

AWS
re:Invent

CON 367

Introducing AWS App Mesh

Shubha Rao
Sr. Product Manager
AWS

Tony Pujals
Sr. Developer Advocate
AWS

Introducing AWS App Mesh



AWS App Mesh

Visibility and control for microservices on AWS

App Mesh is a service mesh

Agenda

A lot of whys ...

Why Microservices

Why Service Mesh

Why Service Mesh Control Plane

Why App Mesh

And some hows ...

What is it

Features

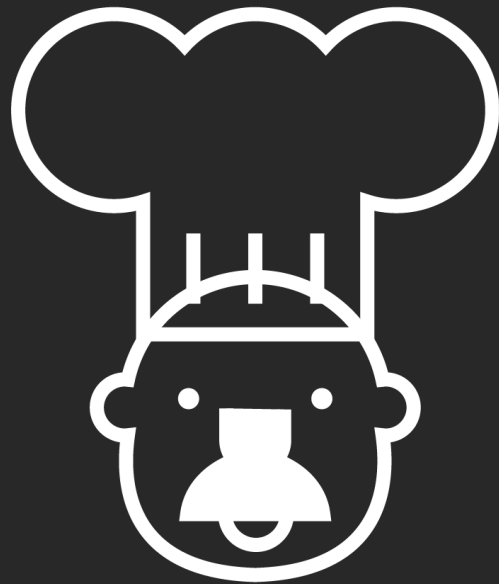
Roadmap

Representing my application
to use App Mesh

Demo

Similar to

Chef-special
boutique restaurant



Franchise



Similar to

Chef-special boutique restaurant

Innovation slow and gradual—relies
on chef brand/recognition

Slow evolution—some succeed,
some fail

Franchise

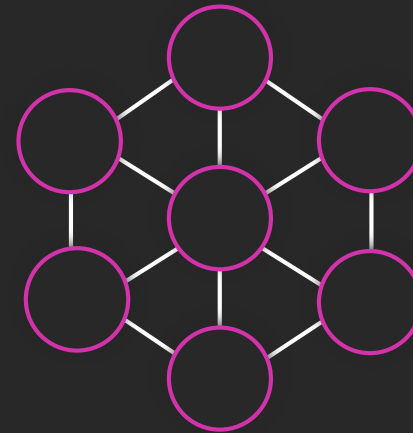
Preset guidelines on menu
options and preparation

Faster rollouts once
success is proven

Microservices increase release agility



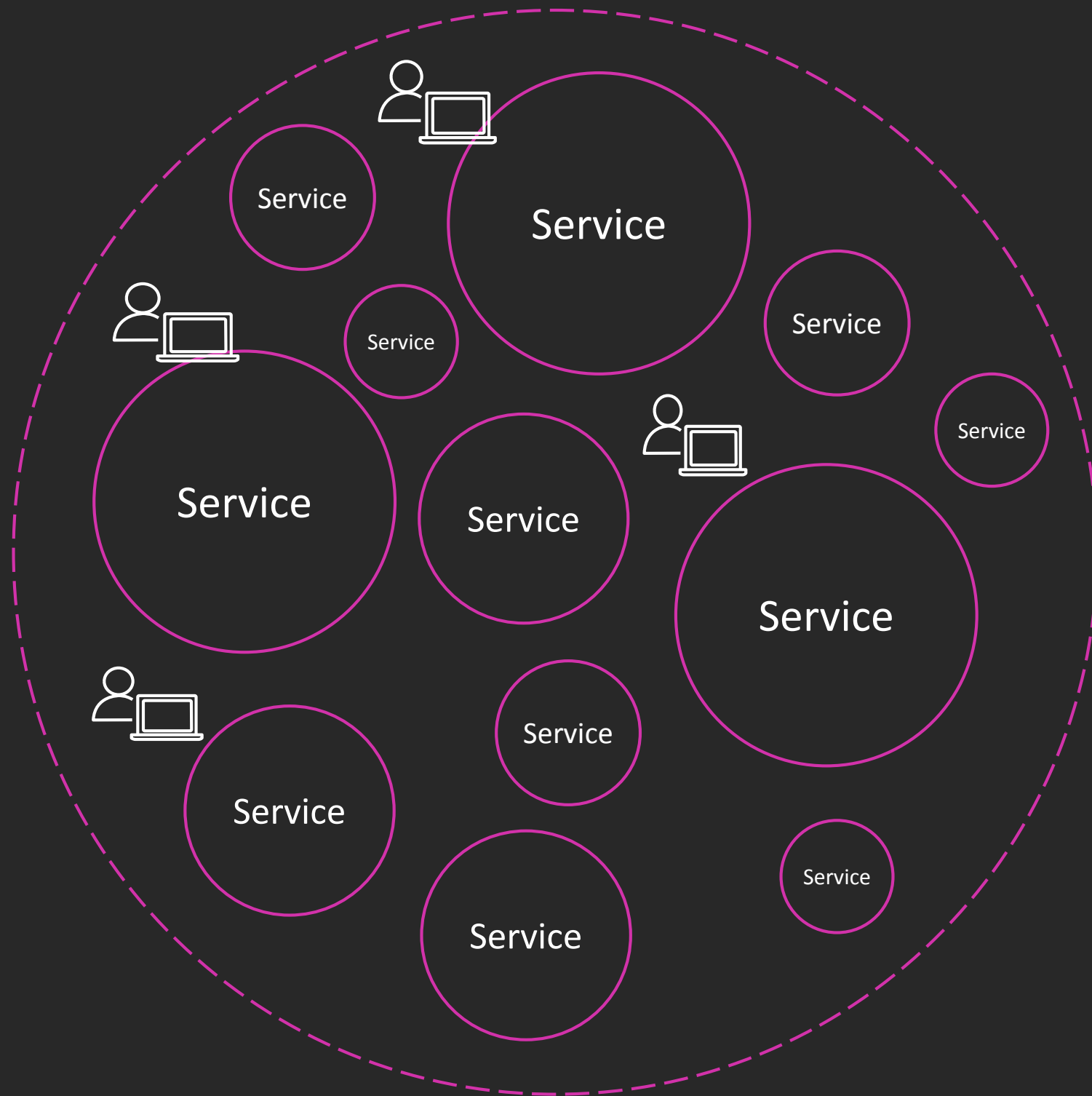
Monolithic application

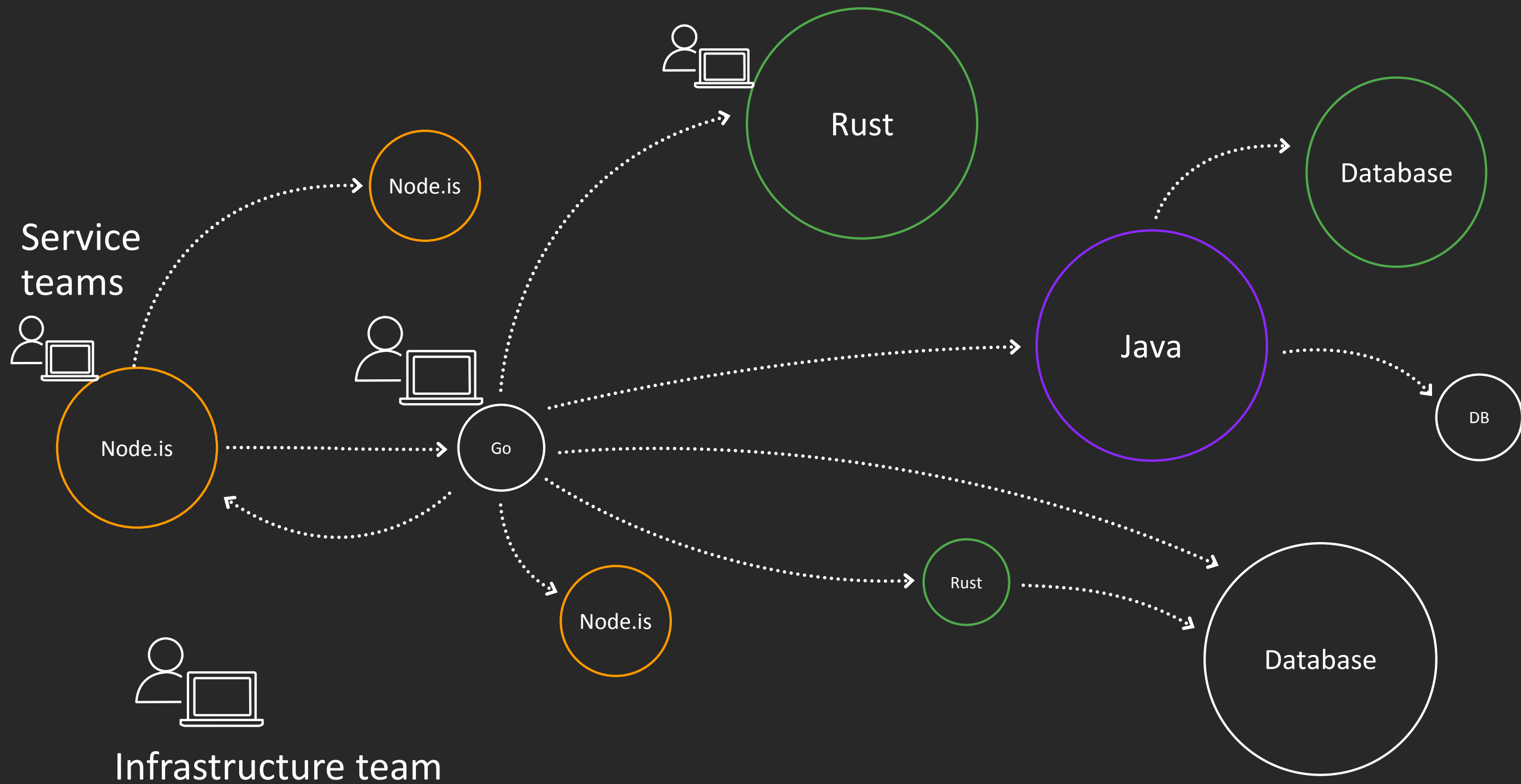


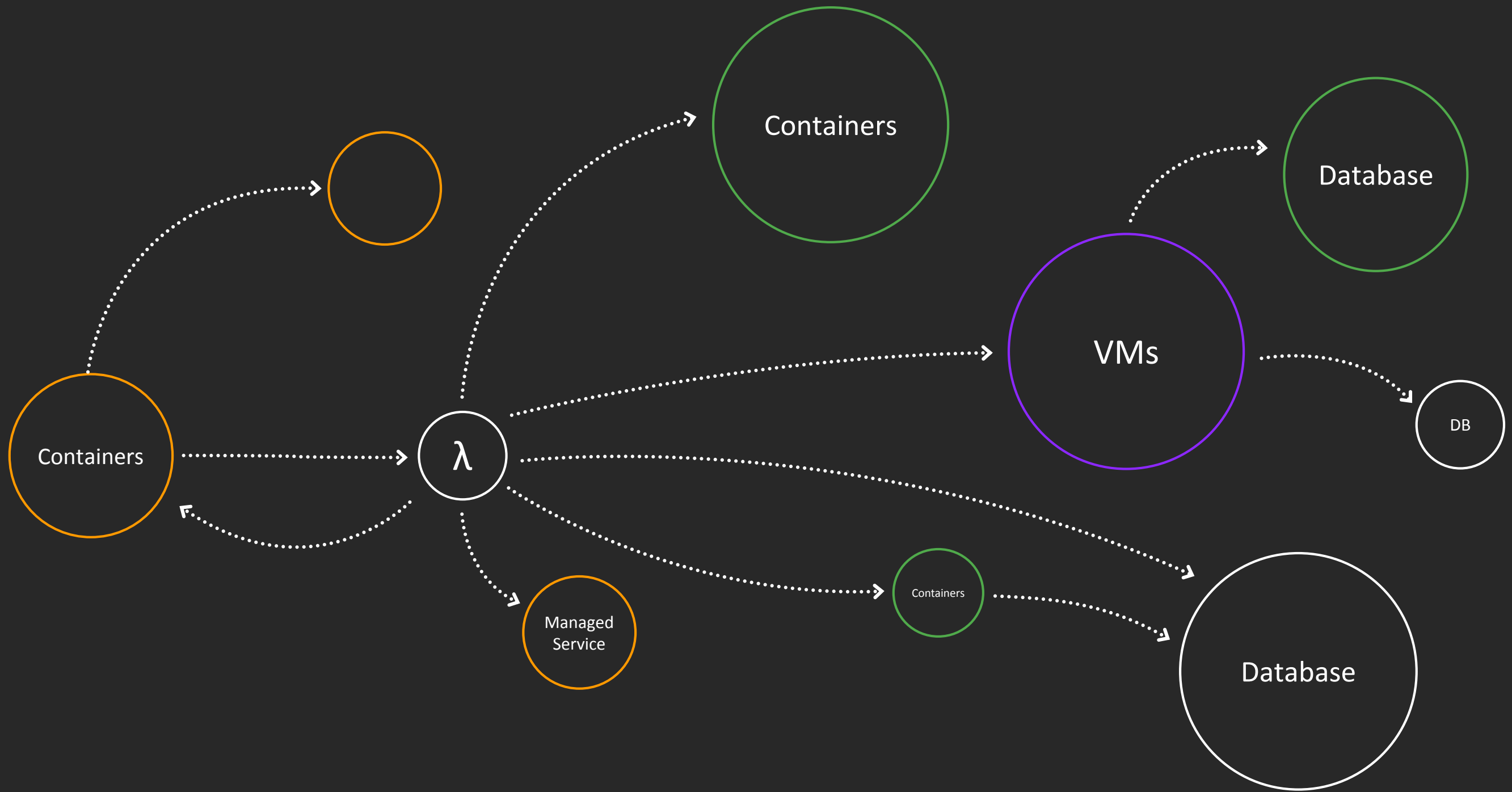
Microservices



Monolith







Best practices are emerging and evolving rapidly

Standardize and automate operations by **modeling infrastructure as code**

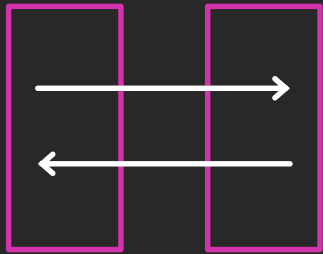
Improve application performance with **full stack observability**

Create a culture of innovation by **organizing into small DevOps teams**

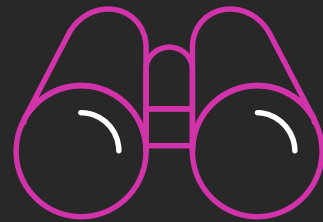
Update applications and infrastructure quickly by **automating CI/CD**

Ensure trust by **automating security and compliance**

What is needed



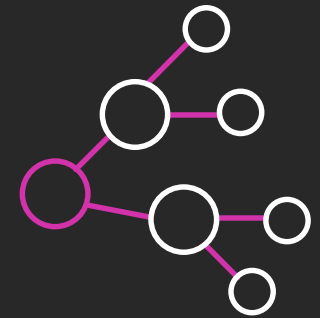
Consistent communications
management



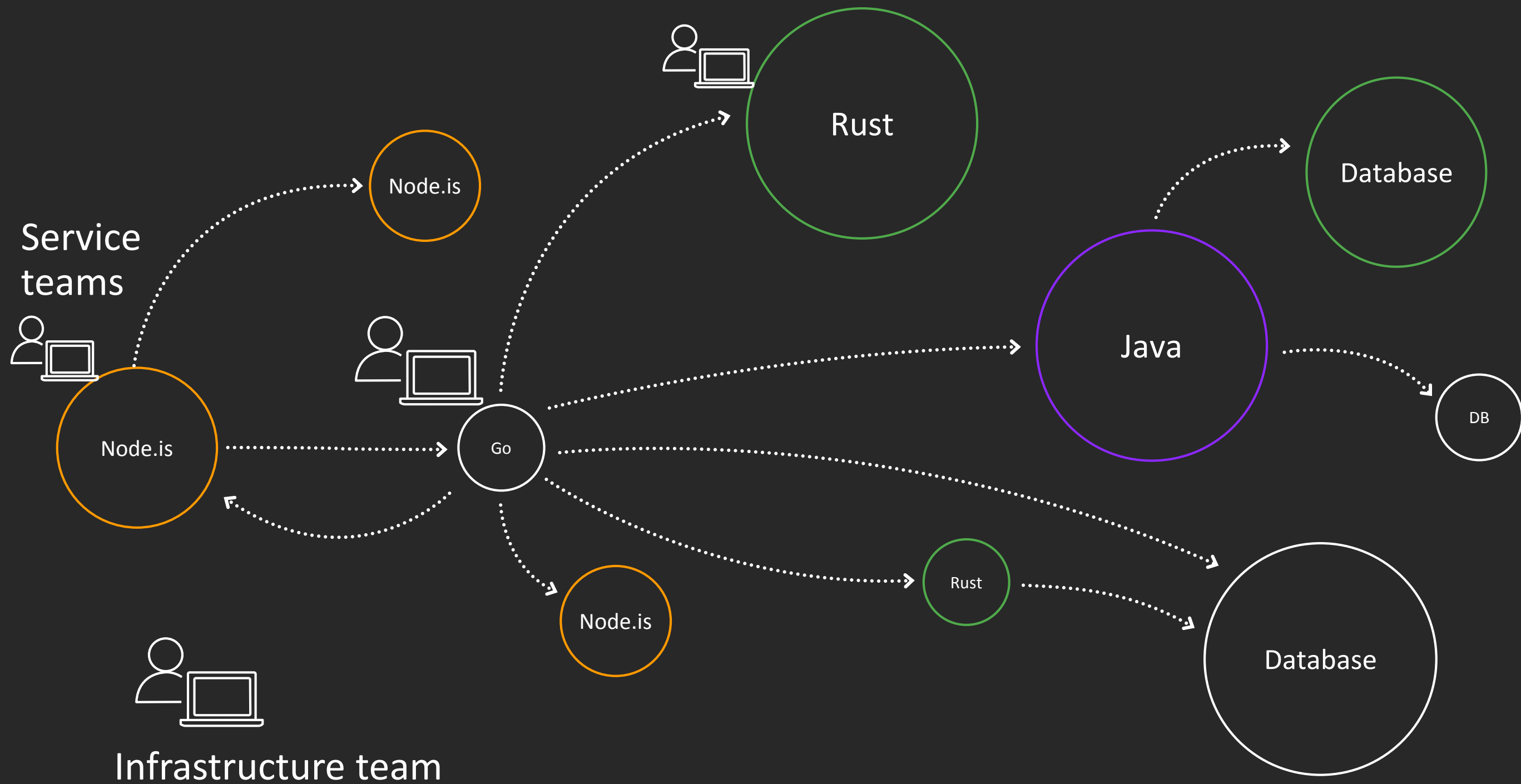
Complete visibility



Failure isolation
and protection



Fine-grained deployment
controls

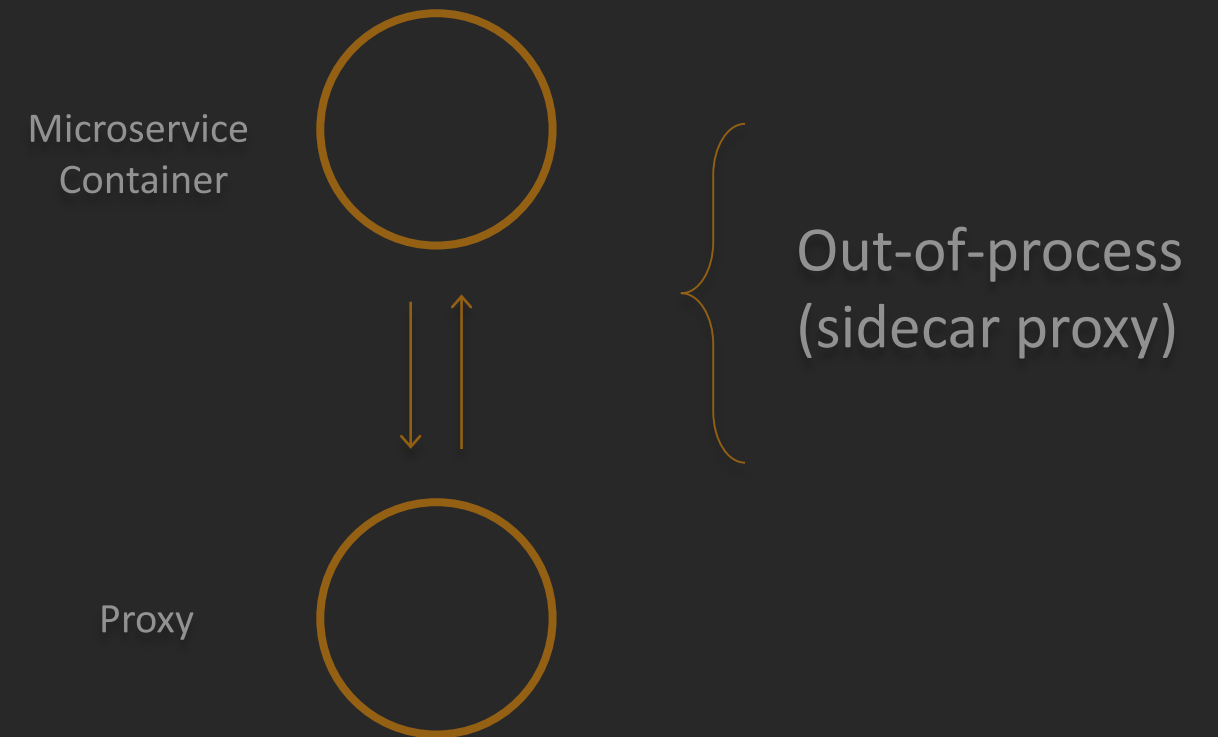


Implementation options

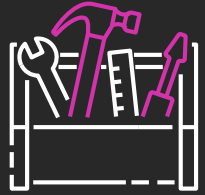
Option 1



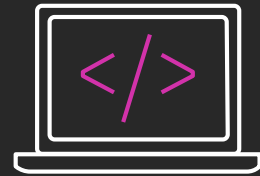
Option 2



Option 1: in-process SDK



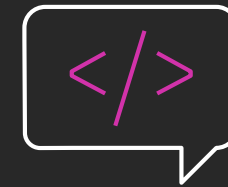
SDK maintenance



Application code changes



Consistency across services

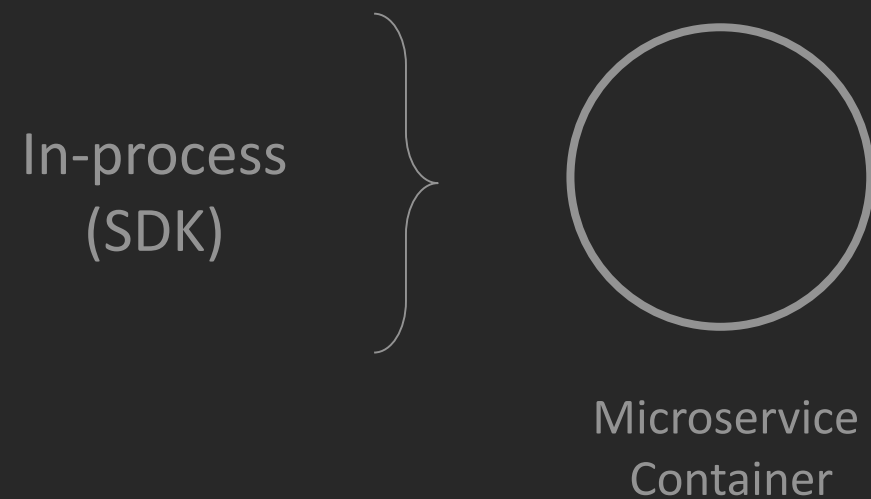


Languages

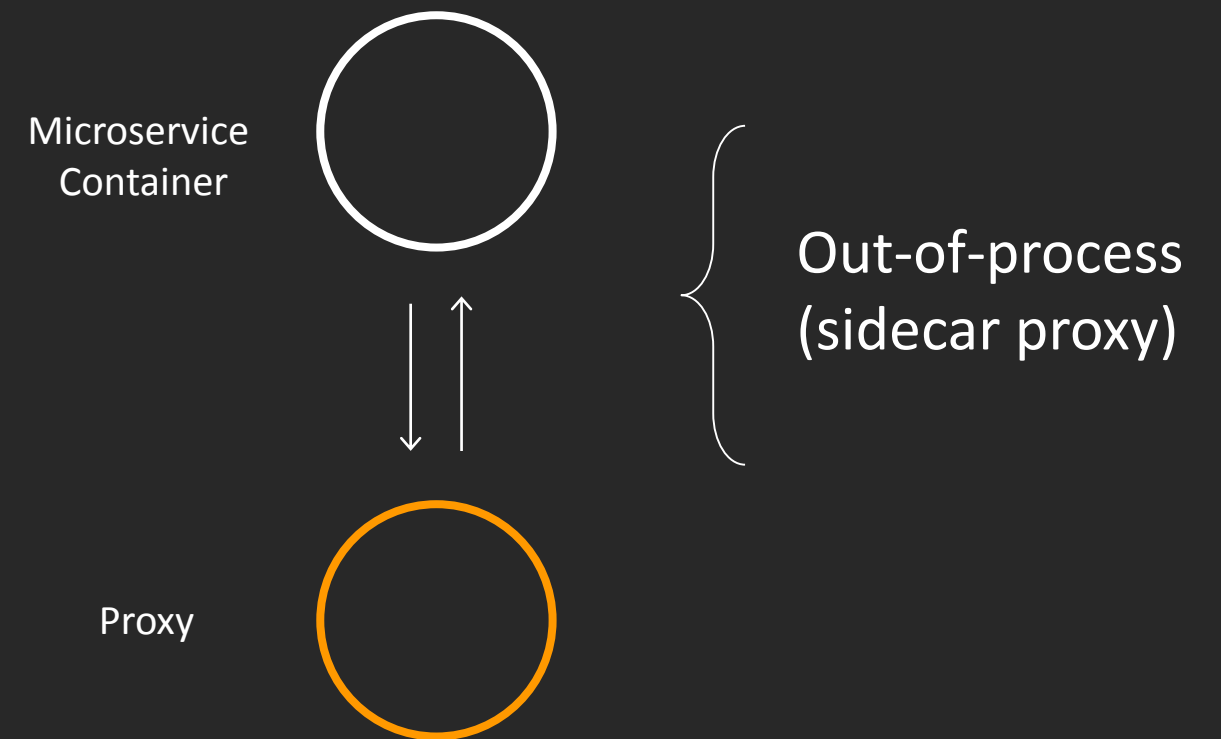
Java
Scala
Node.js
Python
C++
Django
.NET
GO
...

Implementation options

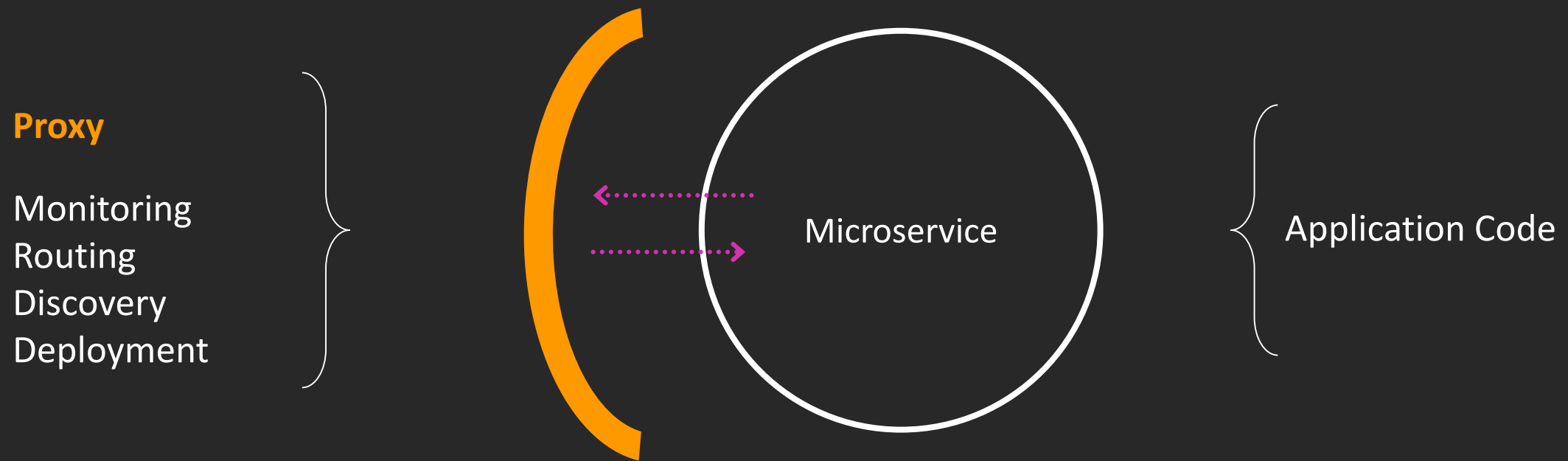
Option 1



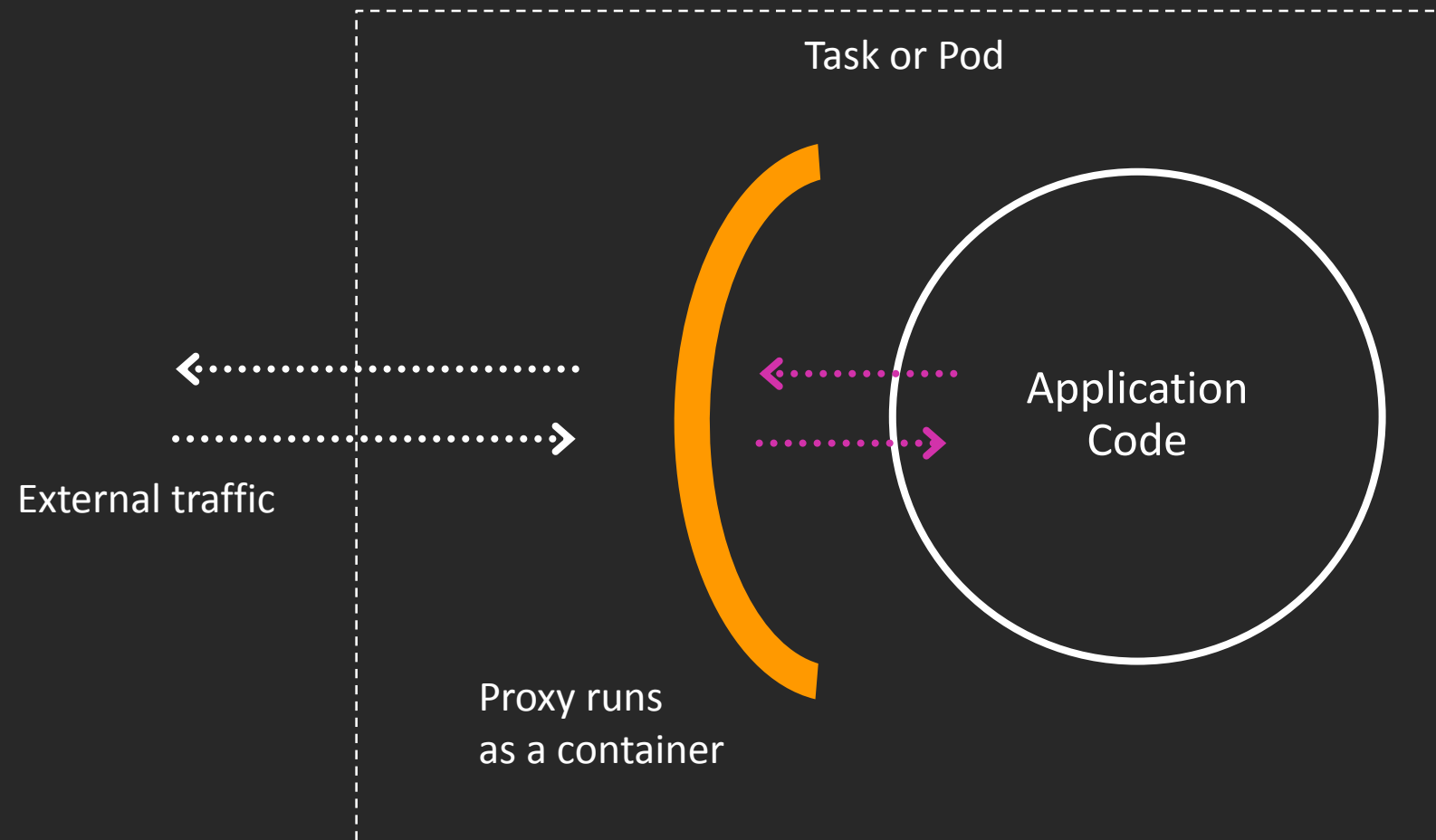
Option 2



Option 2: side-car proxy

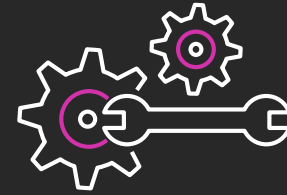


Option 2: side-car proxy

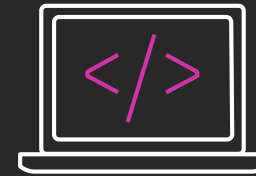


Option 2: side-car proxy

Proxy



Decouples install/upgrade



Configurable—separates business logic from operations



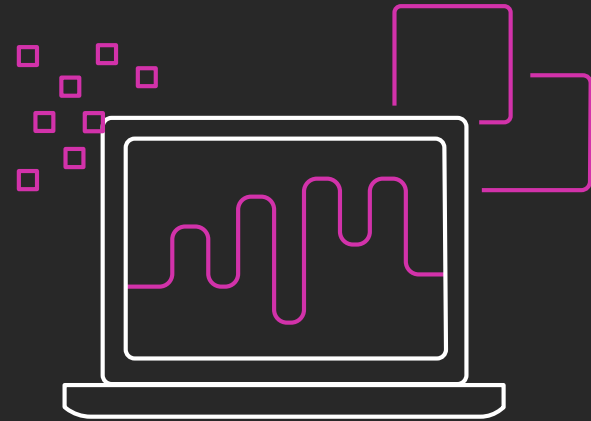
Minimizes inconsistencies

Why service mesh proxy

vs. libraries or app code



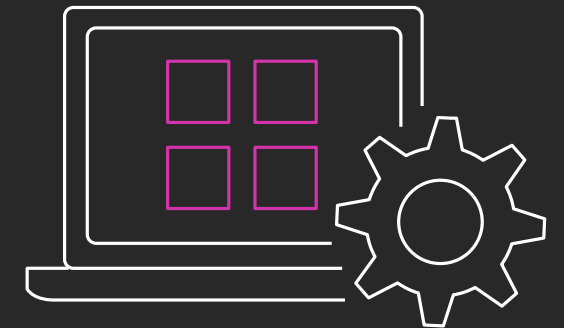
Reduce work required by
developers



Follow best practices



Use any language
or platform



Simplify visibility,
troubleshooting, and
deployments

Overall—migrate to microservices safer and faster

App Mesh uses Envoy proxy



OSS project

Wide community support, numerous integrations

Stable and production-proven

“Graduated Project” in Cloud Native Computing Foundation

Started at Lyft in 2016

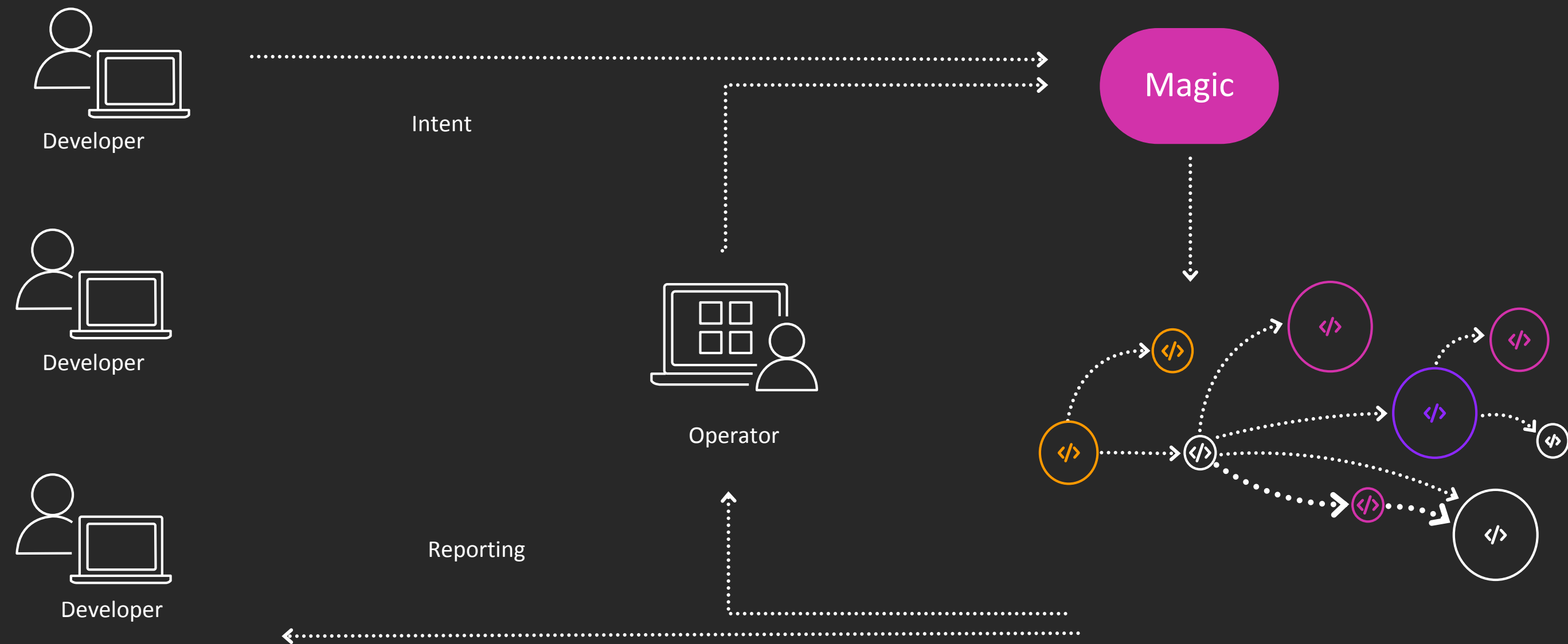
How do we tell every proxy what to do?



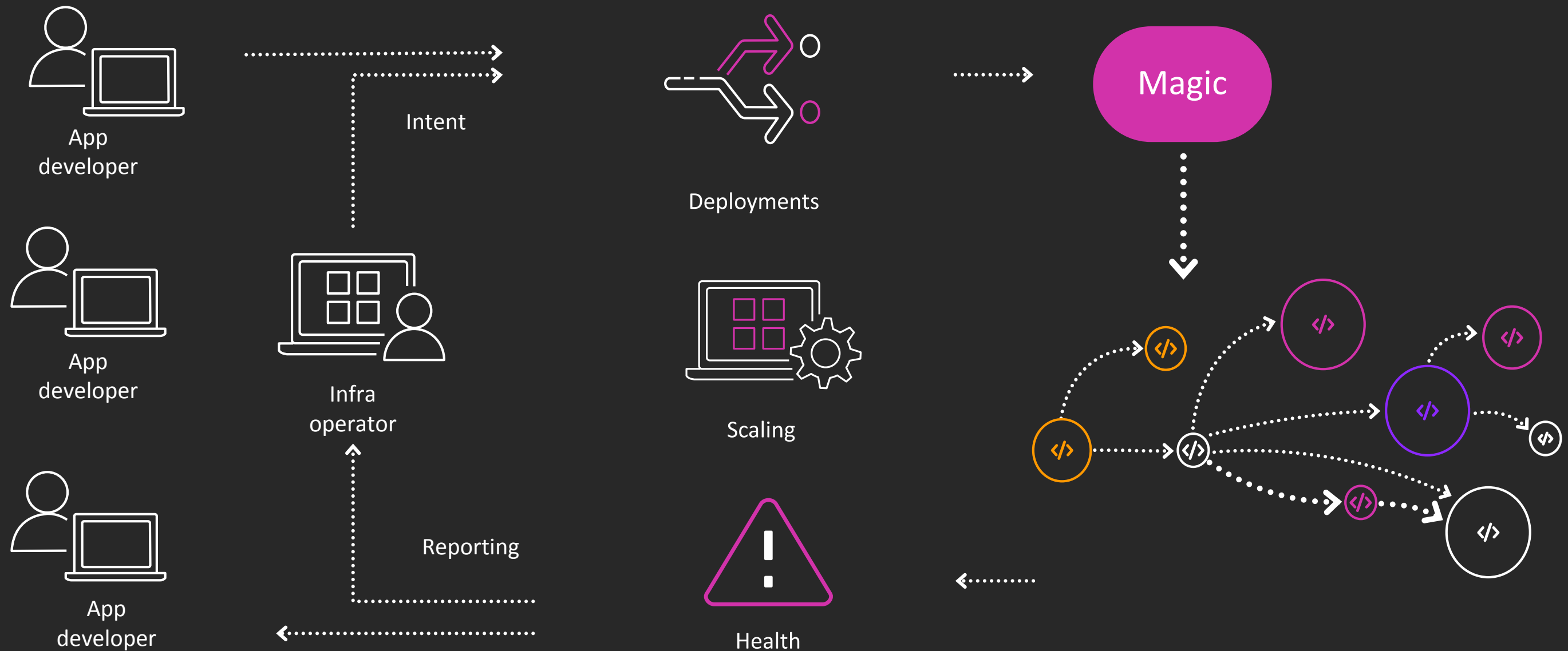
Configuring lots of proxies is hard!



Easily deliver configuration and receive data

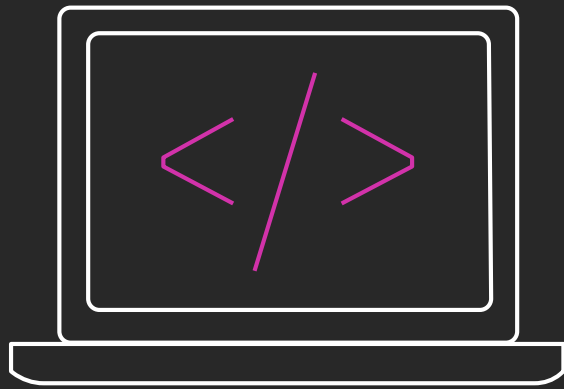


Dynamic state changes impact proxy configuration



Why service mesh control plane

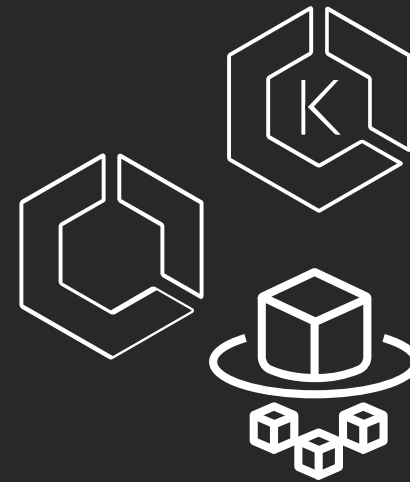
vs. static config or self-built control plane



Don't need to spend dev
to build and ops
to maintain



Not tied to application
deployment system
(e.g., container orchestration)



Works across different
compute systems



Reliably store
and distribute
configuration

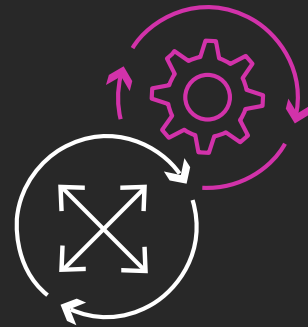
Additional use cases based on customer input

vs. other control plane solutions

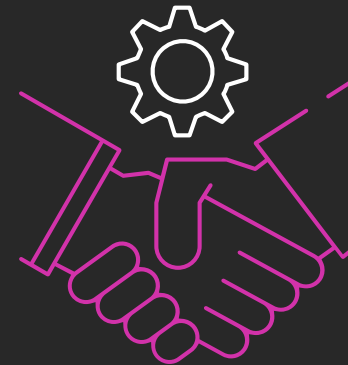


Works across
Clusters

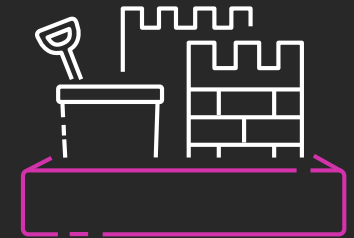
Container services
Containers & VMs



AWS run for scale
and stability

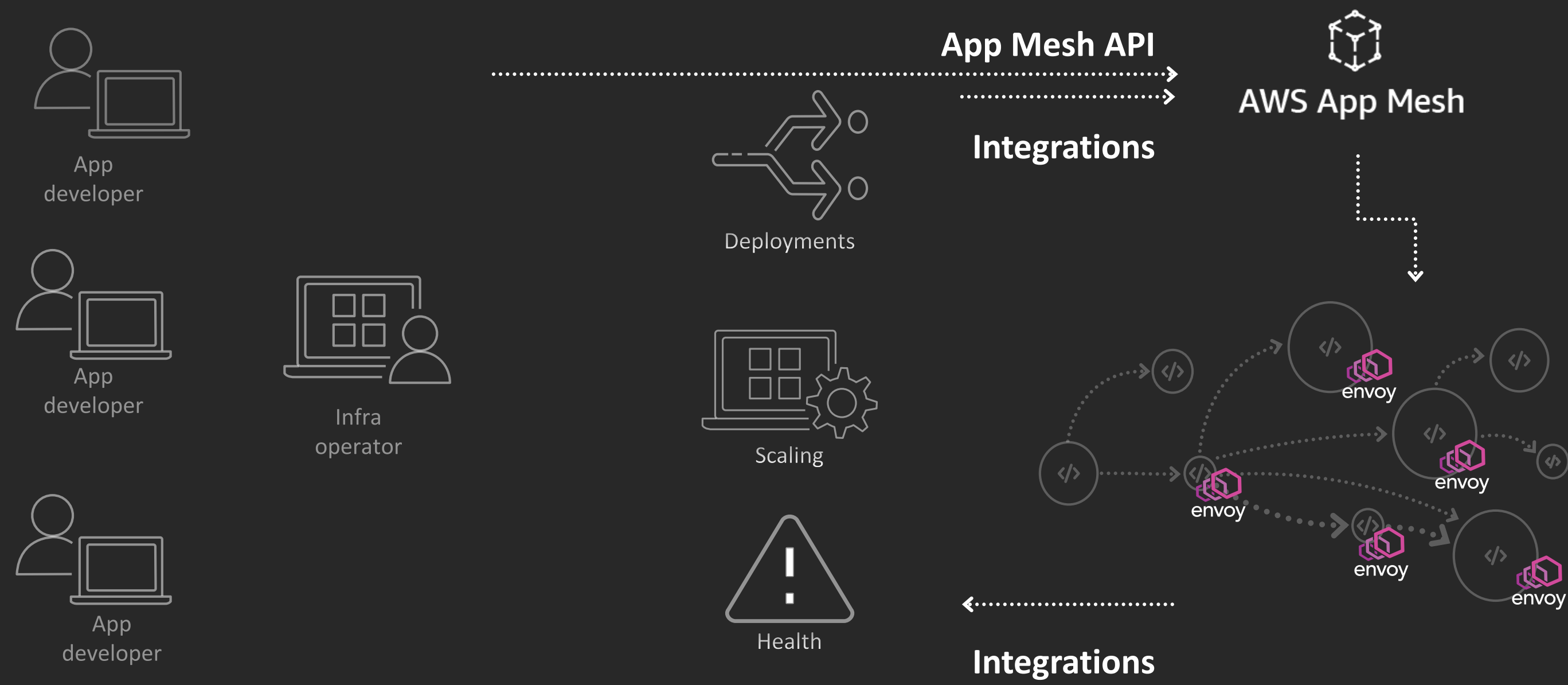


AWS and partner
integrations



Extensible architecture
from OSS base

Dynamic state changes impact proxy configuration



App Mesh configures every proxy



App Mesh configures every proxy



Today App Mesh is available as a preview for all customers



AWS App Mesh

Observability and traffic control

Easily export logs, metrics, and traces (available)

Client side traffic policies—circuit breaking, retries (coming soon)

Routes for deployments (available)

AWS container services compatibility

Today: Amazon Elastic Container Service (Amazon ECS) & Amazon Elastic Container Service for Kubernetes (Amazon EKS)

Coming soon: AWS Fargate

Regions



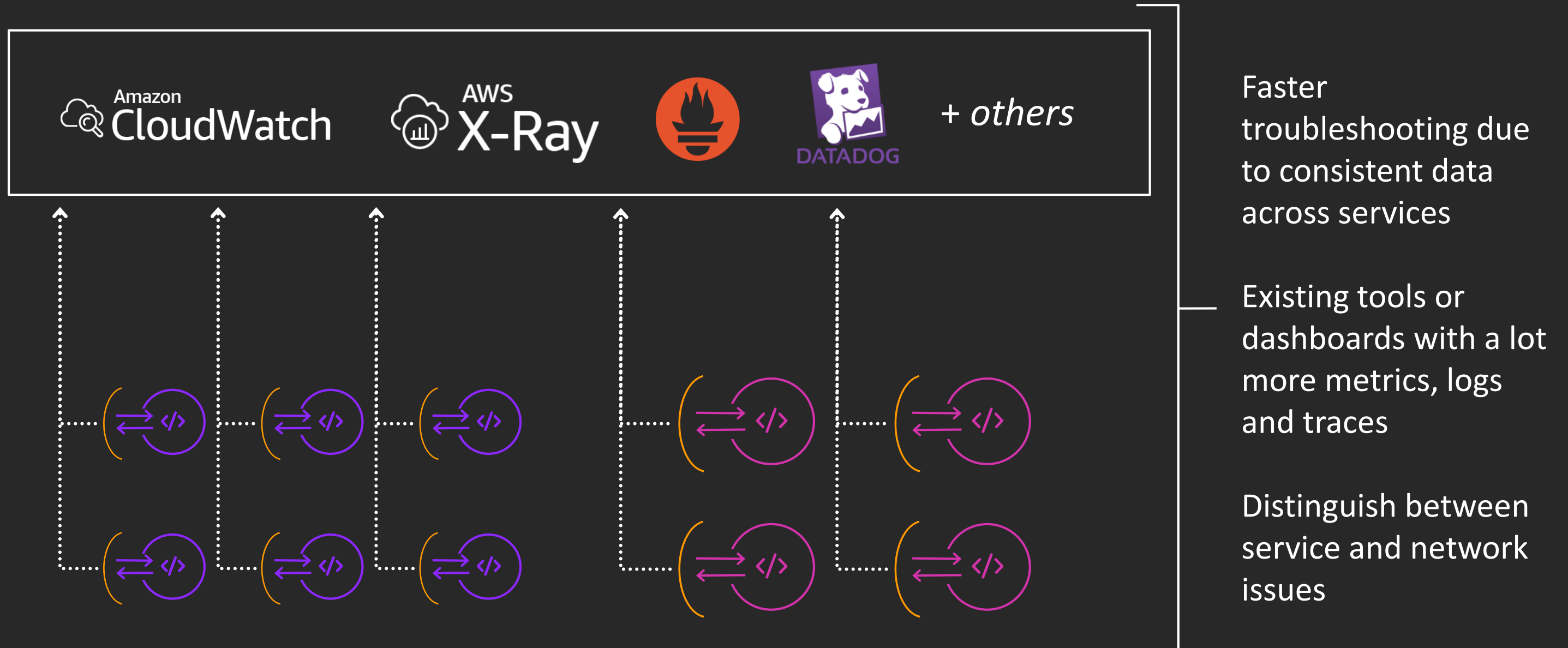
A dark gray world map serves as the background. Three regions are highlighted with white text labels. The 'US West' label is positioned over North America, specifically the western United States. The 'US East' label is positioned over the eastern United States. The 'Europe' label is positioned over Western Europe. Below each region name, specific AWS regions are listed in parentheses.

US West
Oregon (US-West-2)

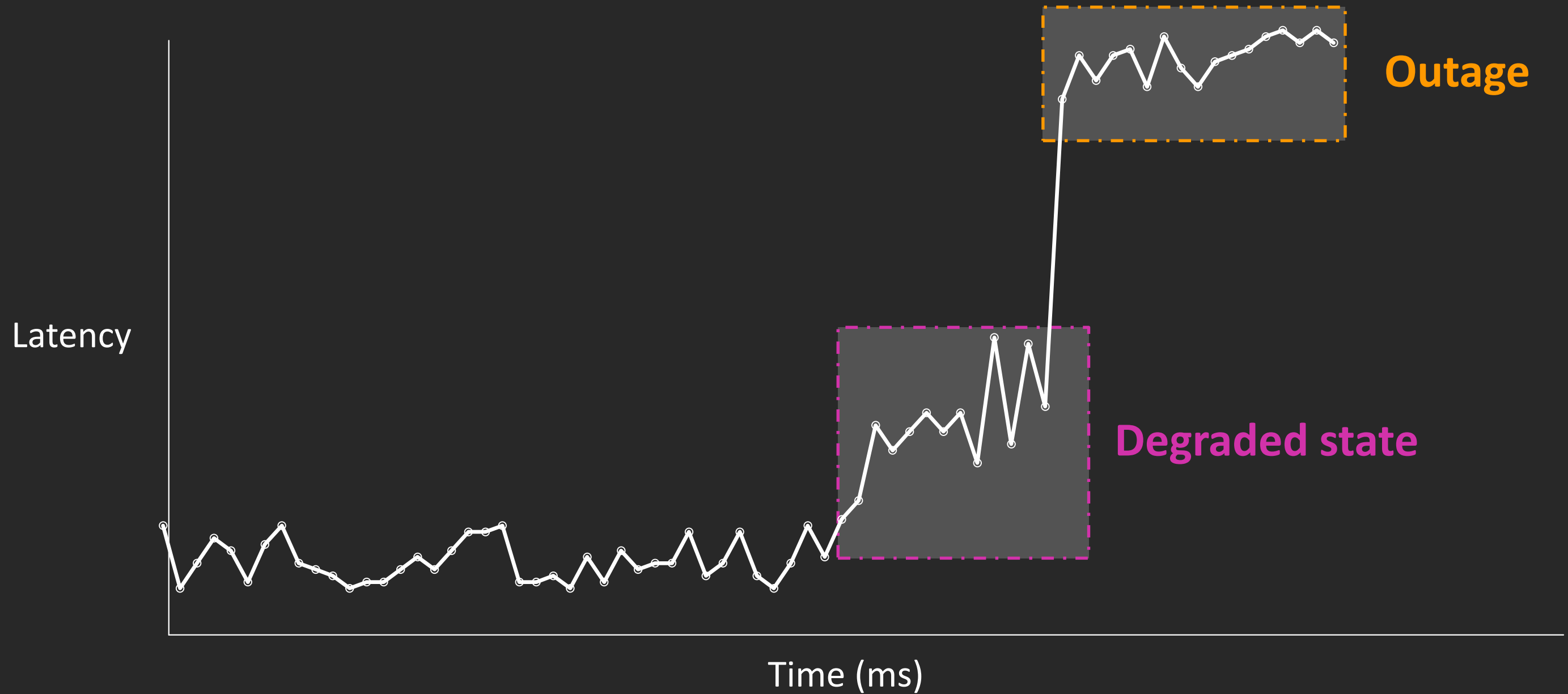
US East
N. Virginia (US-East-1)
Ohio (US-East-2)

Europe
Ireland (EU-West-1)

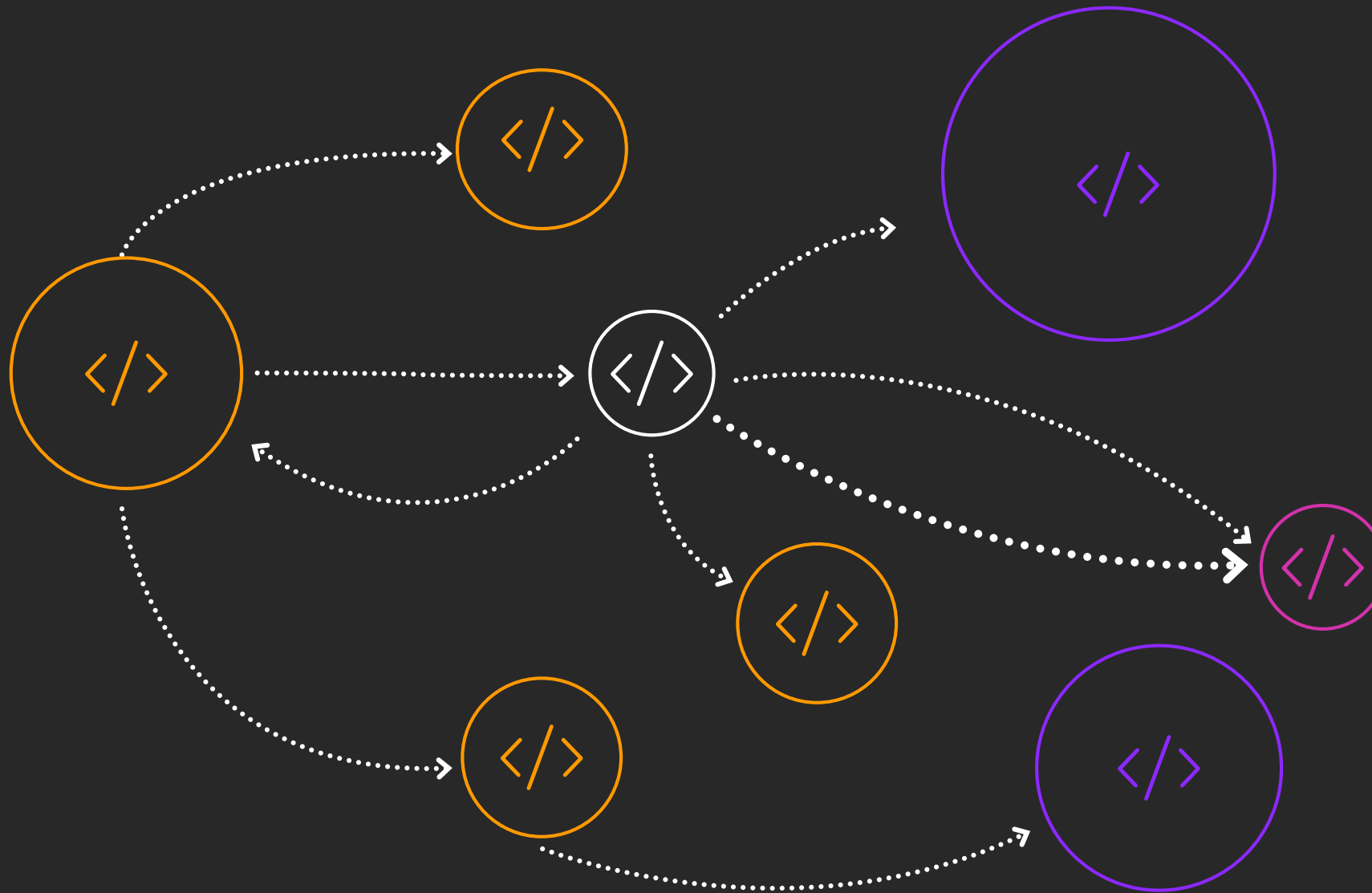
Application observability



Proactive operations helps mitigate issues



Client side traffic management



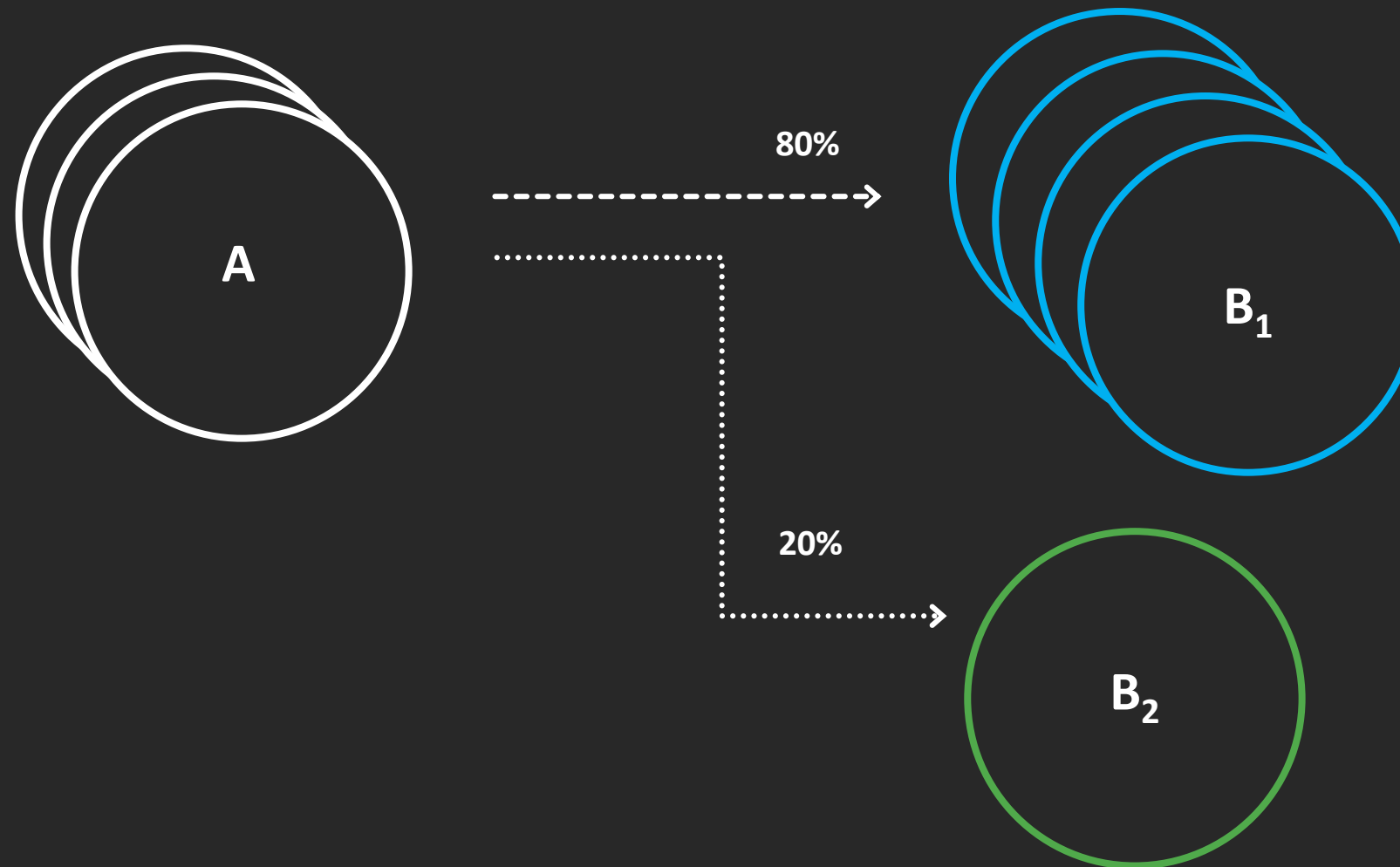
Traffic shaping*

- Service discovery
- Retries
- Timeouts
- Circuit breaks
- Health checks

Routing controls

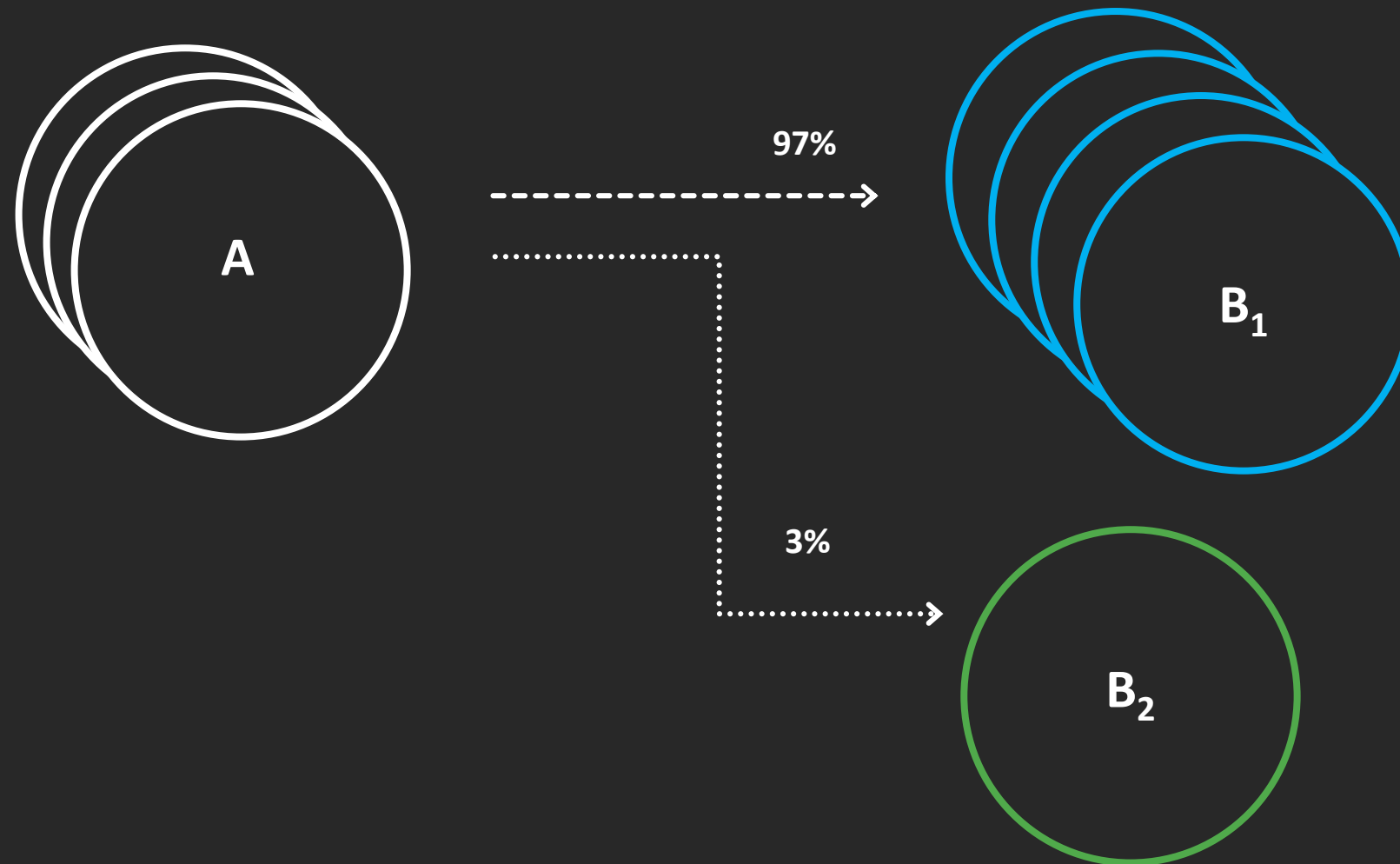
- Protocols support
- Header based
- Cookie based
- Path based
- Host based

Fine-grained deployment control



Can do this now...

Fine-grained deployment control



Preview v/s GA (capabilities)

Preview

API ready

For use with sample apps
not production

HTTP path based routing

Statd based logs,
metrics integrations

GA

Console

Integrations

Traffic management

AWS Cloud Map

Cross account

Amazon EC2

Post GA

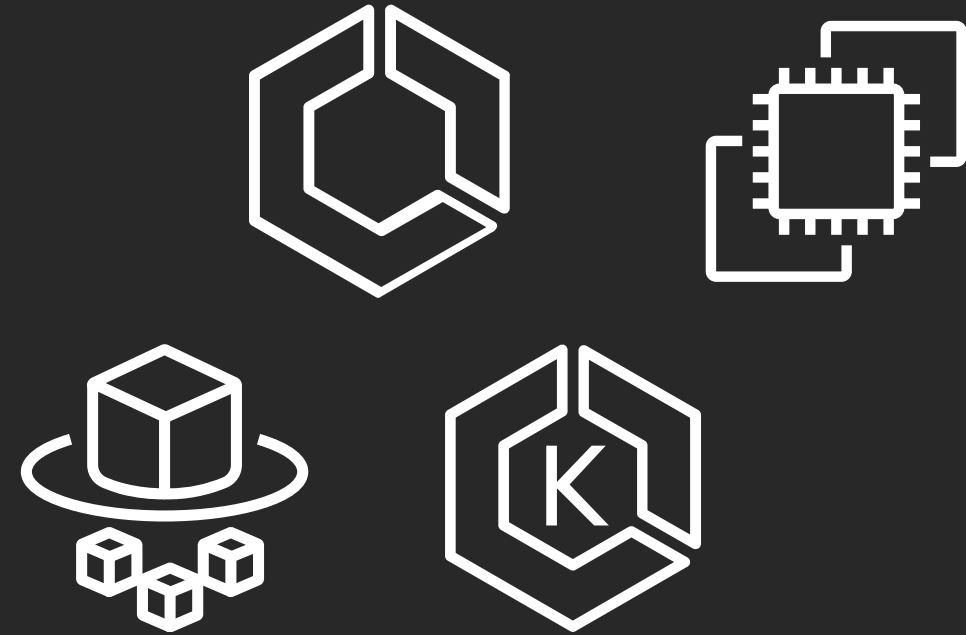
TLS

Ingress

AWS integrations



Service Discovery



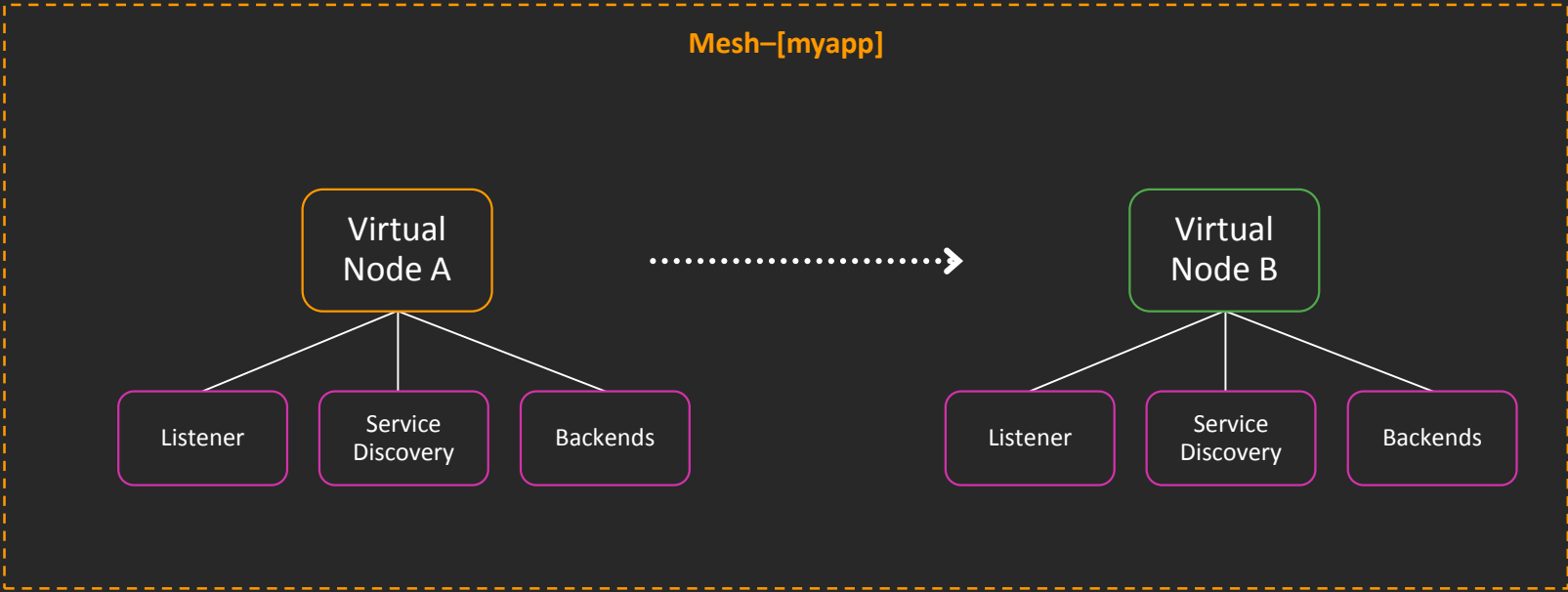
Microservices

How it works

Representing your sample app in App Mesh

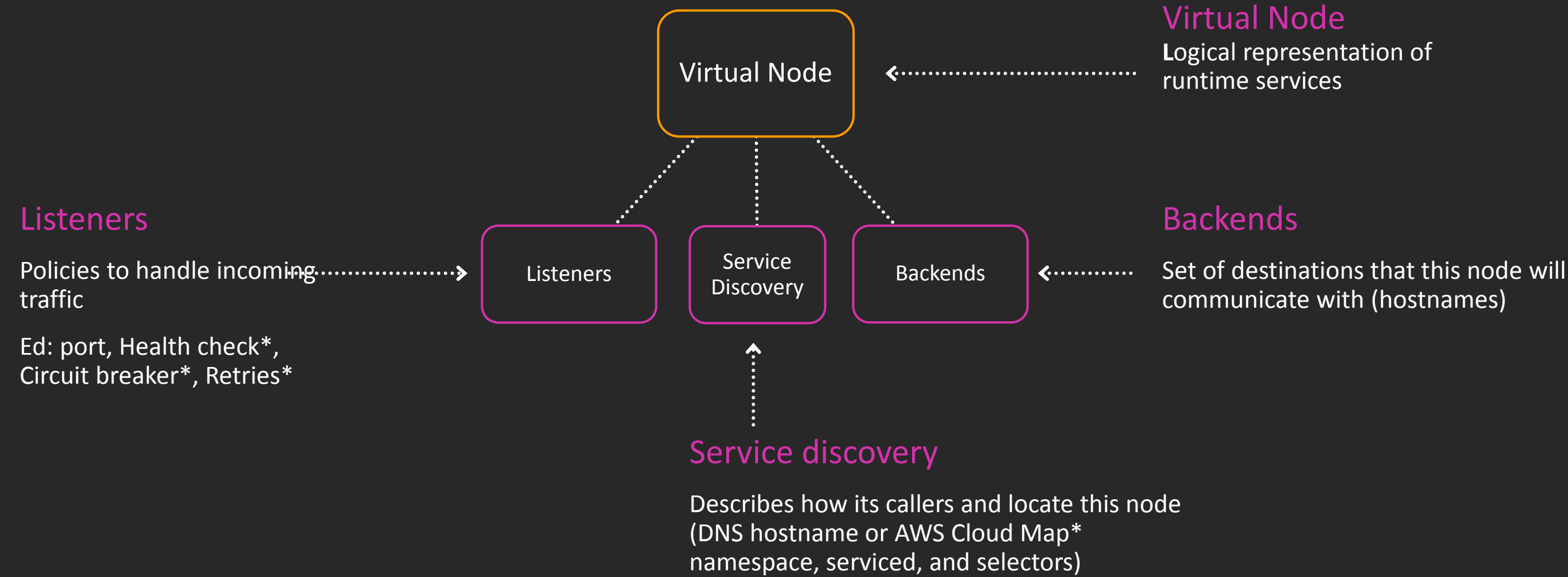


Microservices



App Mesh

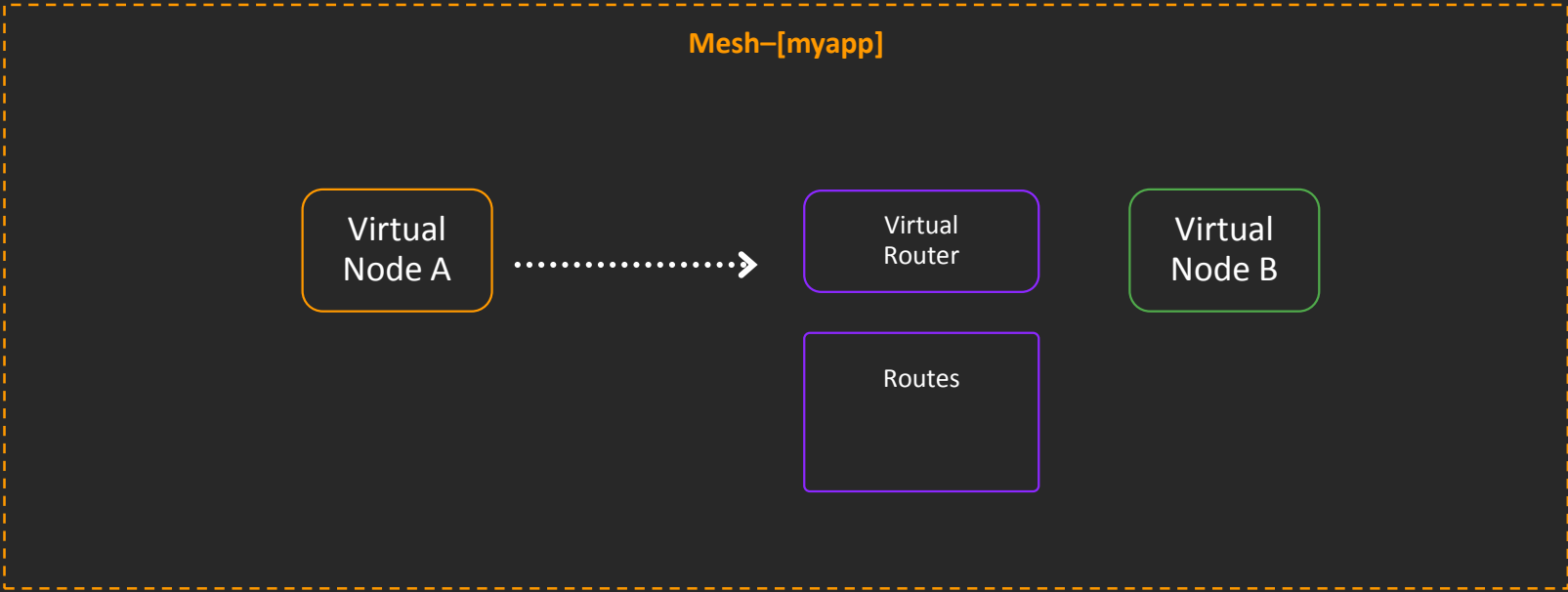
Virtual Node



Create route

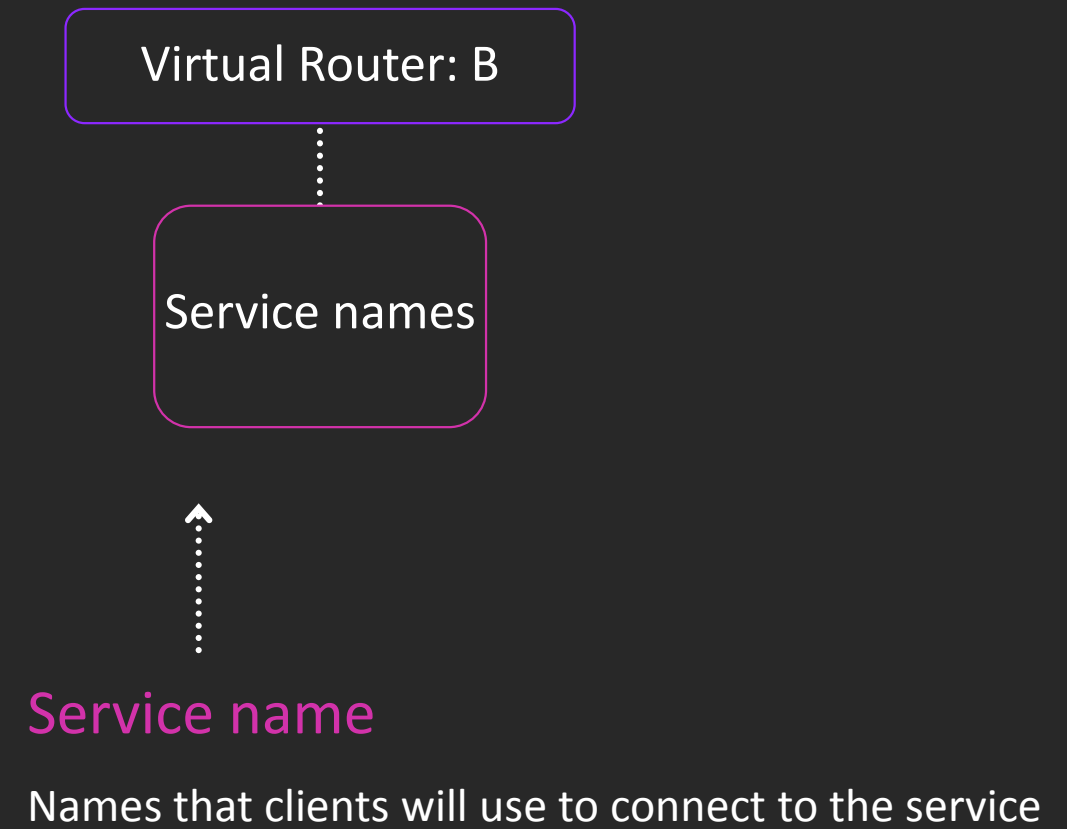
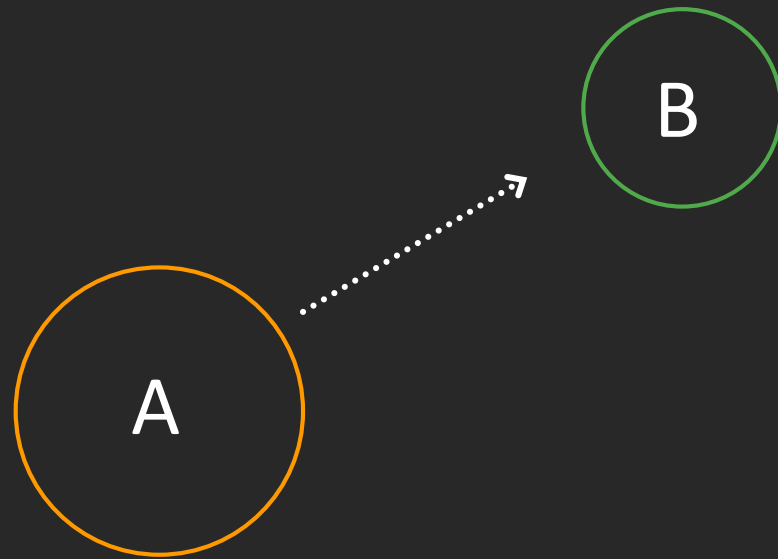


Microservices

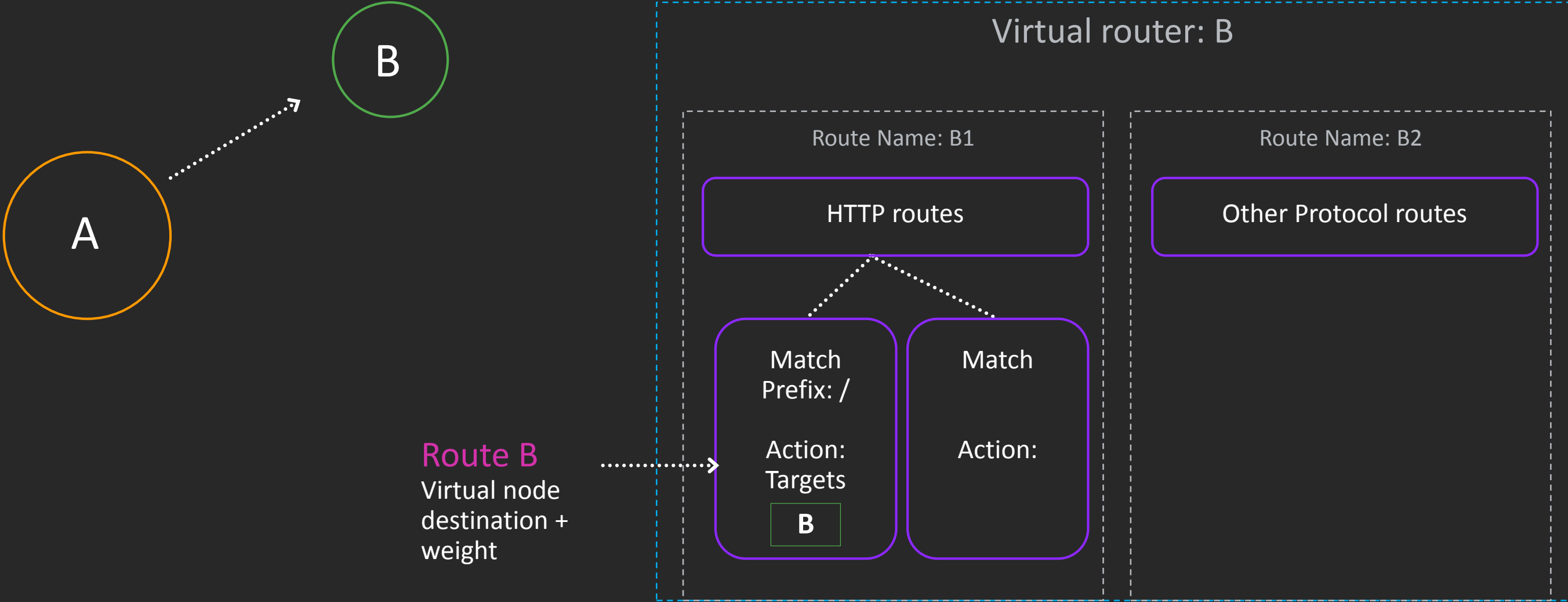


App Mesh

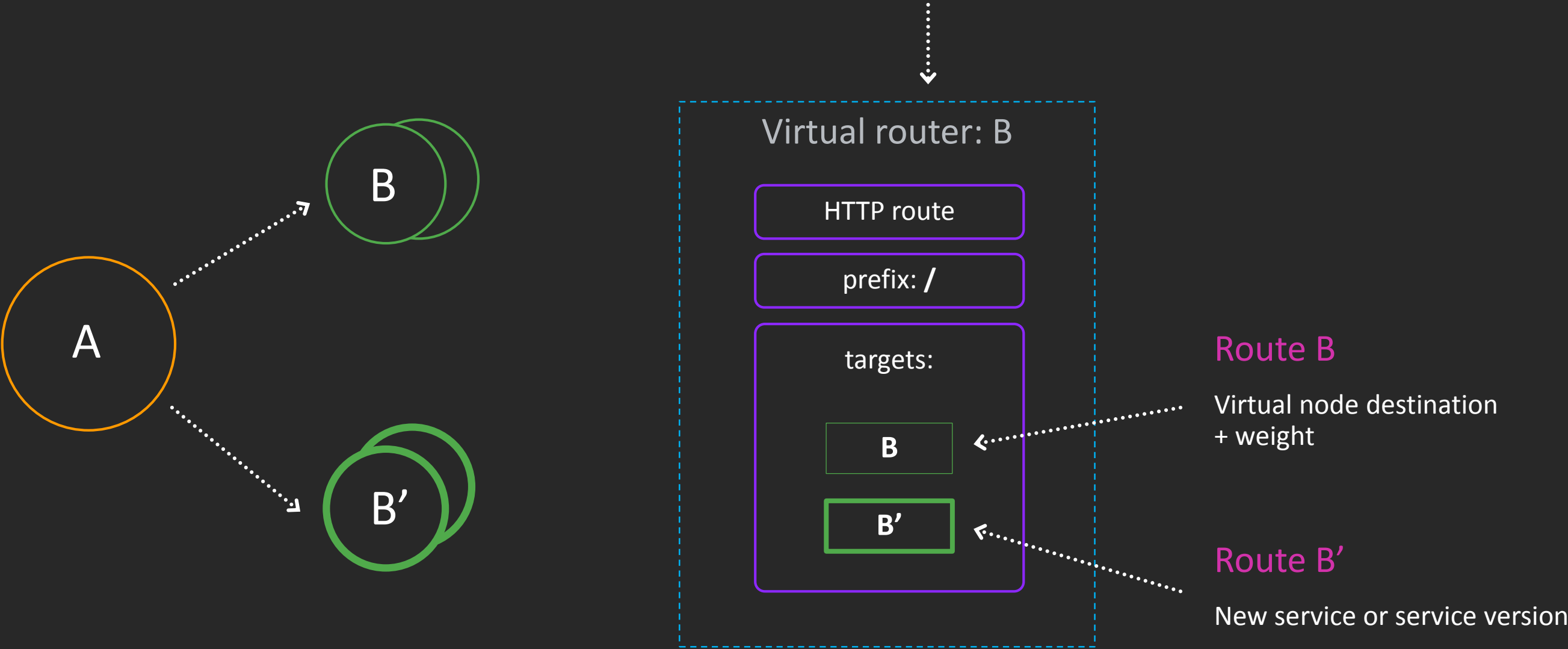
Virtual router

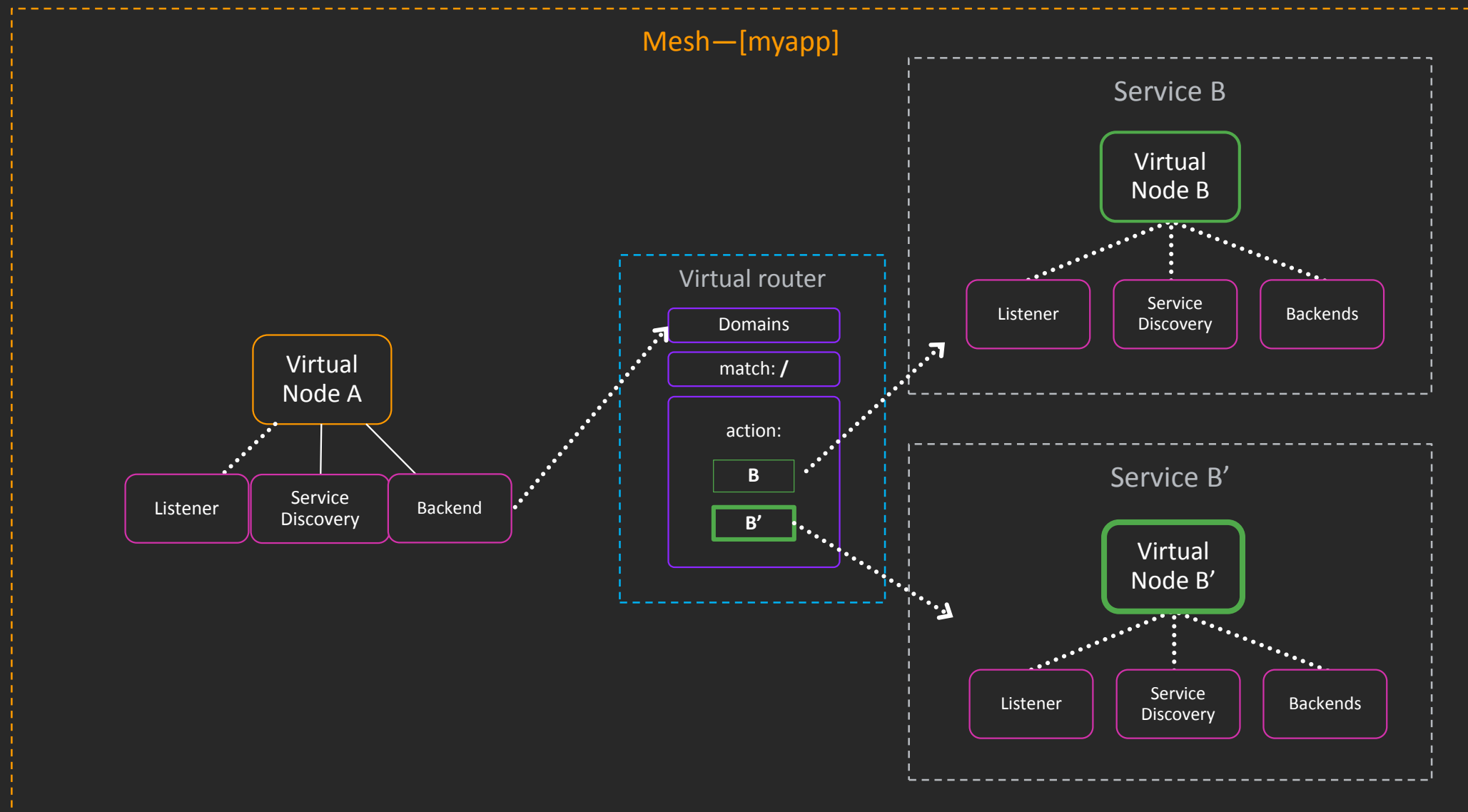
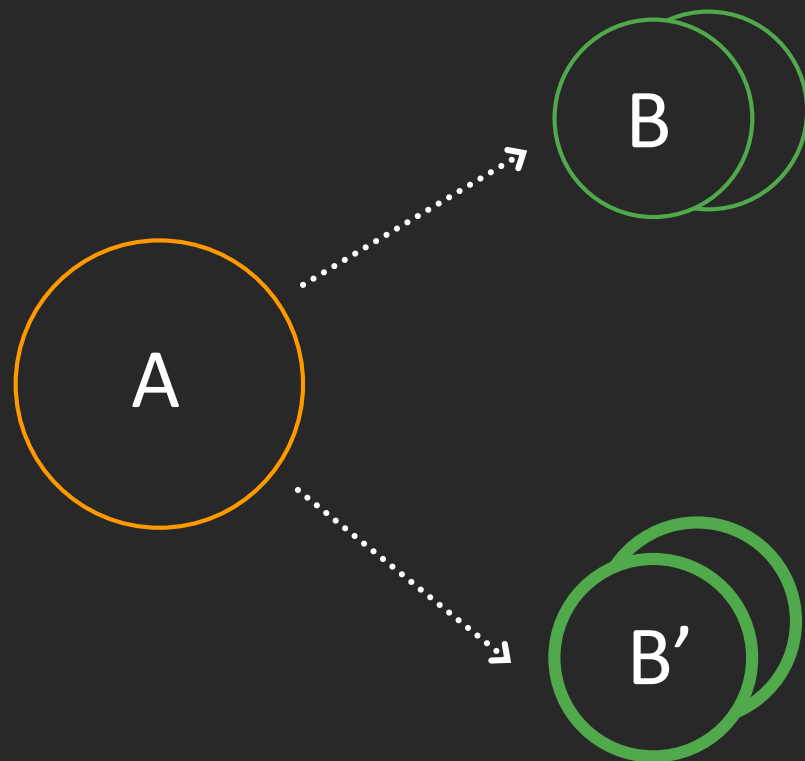


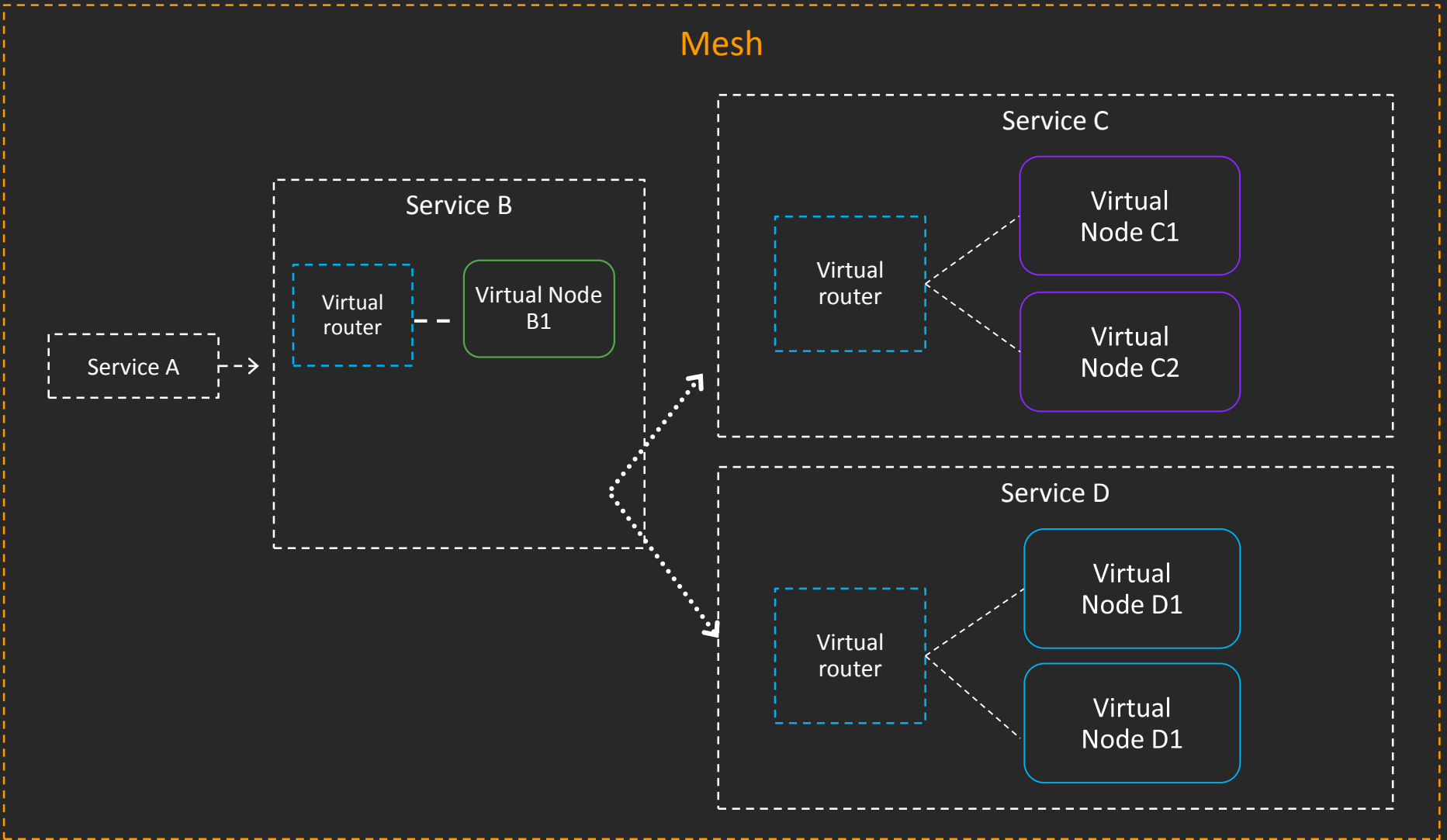
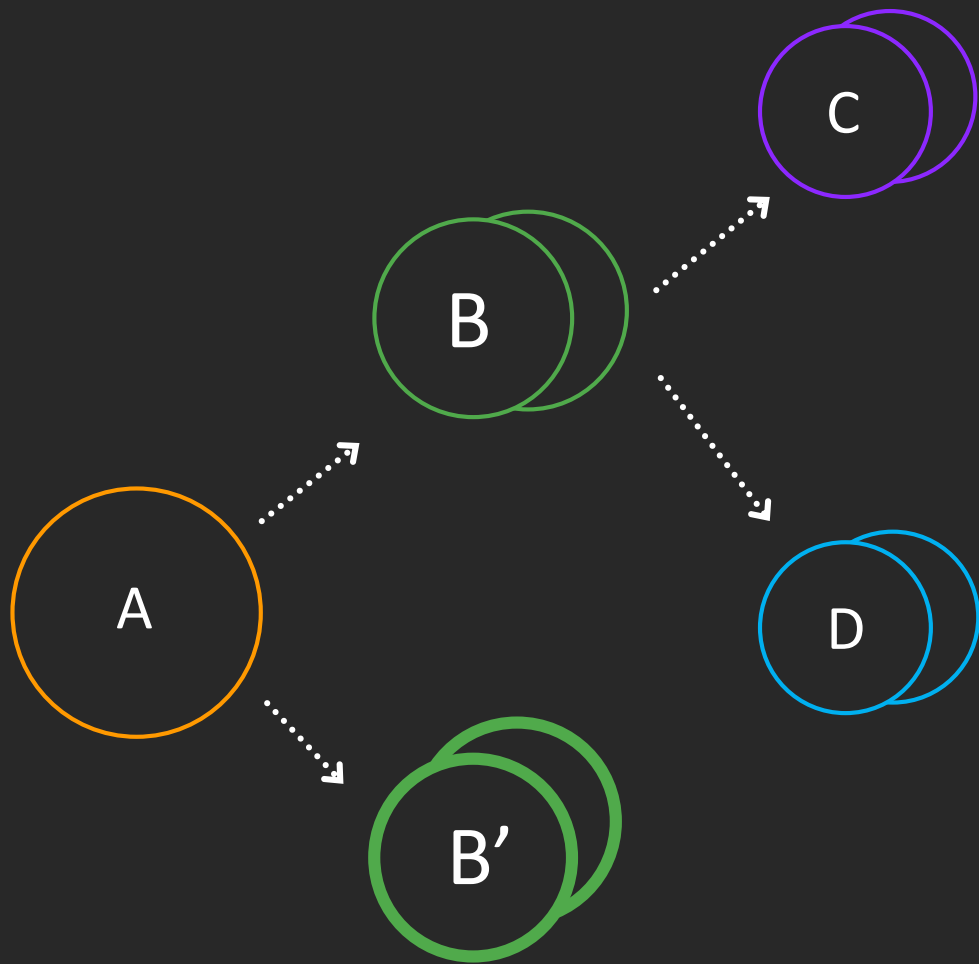
Virtual routes

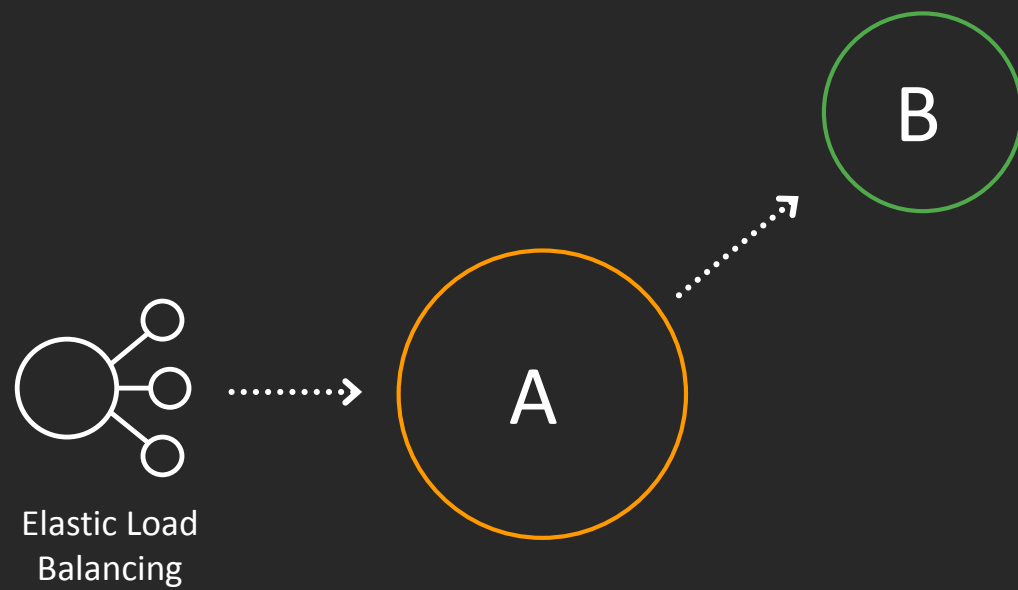


Update routes









Microservices

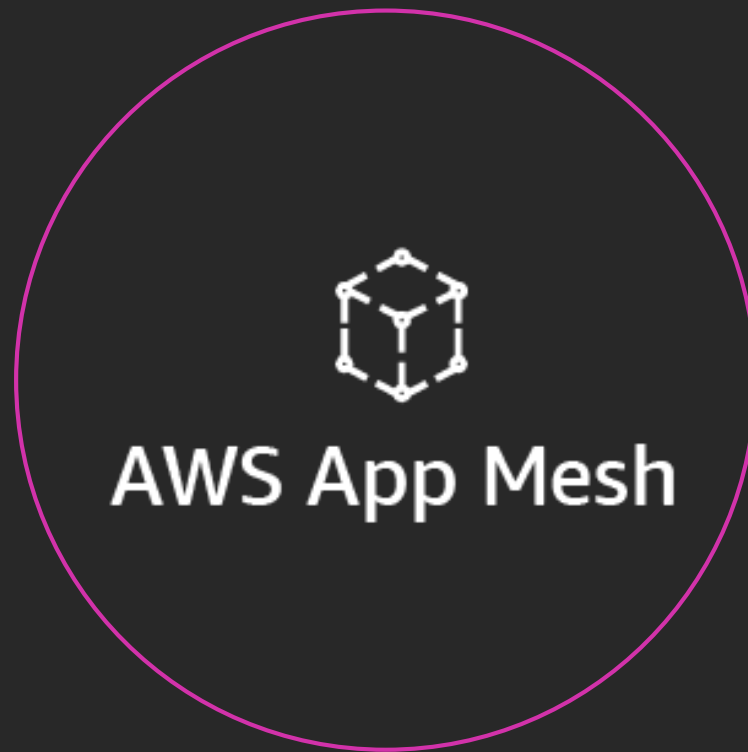
Mesh—myapp

Envoy Bootstrap:

```
//“tracing”: {...},  
//“stats”: {...},  
//“logging”: {...}
```

Demo

Getting started



Product overview

<https://aws.amazon.com/app-mesh>

Documentation

<https://docs.aws.amazon.com/app-mesh/index.html>

Examples

<https://github.com/aws-labs/aws-app-mesh-examples>

Issues & Roadmap

<https://github.com/aws-labs/aws-app-mesh-examples/issues>

Thank you!

Shubha Rao
Sr. Product Manager
AWS

Tony Pujals
Sr. Developer Advocate
AWS



Please complete the session
survey in the mobile app.