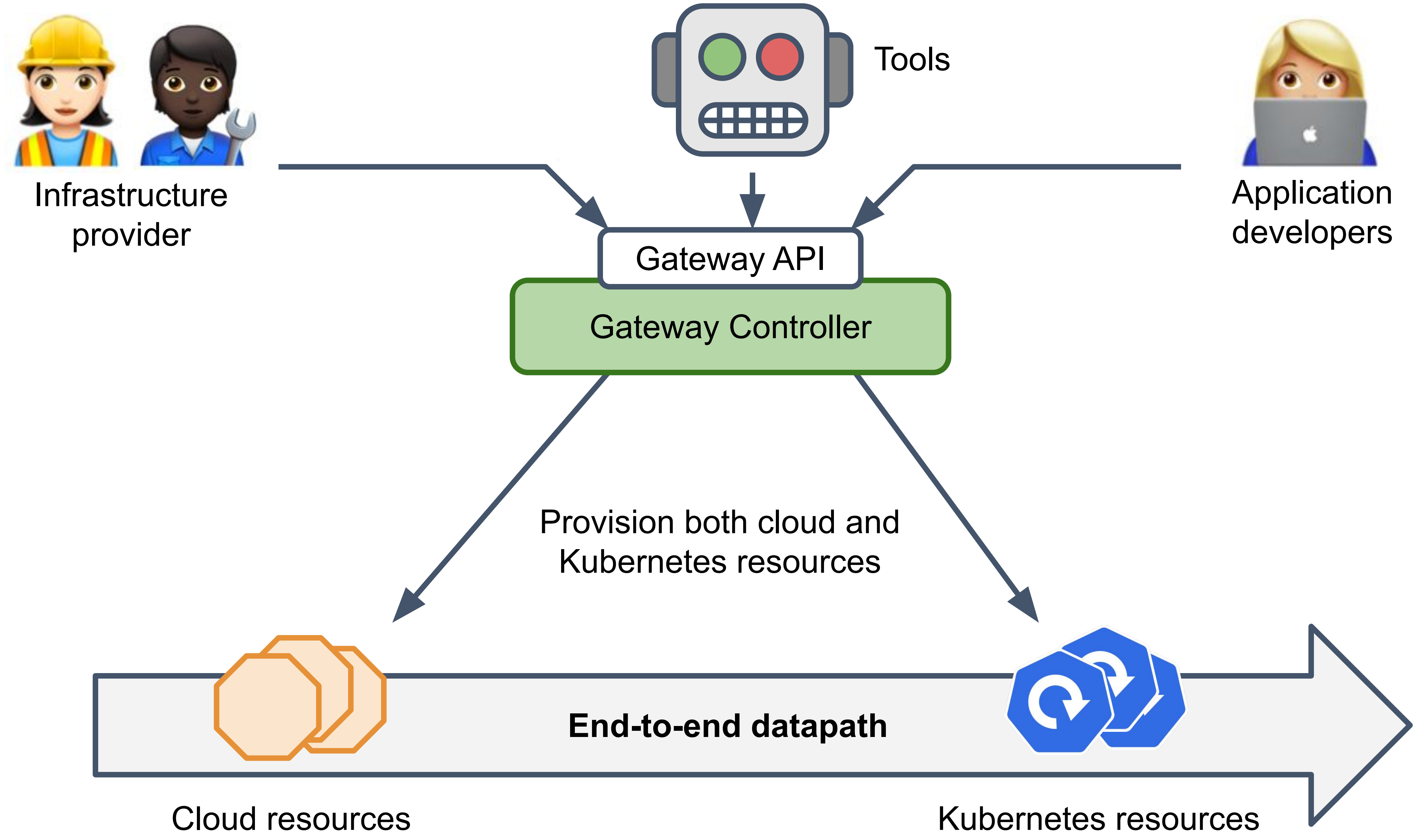
- Only extensible via custom annotations
- Combines multiple roles/responsibilities

Gateway API is flexible and future-proof

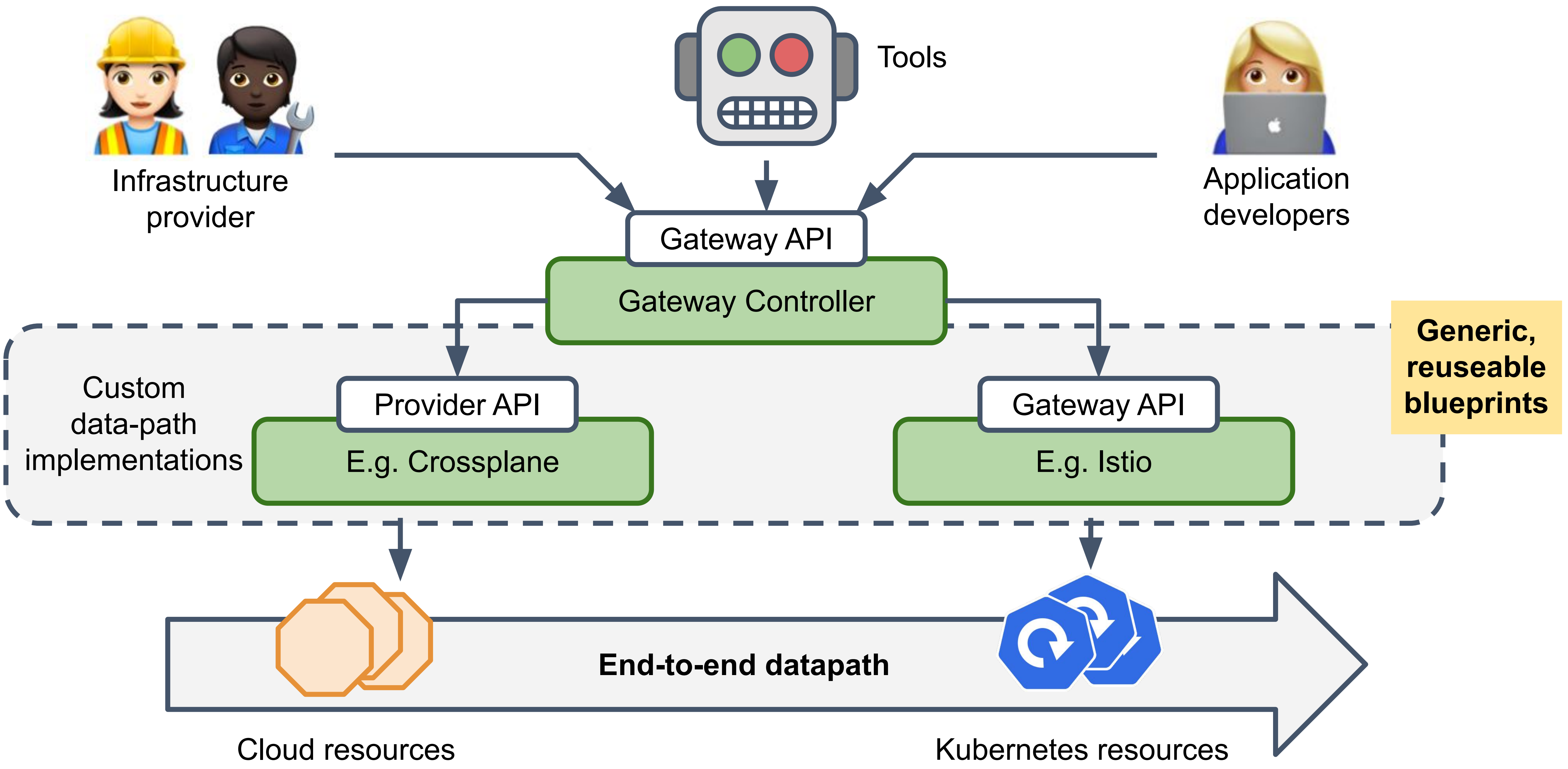
- Role-oriented
- Expressive
- Extensible



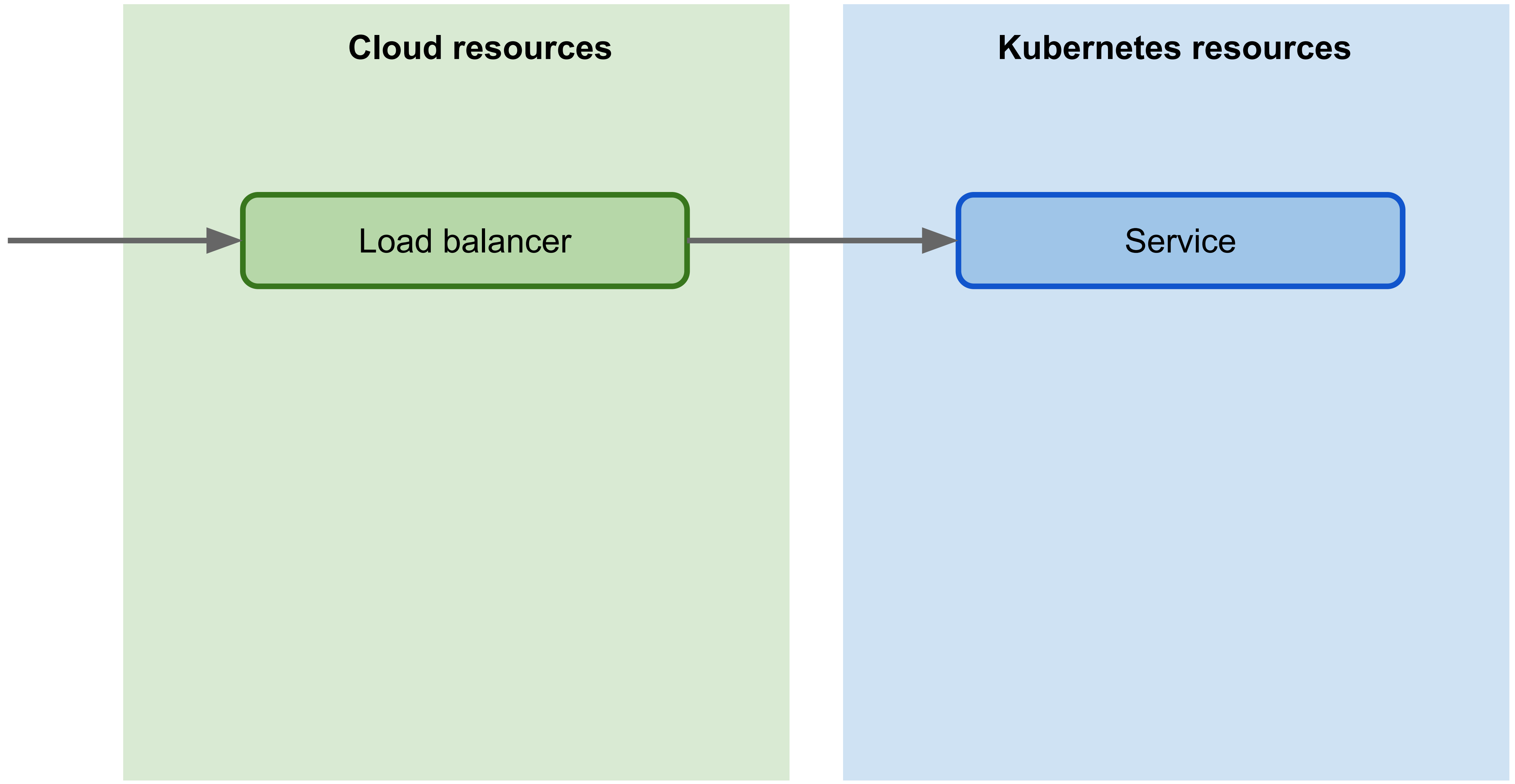
End-to-end Data Path (1)



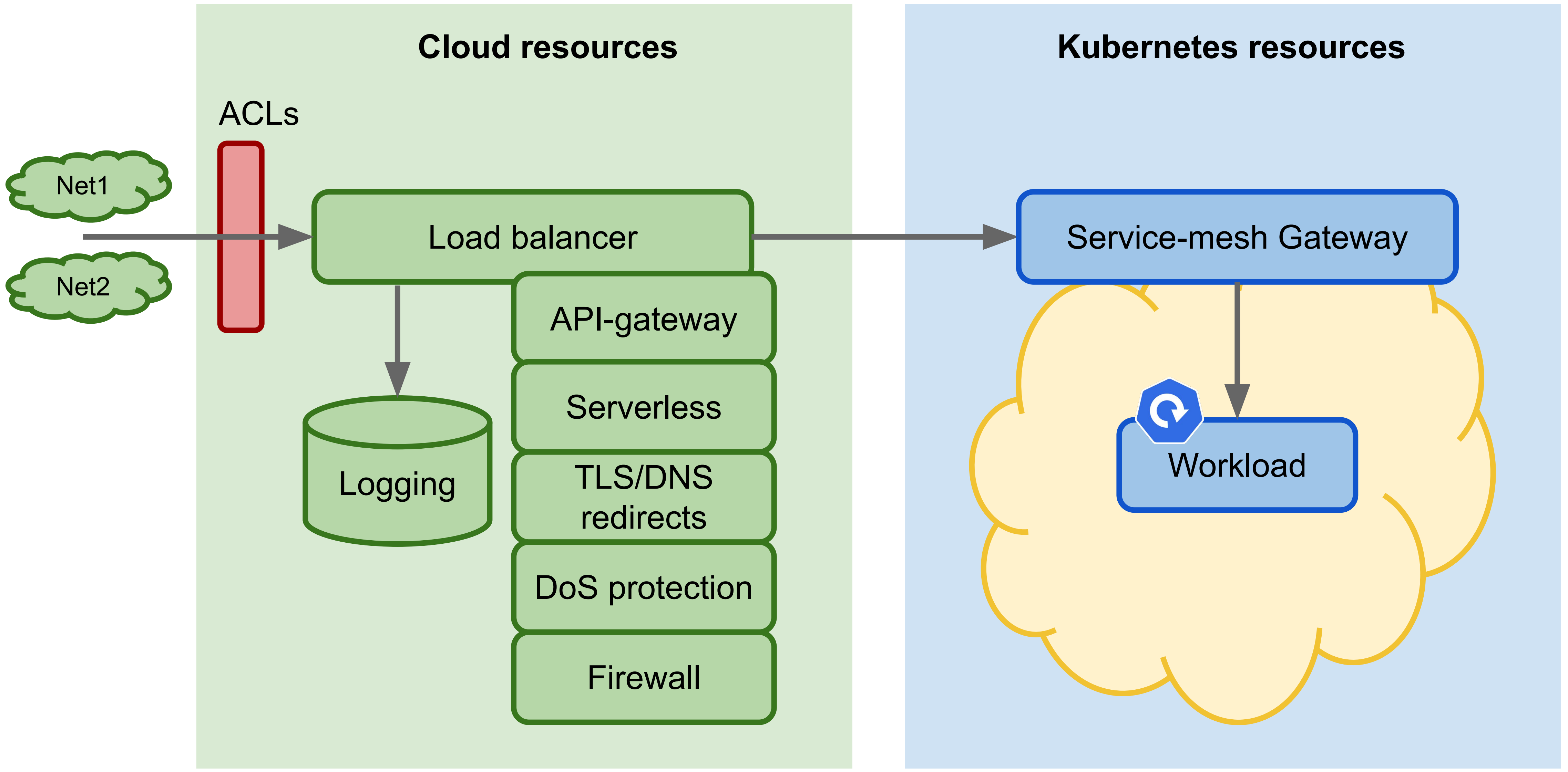
End-to-end Data Path (2)



Different Data Paths (1)



Different Data Paths (2)

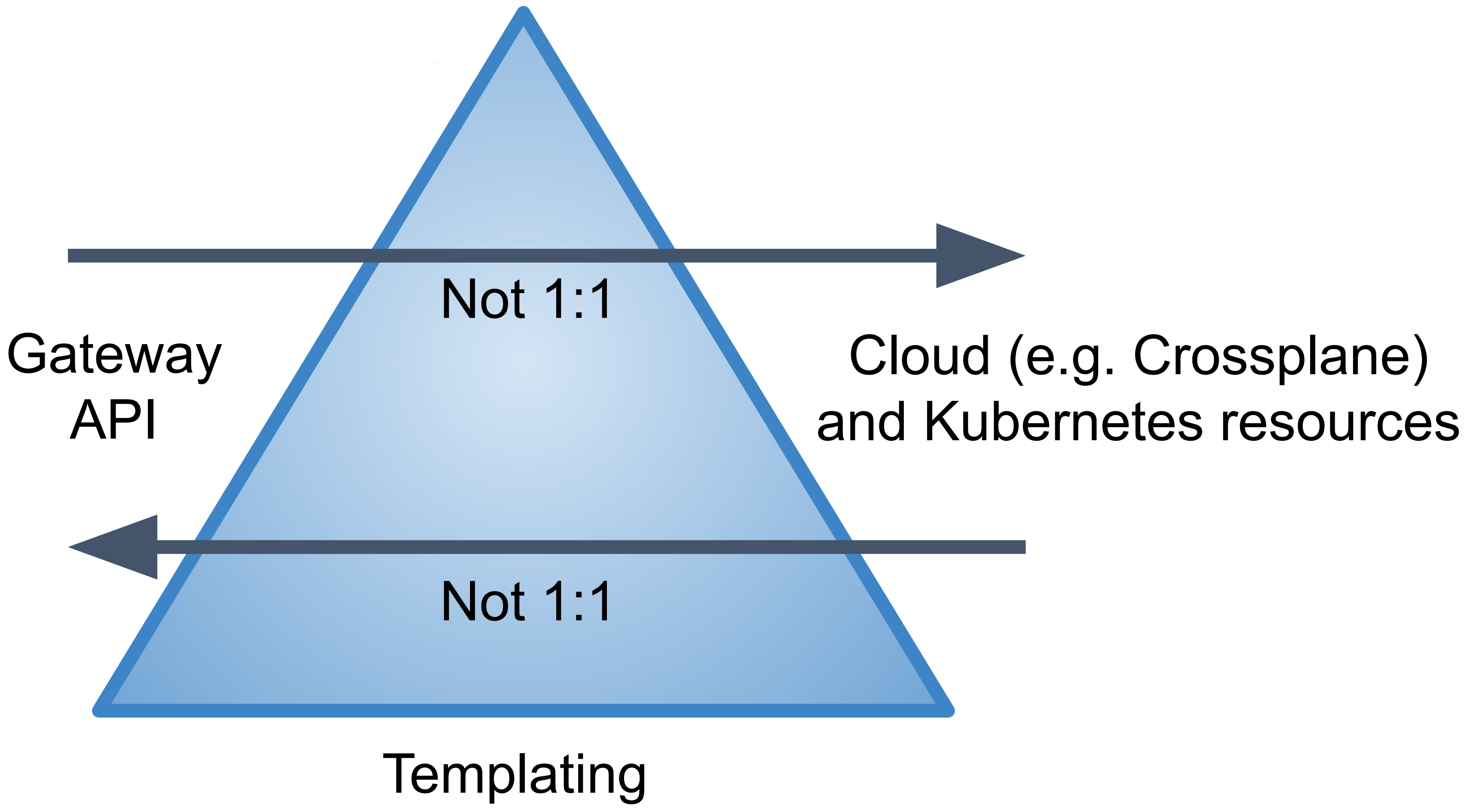


Key Design Principle

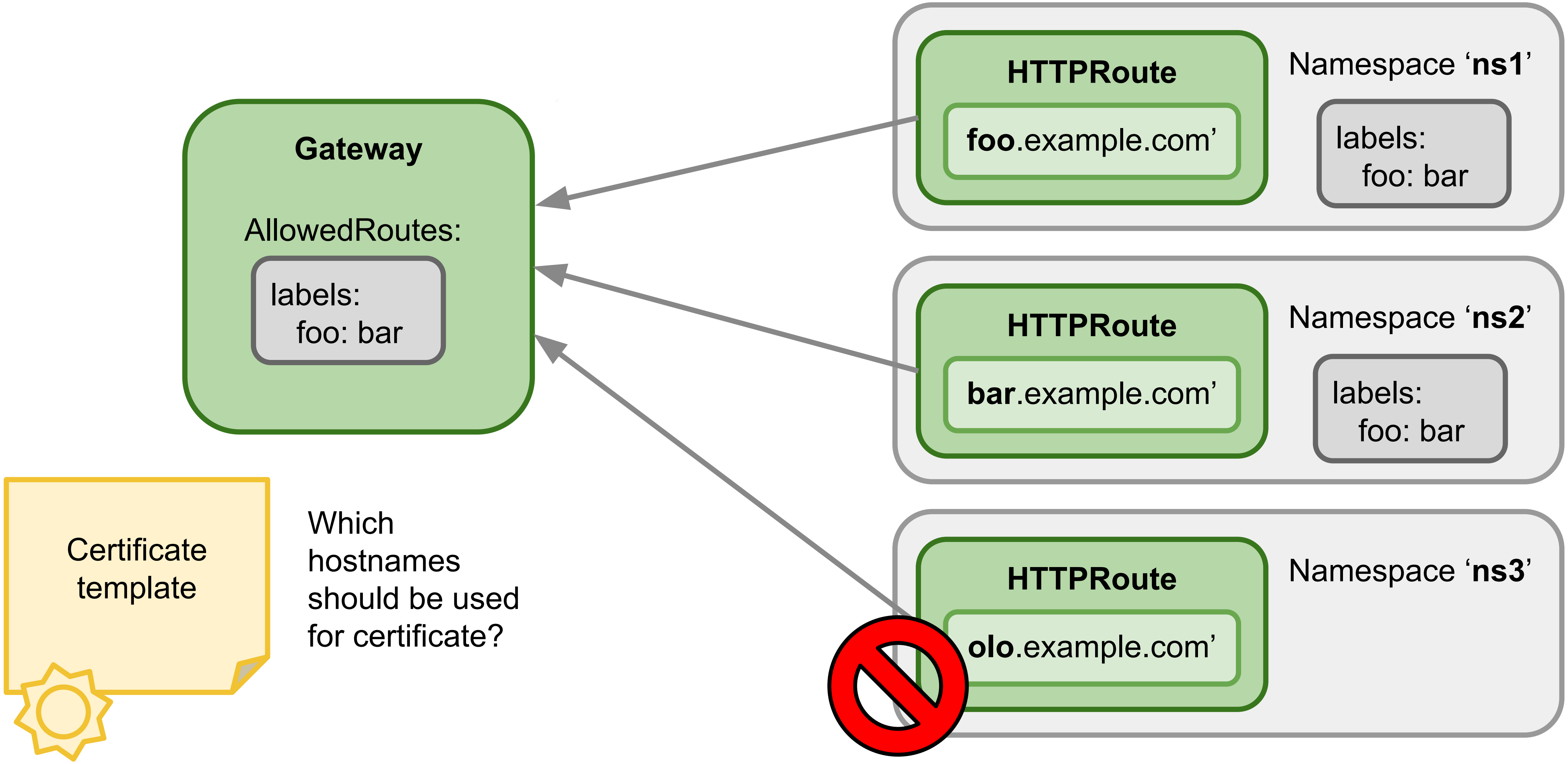
Blueprints **based on templating**, not code

- Evolution without code changes
- Foundation for **generic, shareable blueprints**
- Gateway-controller becomes **cloud agnostic**

Challenge with Templating

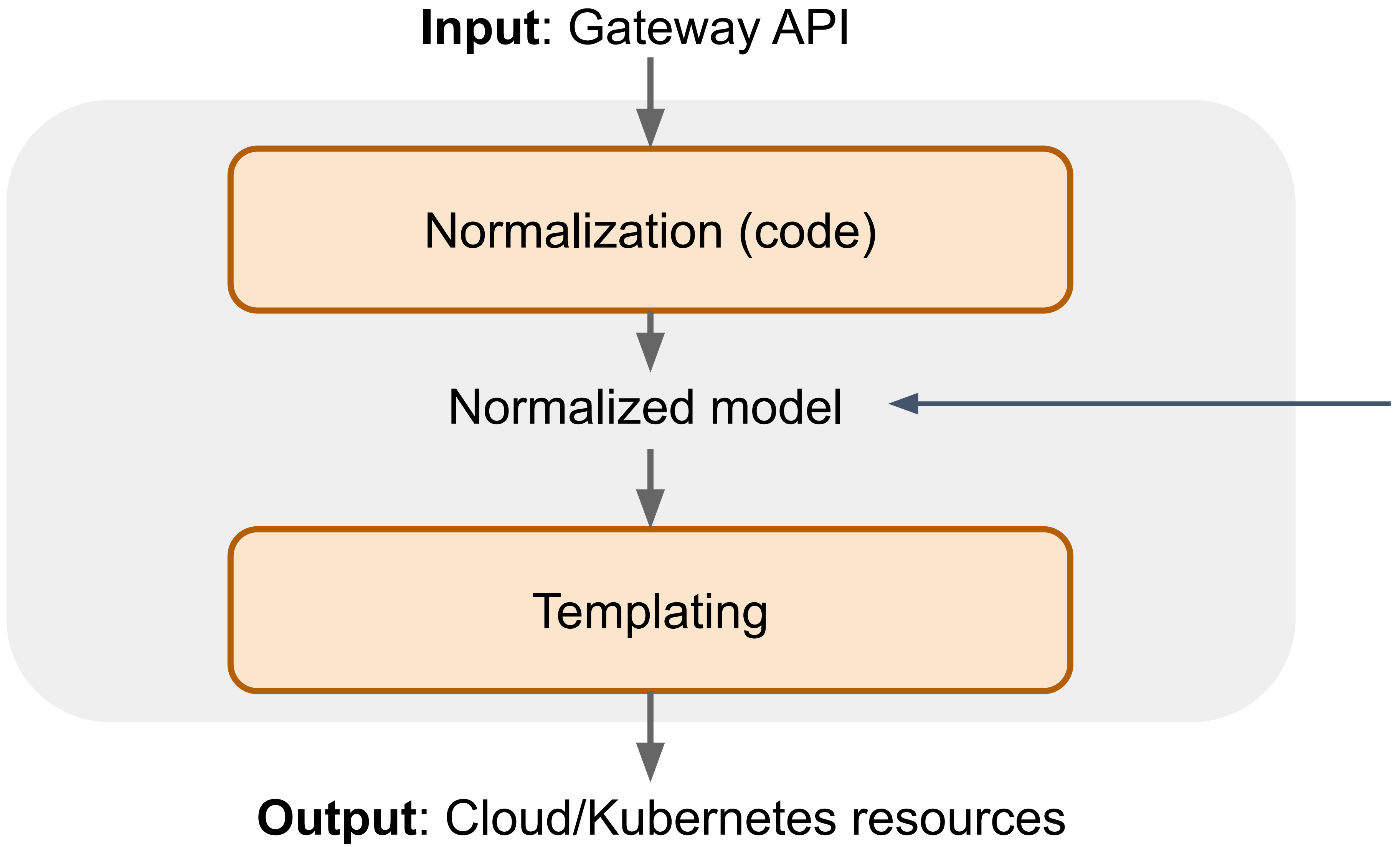


Templating is Difficult



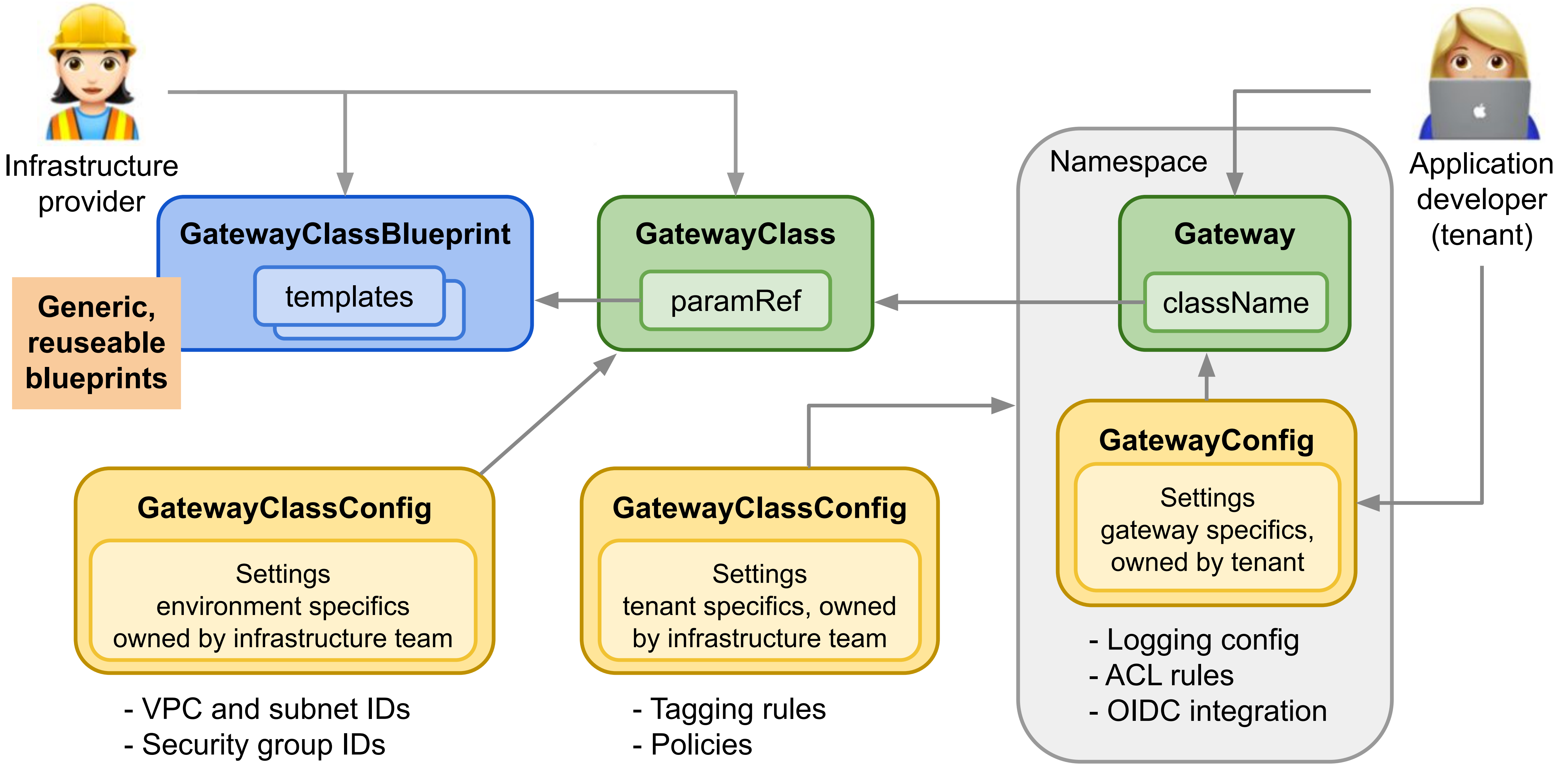
Which hostnames should be used for certificate?

Normalization

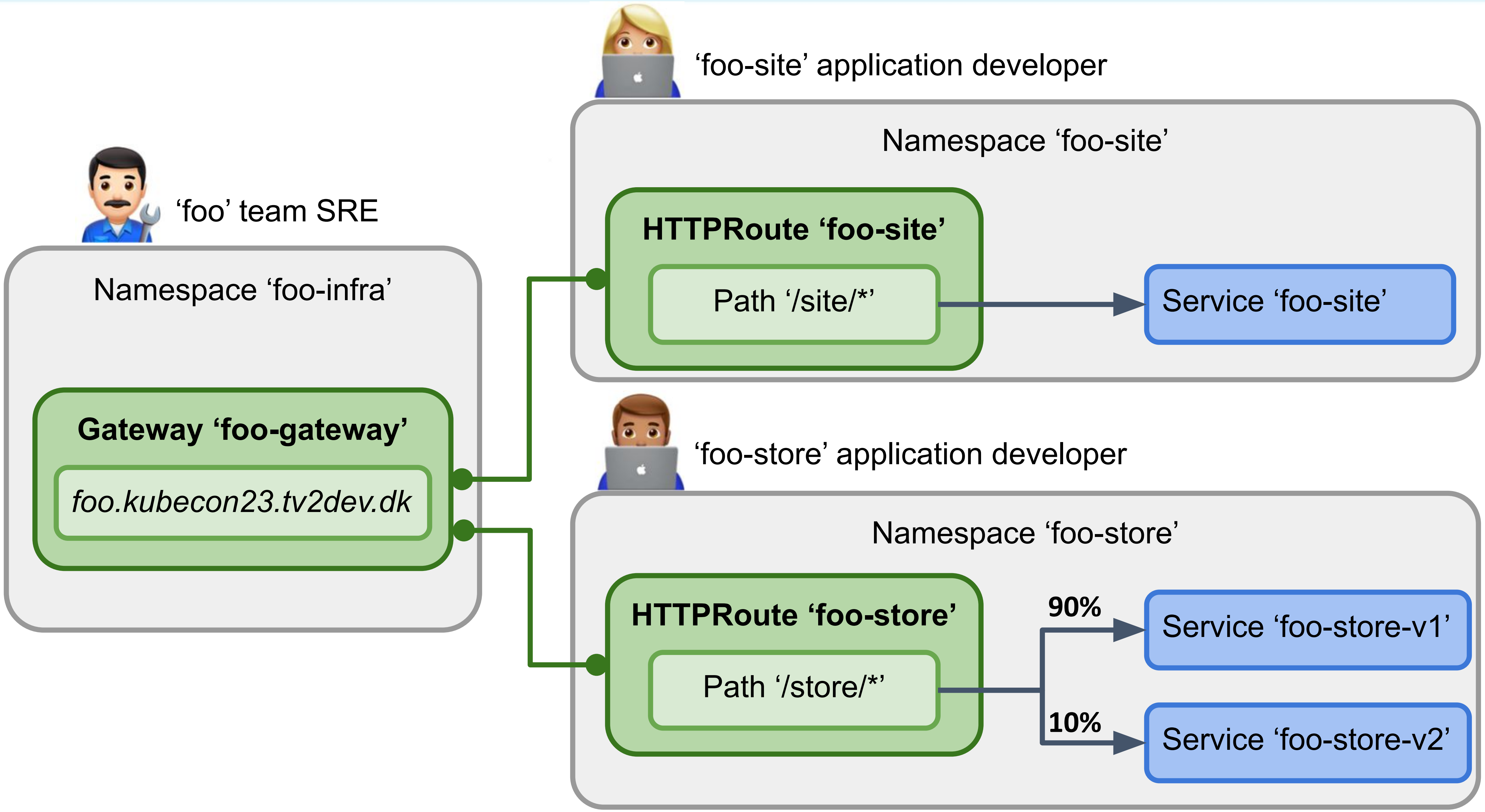


Gateway-controller **pre-processes and generalize** Gateway API specifications into a normalized model

Extending Using GEP-713 (Policy Att.)

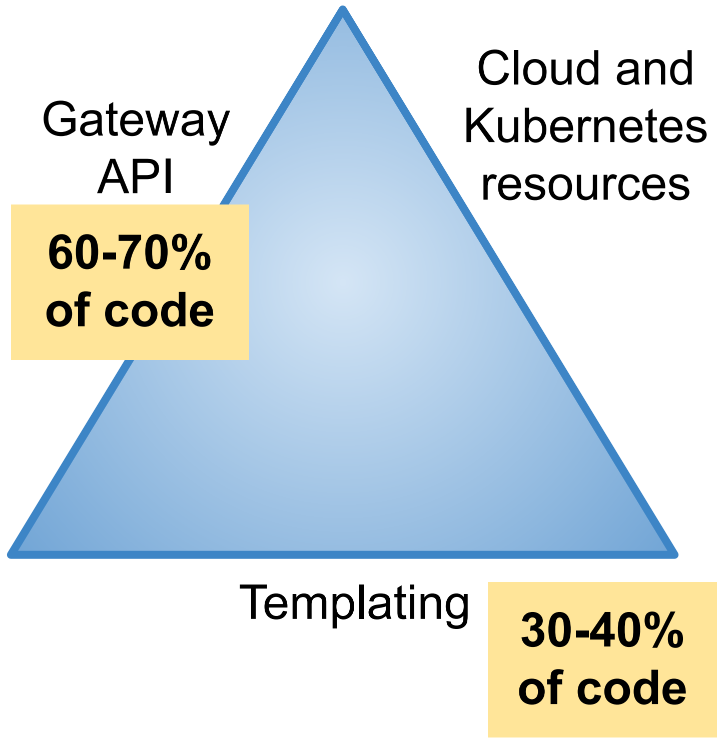


Demo - The ‘Foo’ Example Use case



Learnings

- Gateway API is not a trivial API to implement
- Templating still seems like a balanced choice
- Namespaced resources which owns cluster-scoped resources is a challenge
- Immutable cloud resources are hard to handle
- GatewayClasses are considered immutable - conflicts with e.g. users updating GatewayConfigs
- Cloud resources with ‘Ready’ status but not being Ready
- Normalization process may become a ‘garbage can’ for everything that does not fit templating



Thank You!

KubeCon | CloudNativeCon Europe 2023

It’s open-source, come and help

Feedback and contributions are more than welcome e.g.,

- Blueprints for different cloud providers and use cases

https://github.com/tv2-oss/bifrost-gateway-controller

michaelvl

mvillumsen VillumsenMartin

TV2 Danmark

We are hiring!
Feel free to reach out