Machine Intelligence

Modern Infrastructure

http://mi2.live

Getting Started with AWS App Mesh

What is MI2?

MI2 Webinars focus on the convergence of **machine intelligence** and **modern infrastructure**. Every alternate week, I deliver informative and insightful sessions covering cutting-edge technologies. Each webinar is complemented by a tutorial, code snippets, and a video.

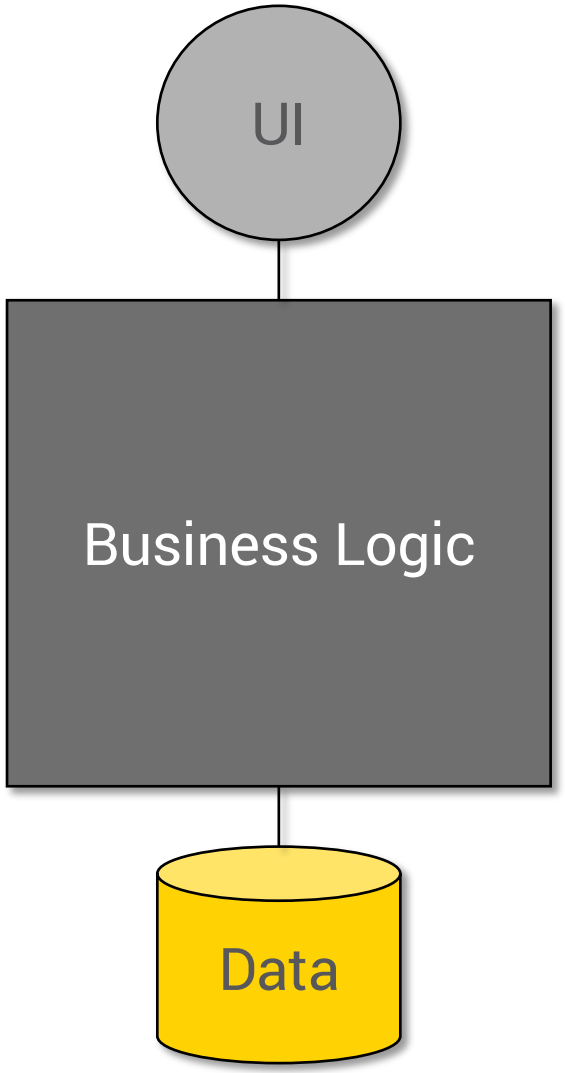
MI2 strives to be an independent and neutral platform for exploring emerging technologies.

Register at http://mi2.live

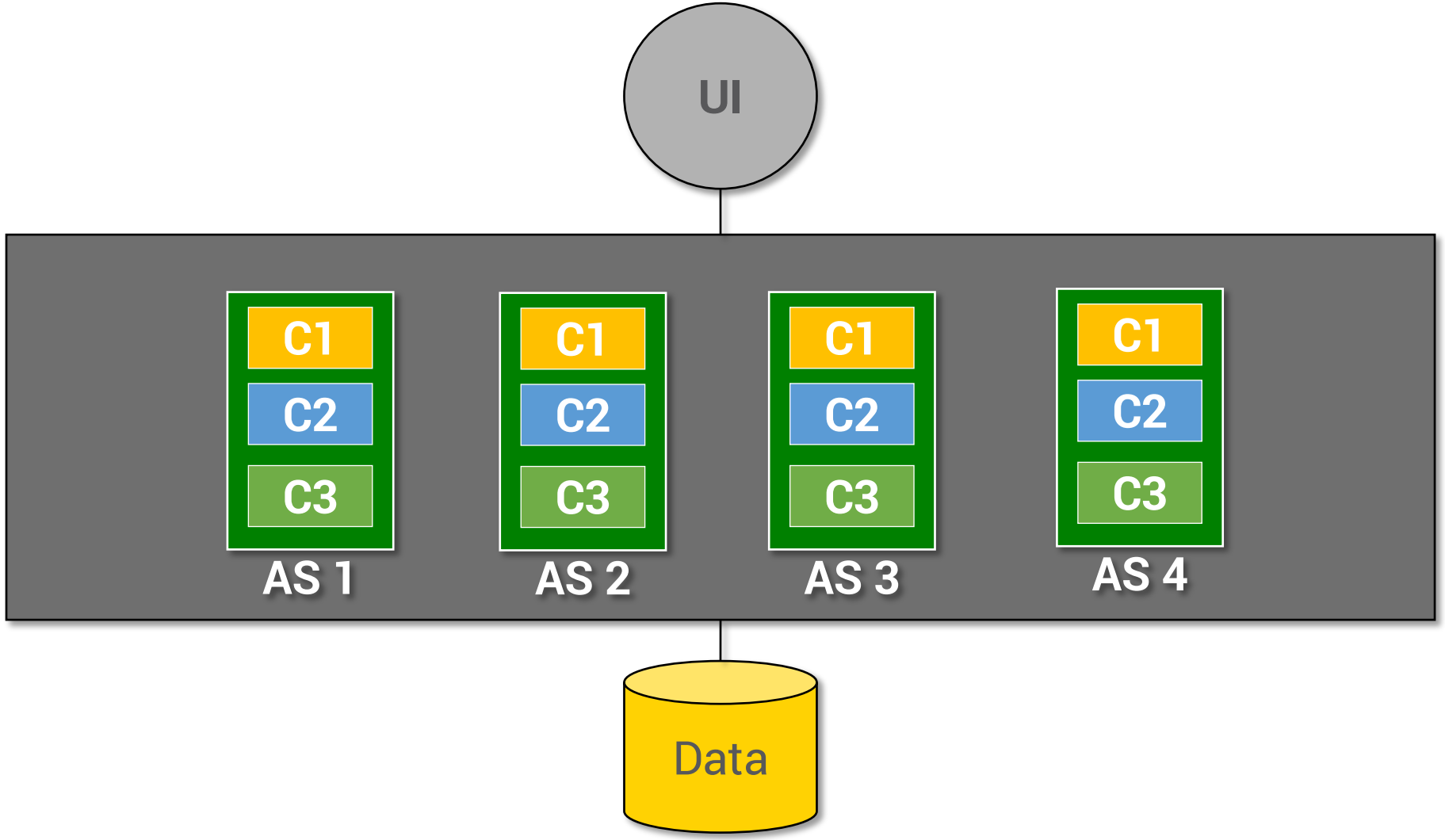
Objectives

- Overview of Microservices
- Challenges involved with Microservices
- Why use a Service Mesh?
- Big Picture of AWS App Mesh
- Demo
- Summary

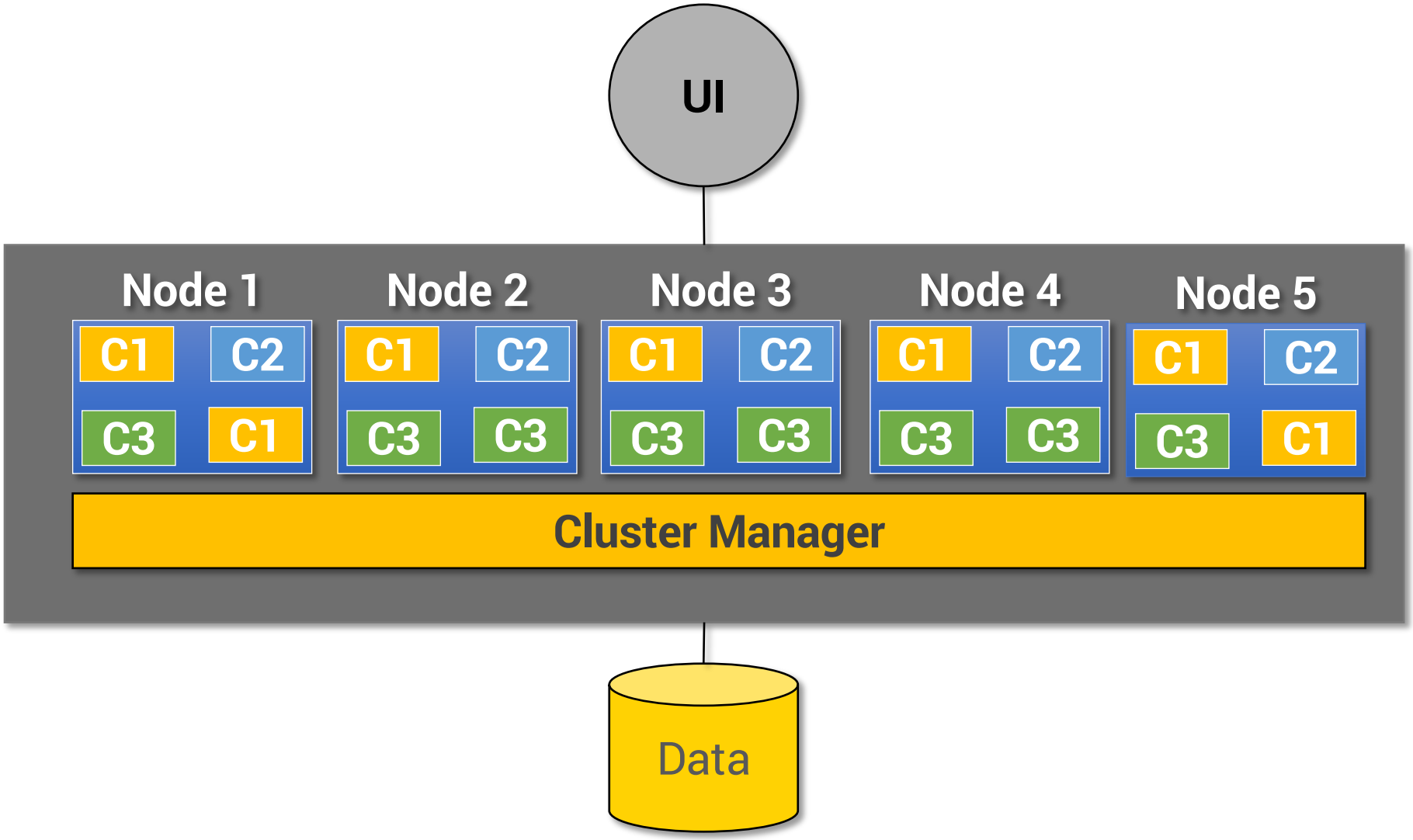
Traditional 3 Tier Architecture



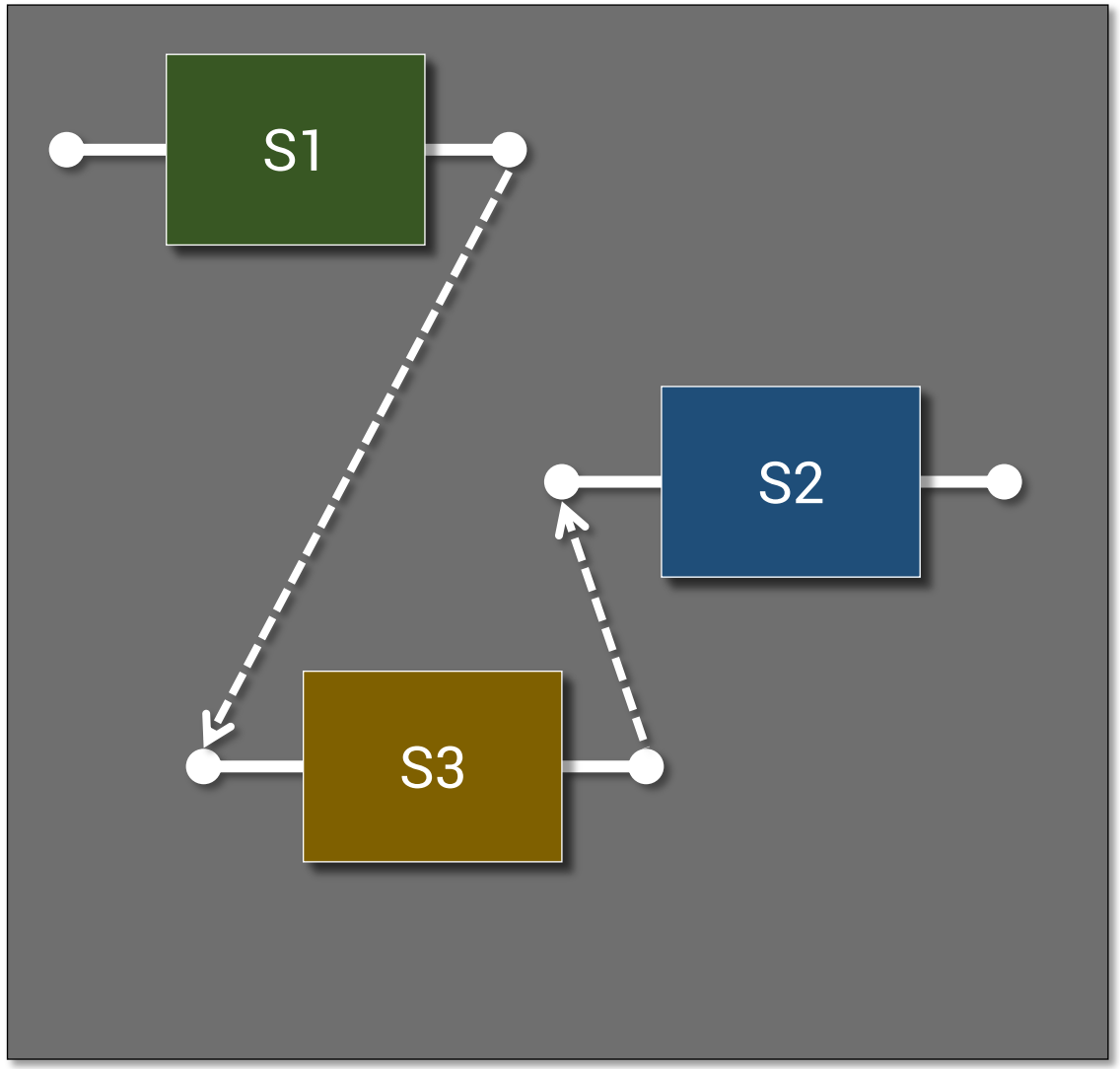
Traditional 3 Tier Architecture



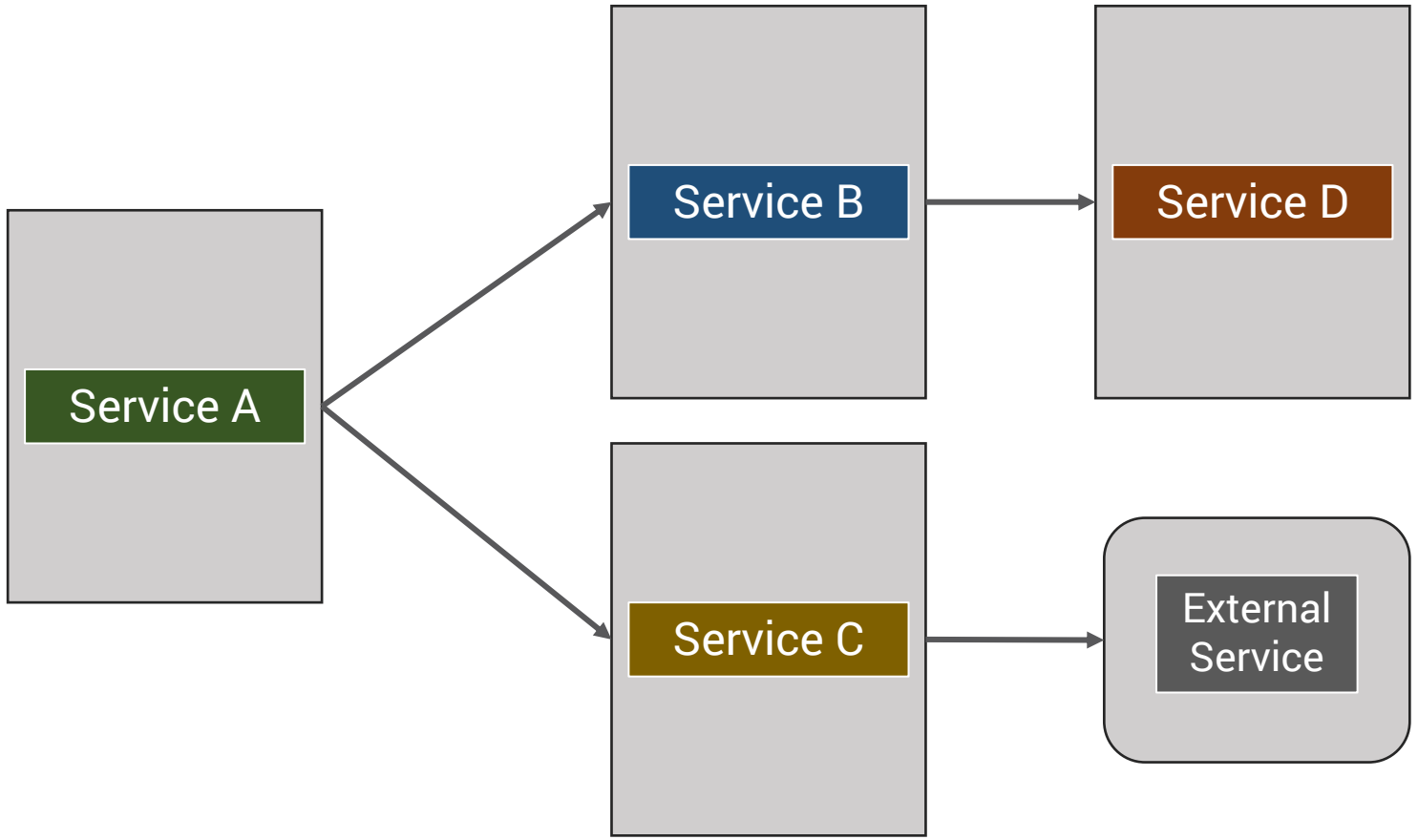
Microservices Architecture



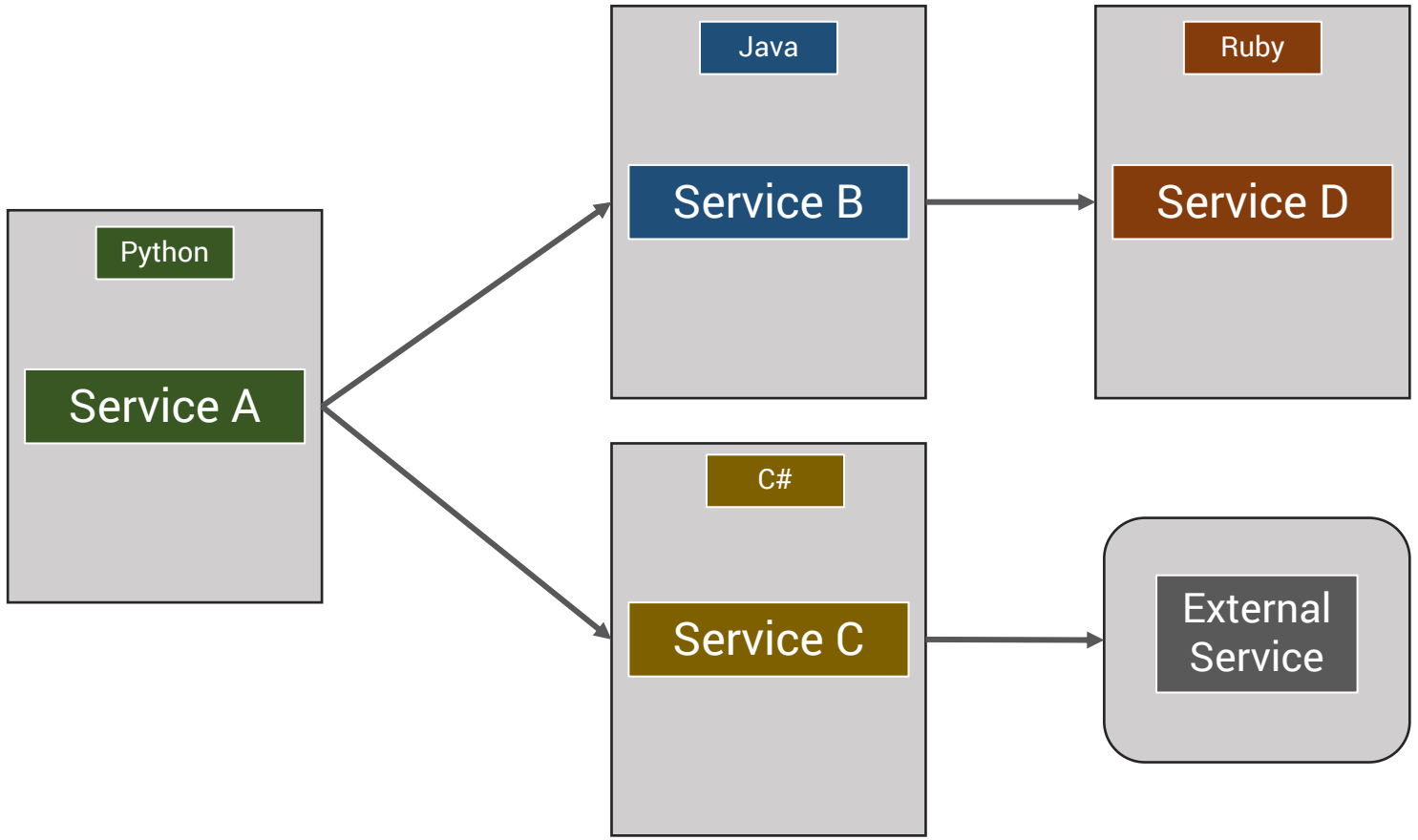
Microservices Architecture



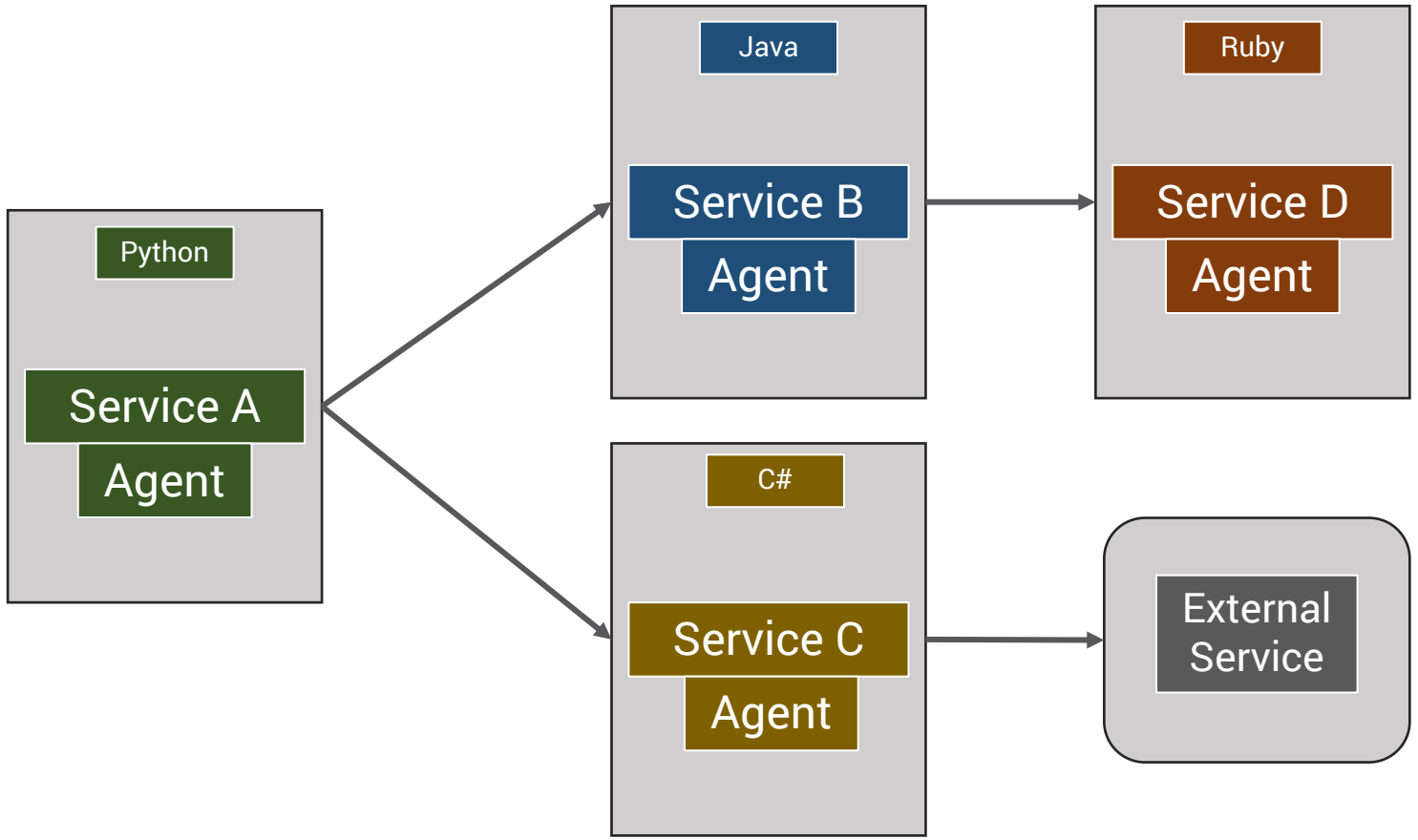
Challenges involved with Microservices



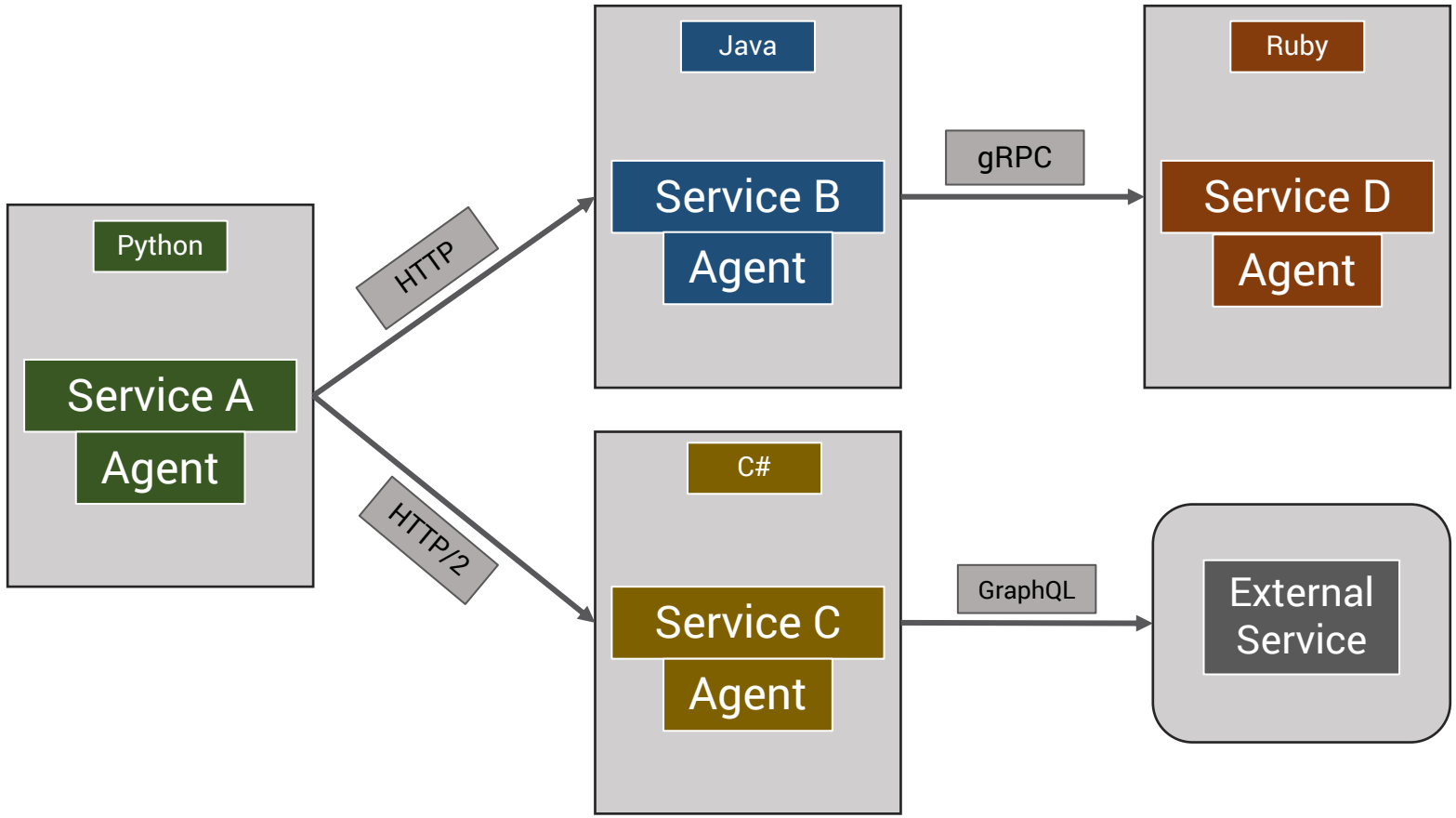
Challenges involved with Microservices



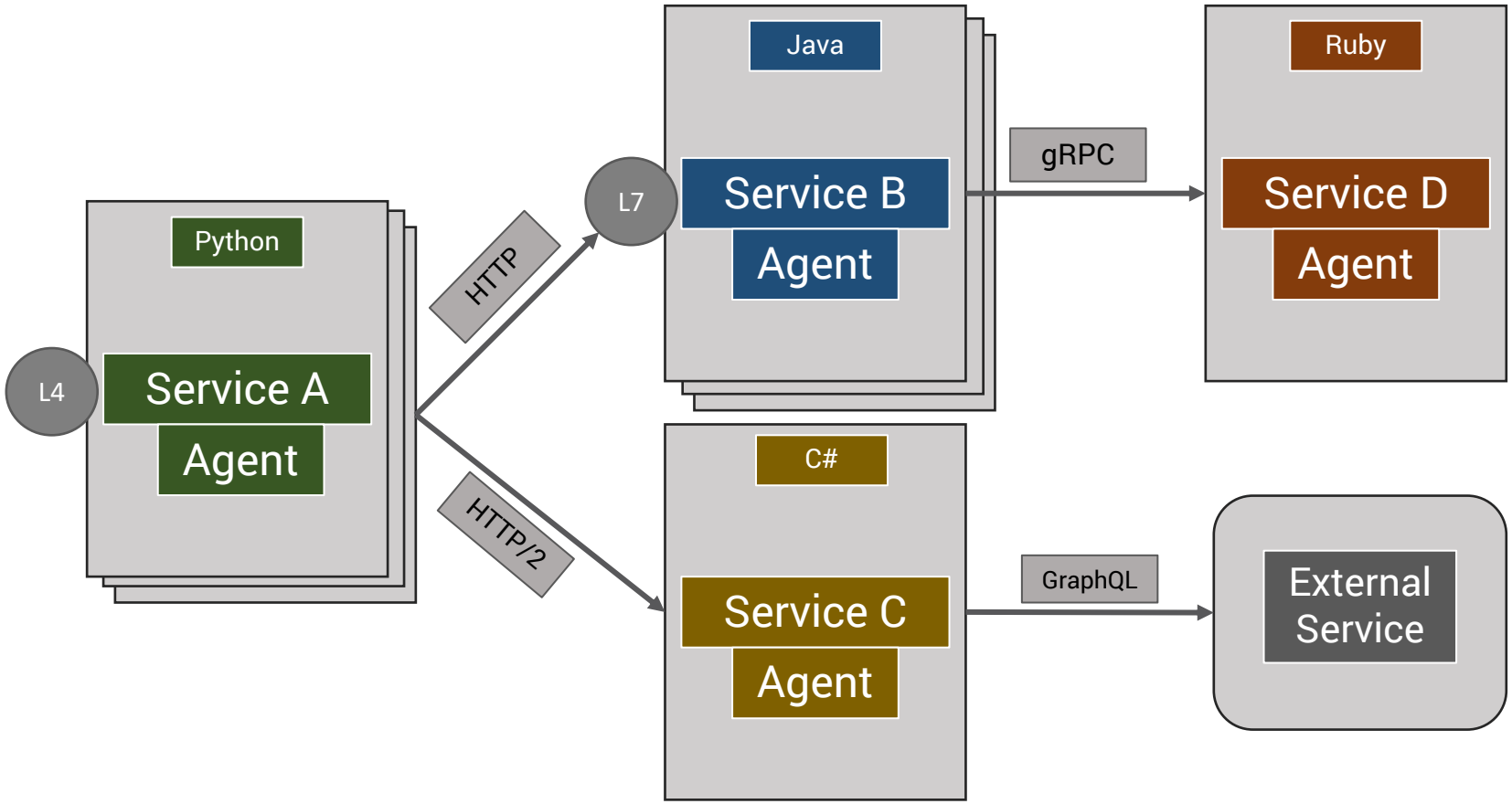
Challenges involved with Microservices



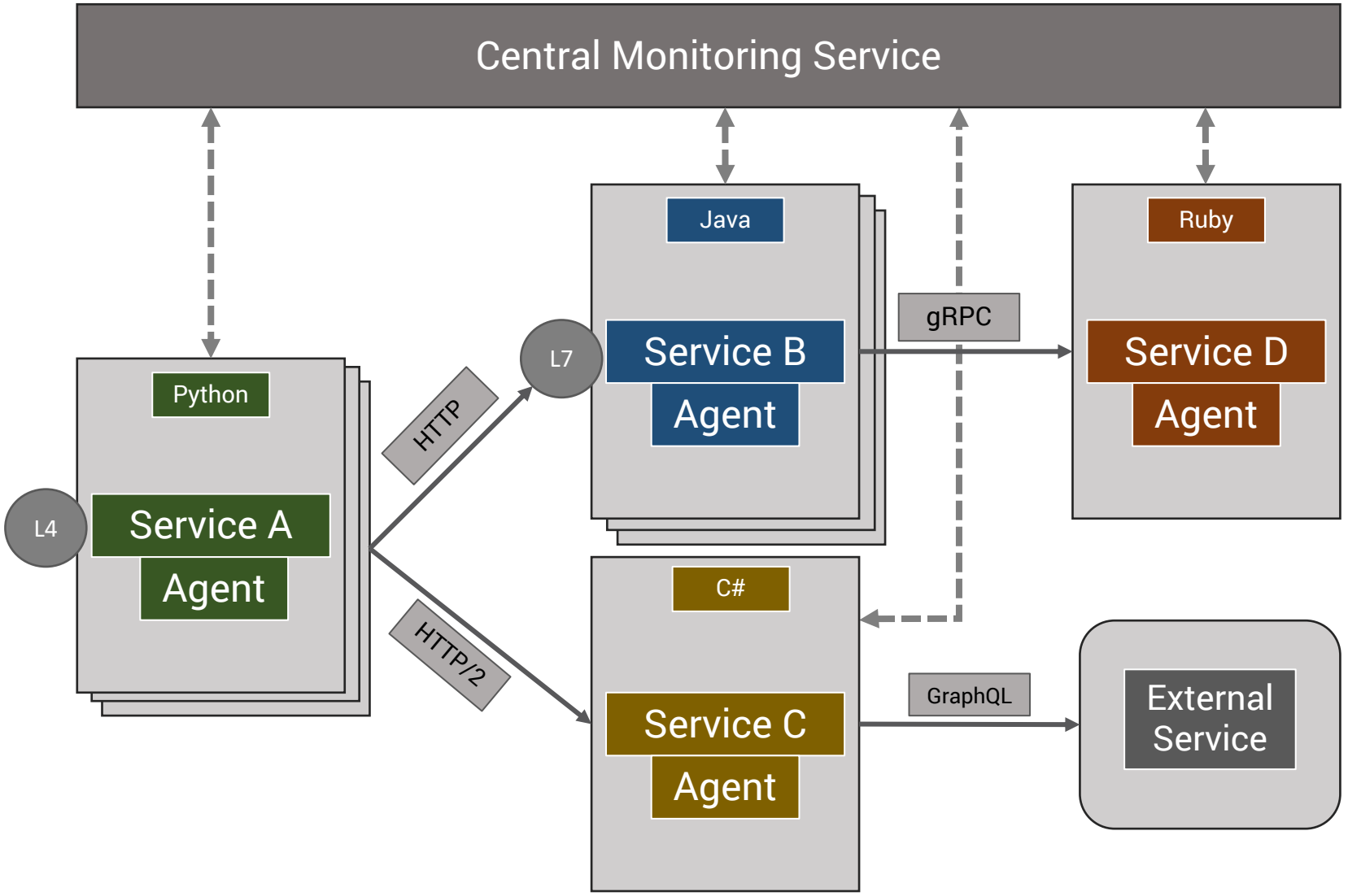
Challenges involved with Microservices



Challenges involved with Microservices



Challenges involved with Microservices



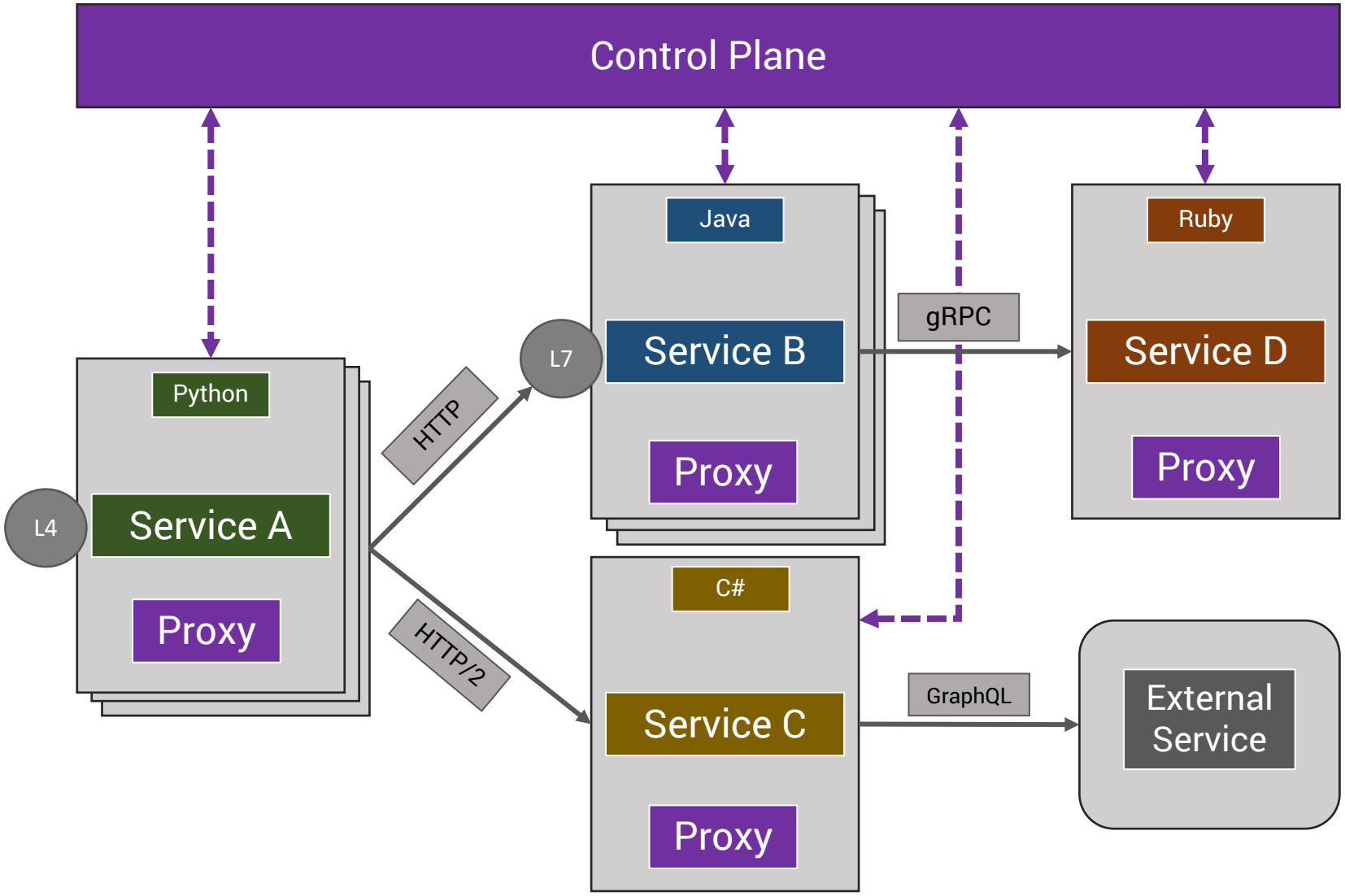
Challenges involved with Microservices

- Polyglot programming model
- Language-specific tools and SDKs
- Diverse set of protocols
- Multiple deployment targets
- Combination of load balancers
- Complex debugging
- Lack of visibility and observability
- Hard to implement the best practices of distributed computing
- Decentralized authentication and authorization

Service Mesh - A Possible Solution

- Out of process architecture
- Clean separation of data plane and control plane
- Support internal and external load balancing (L3/L4/L7)
- Consistent Service discovery
- Extensible protocol support
- Advanced health checks
- Real-time monitoring, logging, tracing
- Best practices of distributed computing

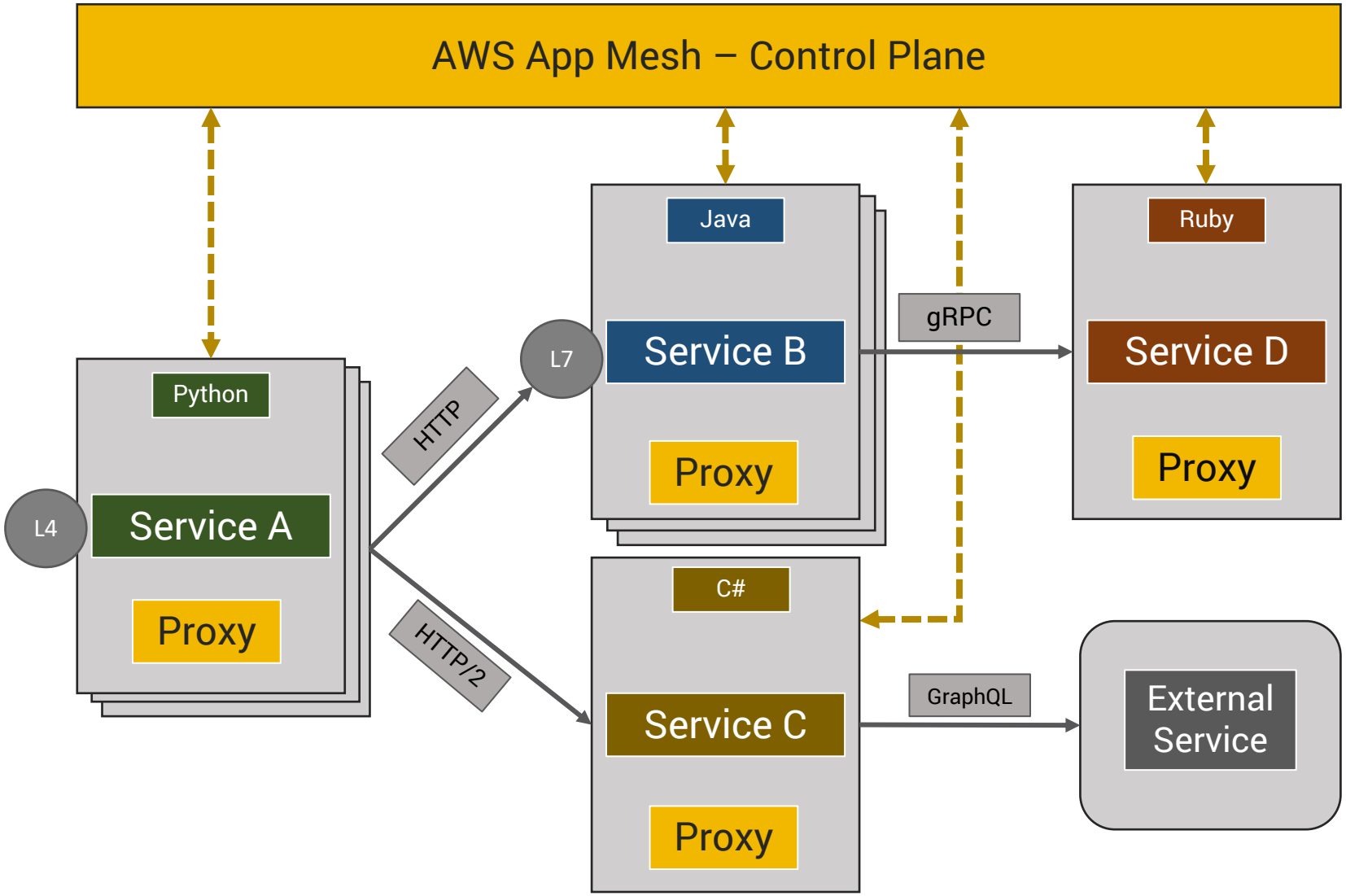
Service Mesh – Control Plane vs Data Plane



What is AWS App Mesh?

- Managed service to monitor microservices running in AWS
- Data plane is a combination of Envoy proxy and a router
- Control plane is managed by AWS – exposed via CLI, API, Portal
- Observability for microservices
- Policy-driven communication and traffic routing
- Scalable and available control plane
- Tight integration with AWS services

AWS App Mesh – Control Plane vs. Data Plane



Key Concepts of AWS App Mesh

Service Mesh

Logical boundary for network traffic between the services that reside within it

Virtual Node

Logical pointer to a particular task group, such as an ECS service or a Kubernetes deployment

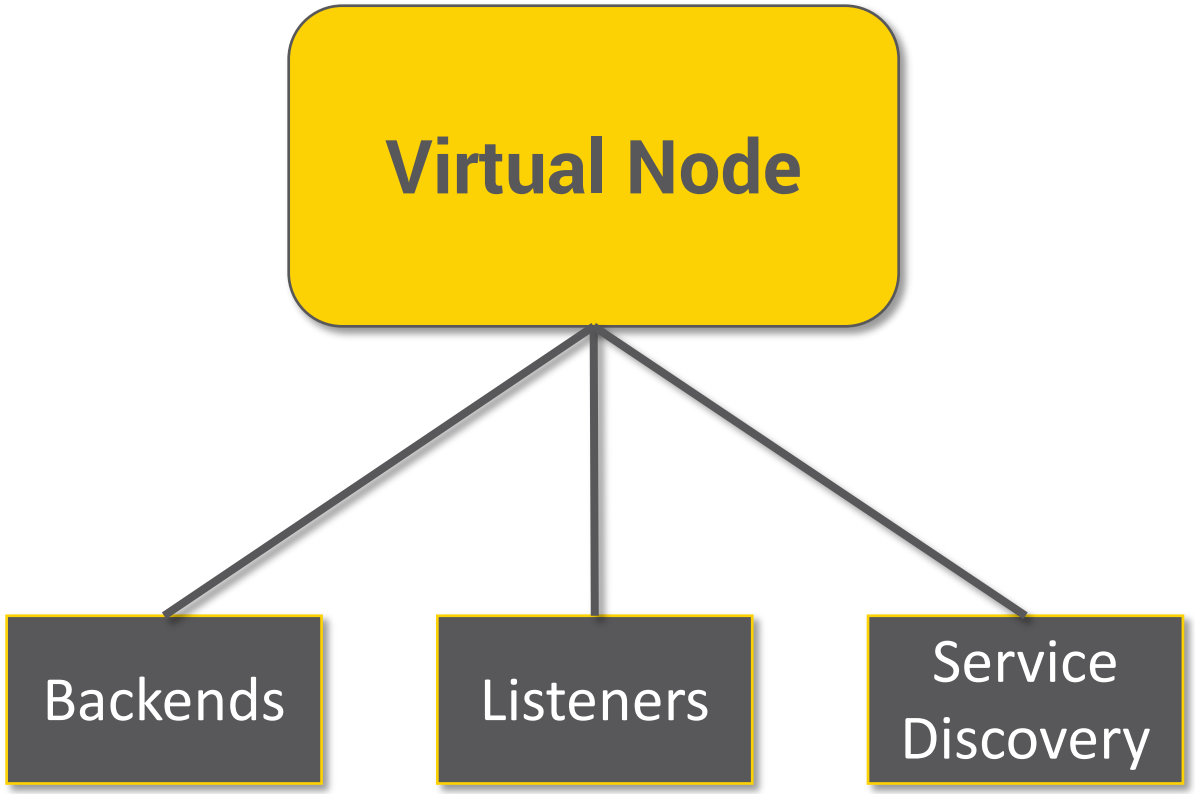
Virtual Router

Handles traffic for one or more service names within your mesh

Virtual Route

Associated with a virtual router to direct traffic to one or more virtual nodes

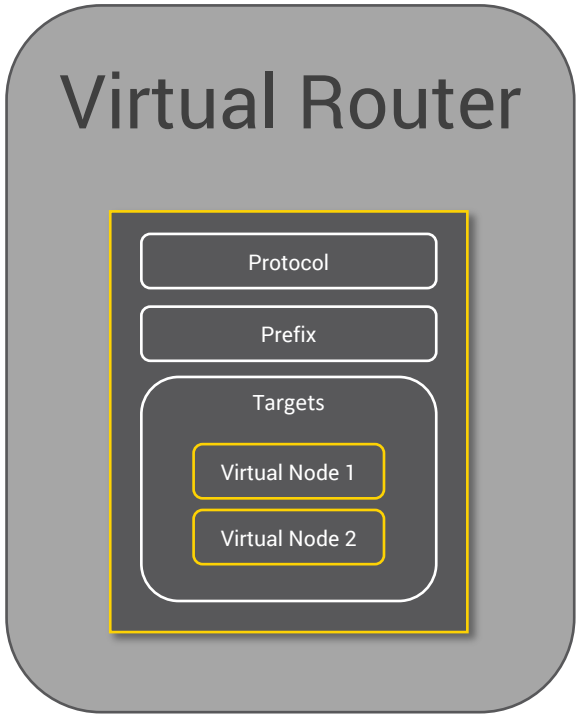
Virtual Node



order-vn.json

```
{
  "meshName": "shopmesh",

  "virtualNodeName": "order-vn",

  "spec": {

    "backends": [
      "product.default.svc.cluster.local",
      "customer.default.svc.cluster.local"
    ],

    "listeners": [
      {
        "portMapping": {
          "port": 5000,
          "protocol": "http"
        }
      }
    ],

    "serviceDiscovery": {
      "dns": {
        "serviceName": "order.default.svc.cluster.local"
      }

    }
  }
}
```

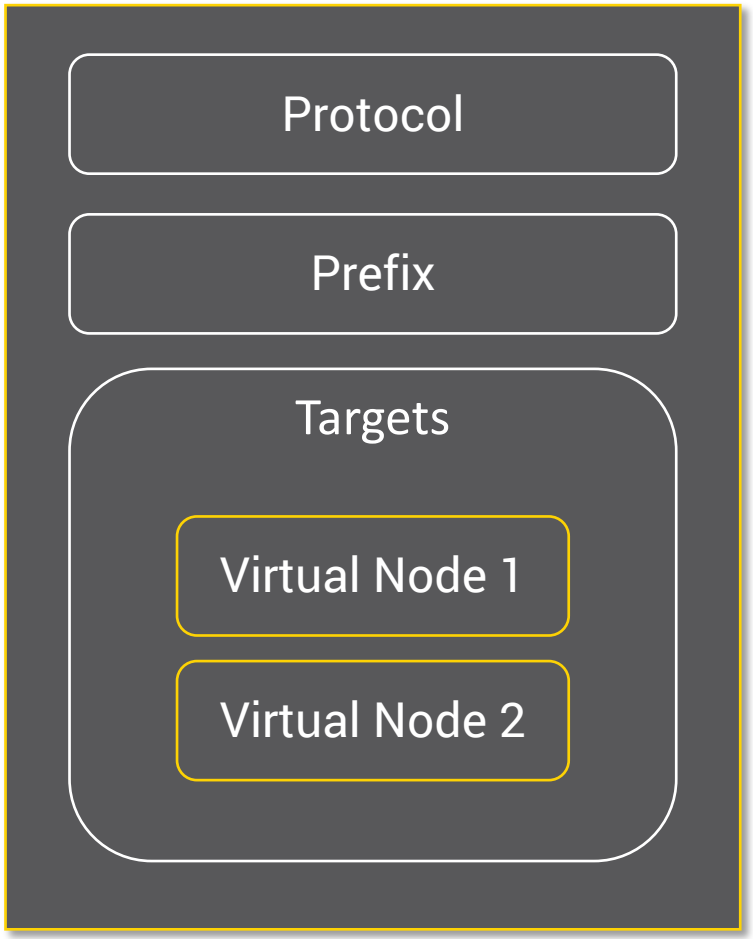
Virtual Router



order-vr.json

```
{
  "meshName": "shopmesh",

  "virtualRouterName": "order-vr",

  "spec": {
    "serviceNames": [
      "order.default.svc.cluster.local"
    ]
  }
}
```

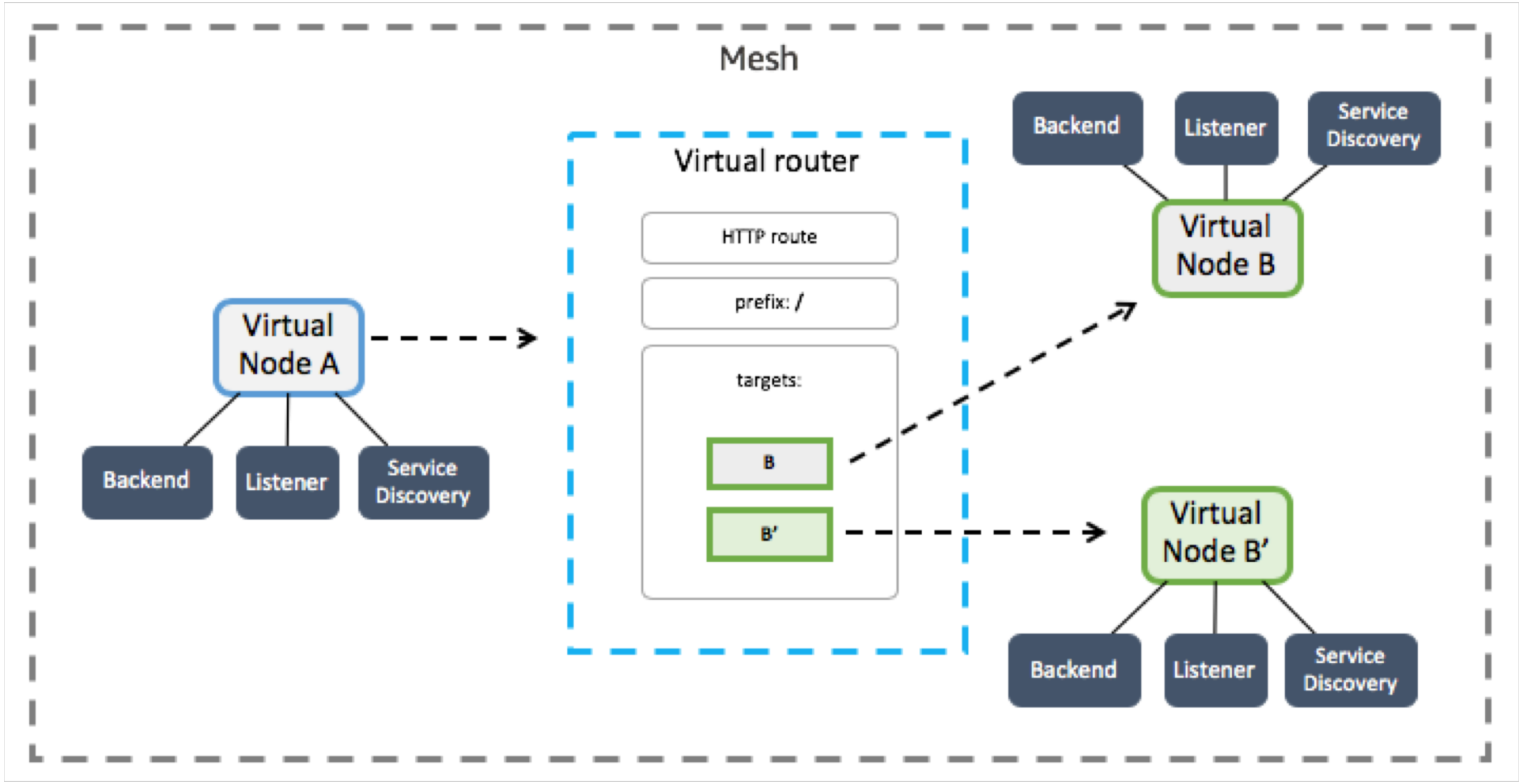
Virtual Route



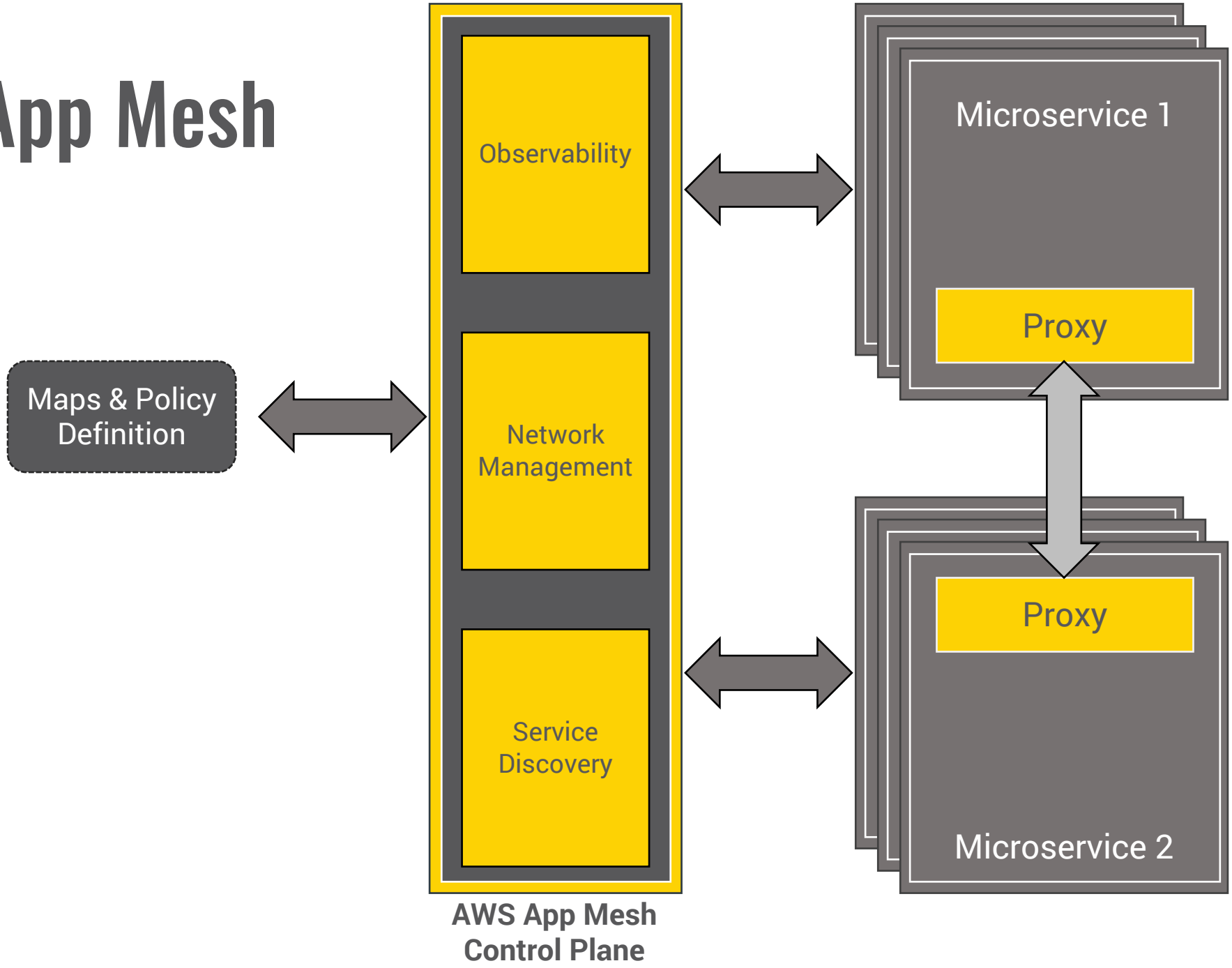
order-r.json

```
{
  "meshName": "shopmesh",

  "virtualRouterName": "order-vr",

  "routeName": "order-r",

  "spec": {
    "httpRoute": {
      "action": {
        "weightedTargets": [
          {
            "virtualNode": "order-vn",
            "weight": 100
          }
        ]
      },
      "match": {
        "prefix": "/"
      }
    }
  }
}
```

AWS App Mesh – The Big Picture



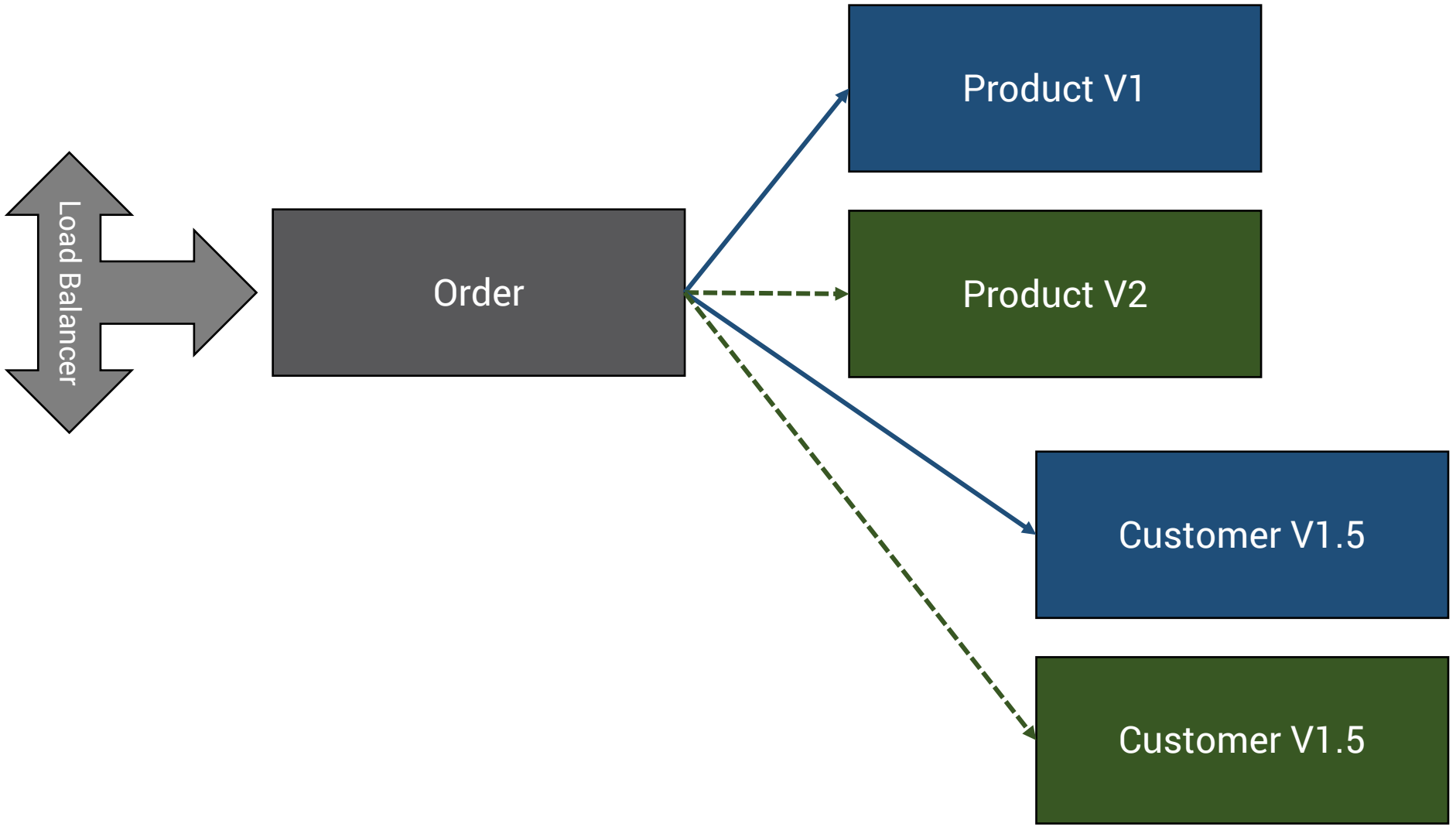
AWS App Mesh


Observability
Network Management
Service Discovery
AWS App Mesh Control Plane
Maps & Policy Definition
Microservice 1
Proxy
Proxy
Microservice 2

What are we building?

- Three microservices connected via AWS App Mesh
  - Order
  - Product
  - Customer
- Each service is a Kubernetes deployment
- Order is exposed via ELB
- Canary releases of Product and Customer services
  - Divert 50% of the traffic to Product v2
  - Divert 25% of the traffic to Customer v1.5
- Maintain zero downtime!

Demo Scenario



DEMO

Performing Canary Releases with App Mesh

https://github.com/janakiramm/app-mesh-tutorial

Summary

- App Mesh is AWS' implementation of Service Mesh
- Each microservice runs an Envoy proxy
- App Mesh control plane governs the traffic policies
- Policies can be updated via CLI, SDK, or Console
- Observability is available through CloudWatch and 3rd parties

MI2 Sponsors

Next Webinar

Building Convolutional Neural Networks with NVIDIA DIGITS

NVIDIA DIGITS is a wizard-style framework to build convolutional neural networks. Attend this session to learn how to setup and configure NVIDIA DIGITS for building deep neural networks based on popular frameworks such as Caffe and TensorFlow. In this webinar, I will walk you through everything from data preparation to hyperparameter tuning to model serving.

Thursday, February 7th, 2019
9:00 AM PST / 10:30 PM IST

Register at http://mi2.live