Machine Intelligence

Modern Infrastructure

http://mi2.live

Everything you want to know about Istio

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

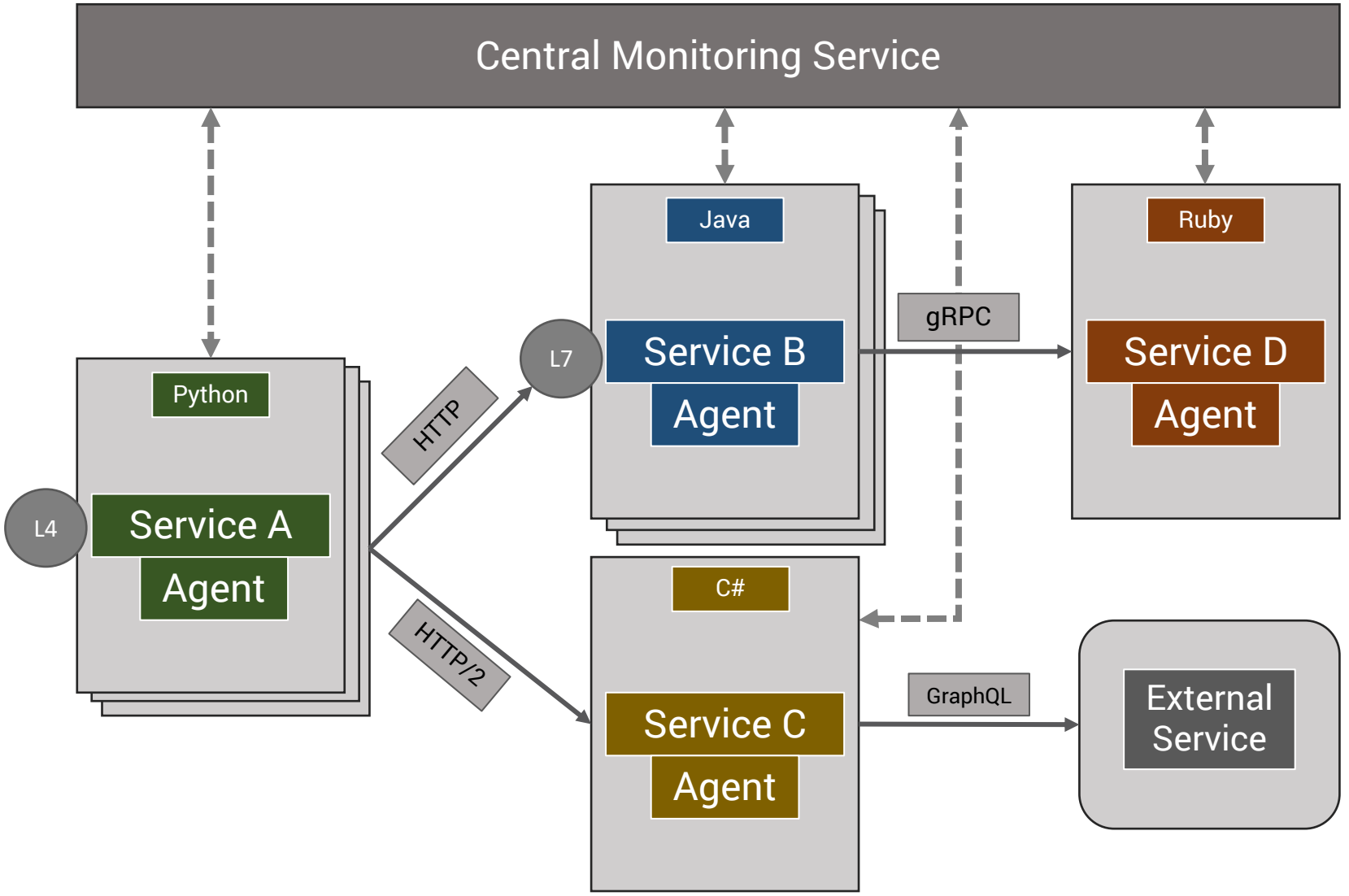
Objectives

- Overview of service mesh
- Motivation to use Istio
- Istio architecture
- Demo
- Summary

Challenges with Microservices

- Based on polyglot development
- Highly distributed
- Difficult to debug
- Hard to implement logging and tracing
- Dynamic scale-in and scale-out
- Disparate protocols
- Implements internal and external load balancers

Challenges involved with Microservices



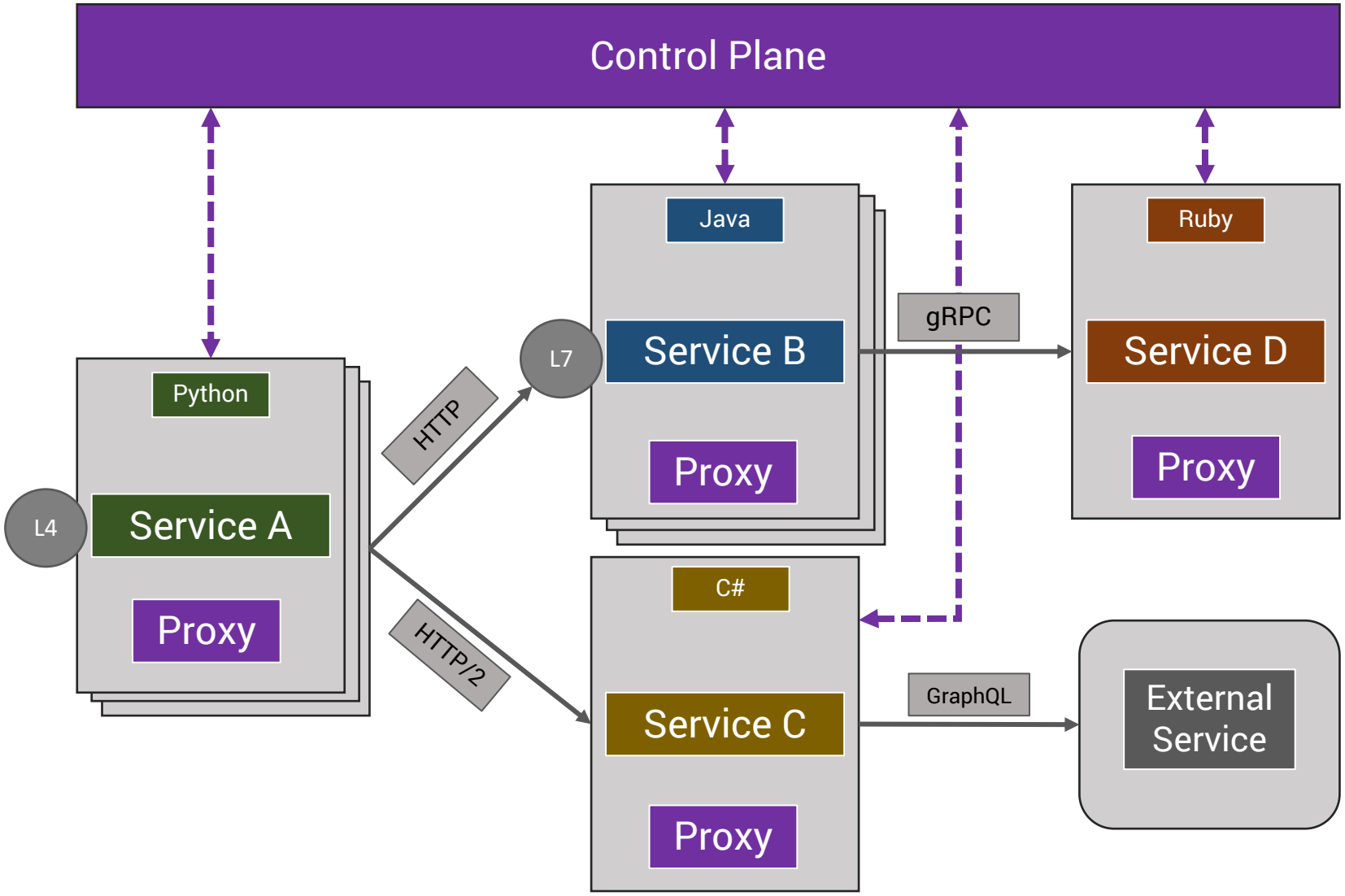
What is a Service Mesh?

- Plugs itself into the intra-service communication
- Intercepts east-west (even north-south) traffic
- Captures telemetry related to services and traffic
- Adds an implicit security layer
- Enables service discovery
- Implements policy-driven routing and traffic management
- Interfaces well with legacy and modern infrastructure

Why Service Mesh?

- 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

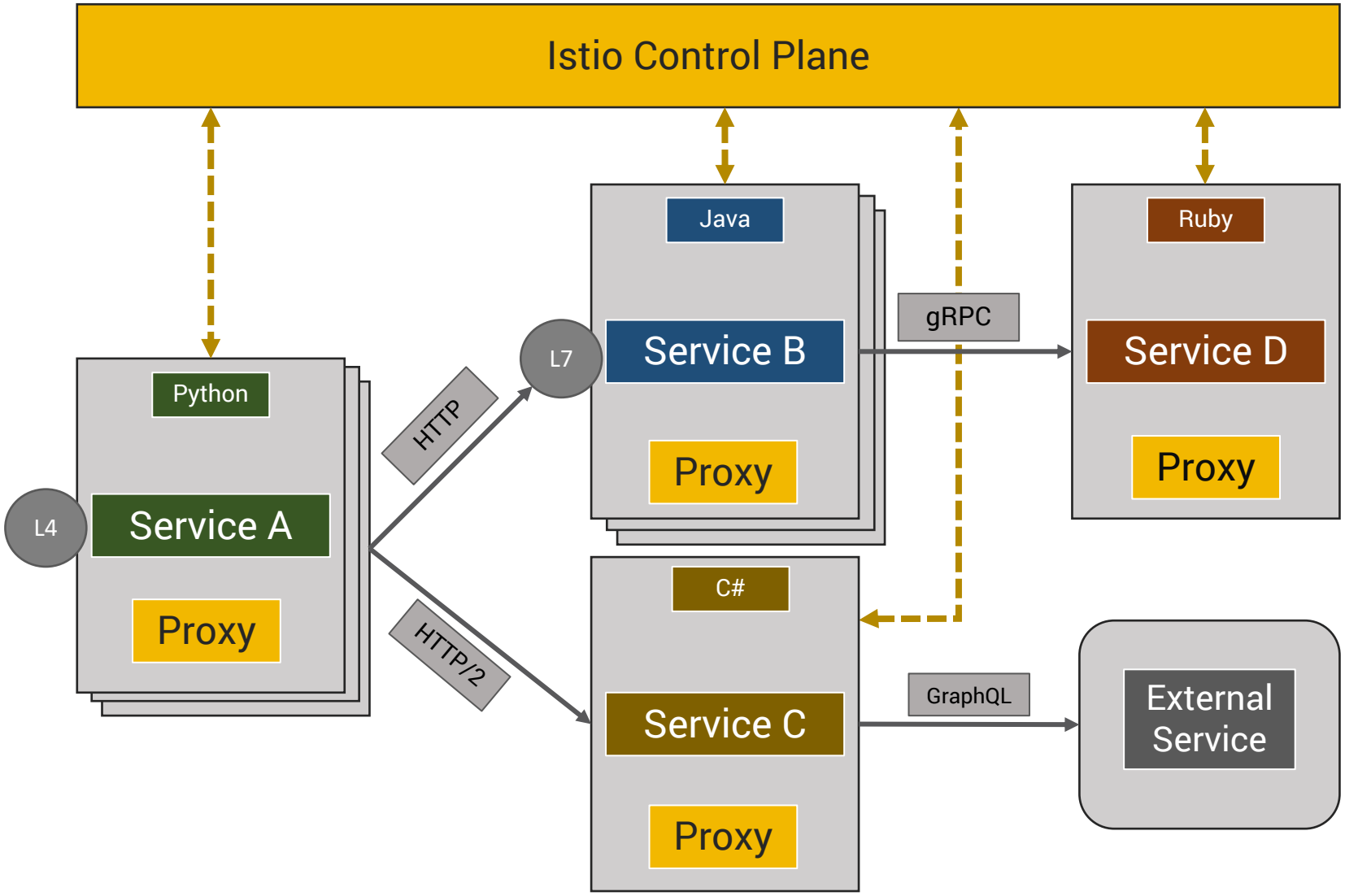
Service Mesh – Control Plane vs Data Plane



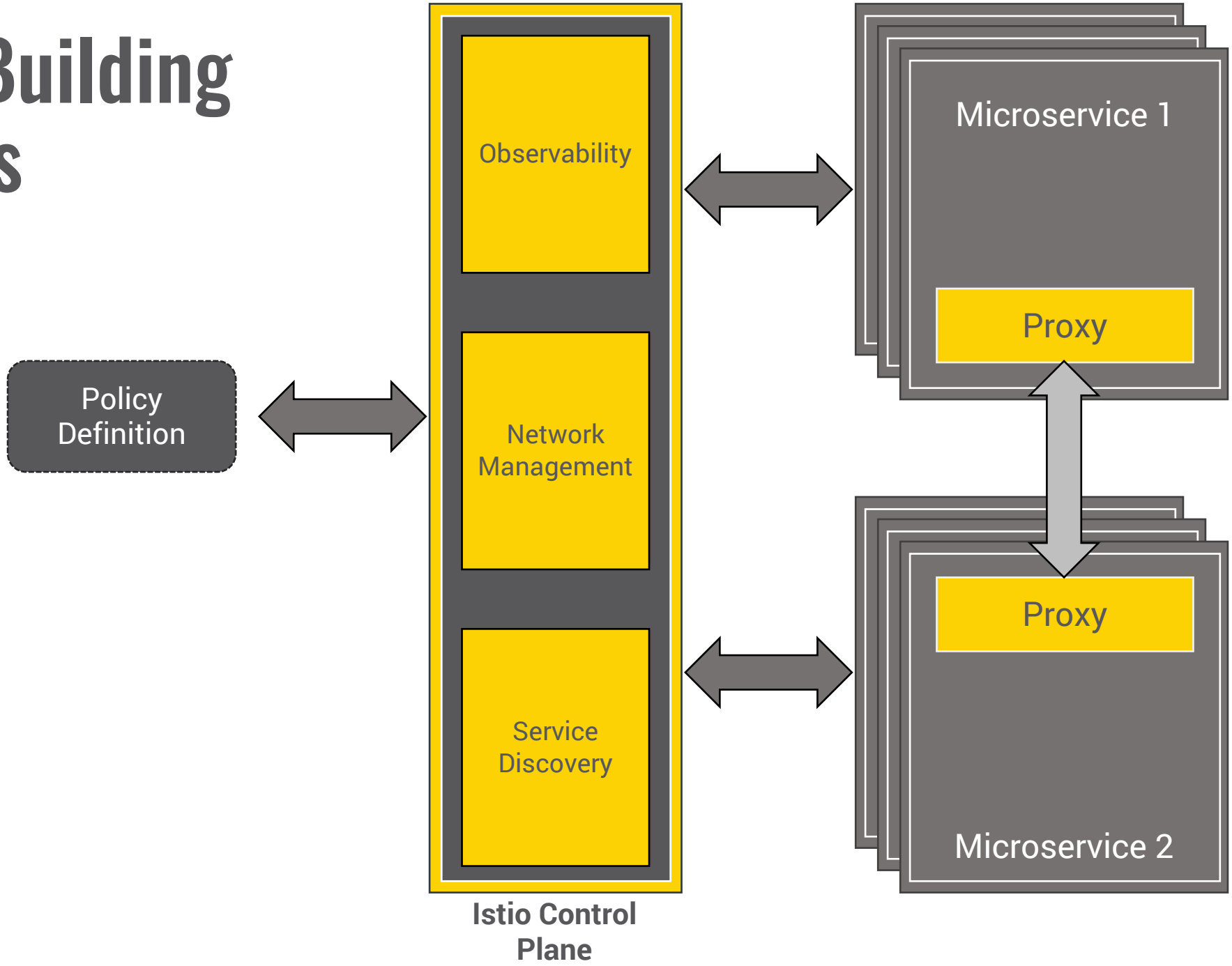
What is Istio?

- **Connect**
  - Intelligent traffic routing and flow
- **Secure**
  - Managed authentication, encryption
- **Control**
  - Enforce policy-driven communication across services
- **Observe**
  - Automatic tracing, monitoring, and logging

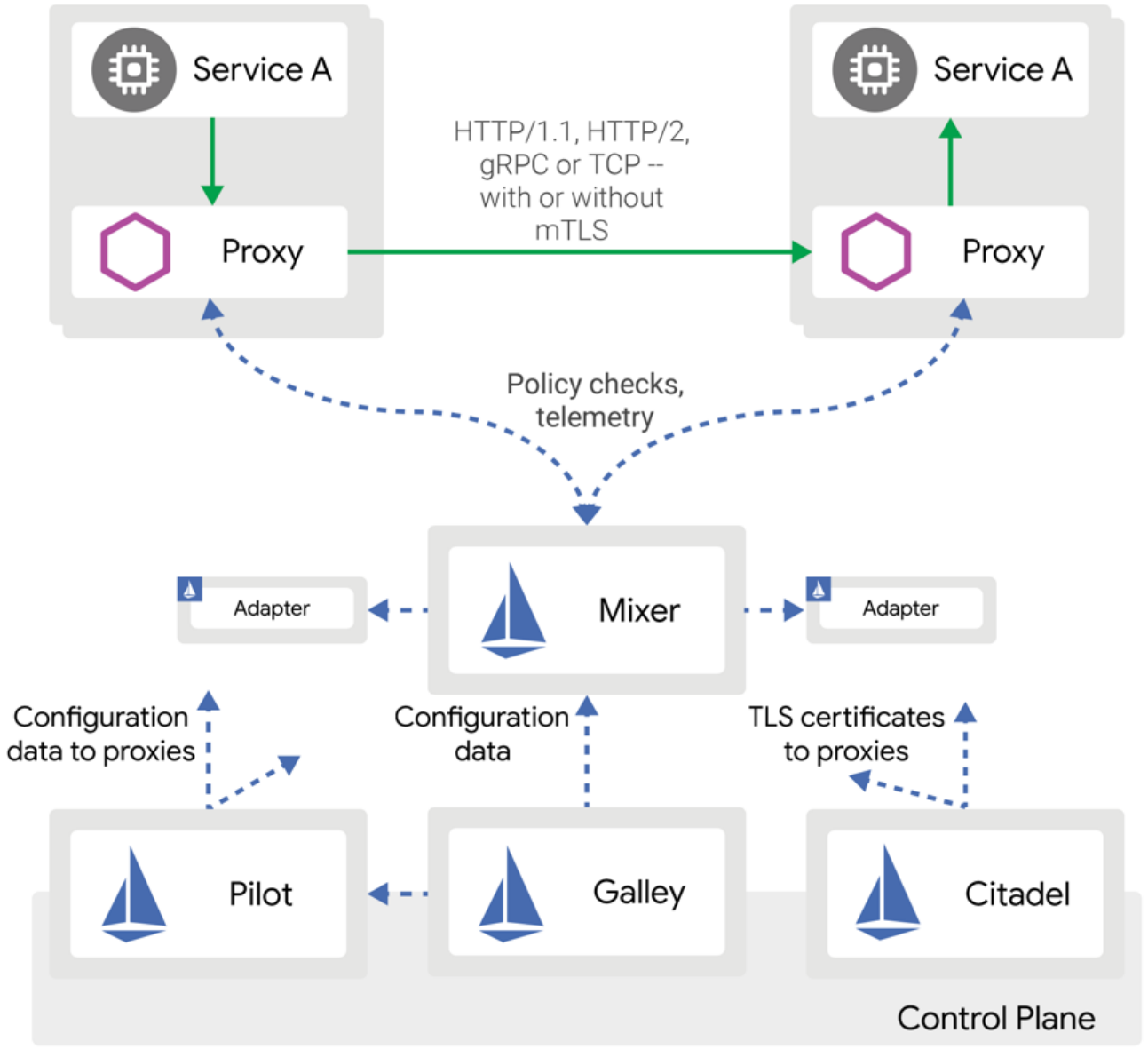
Istio – Control Plane vs. Data Plane



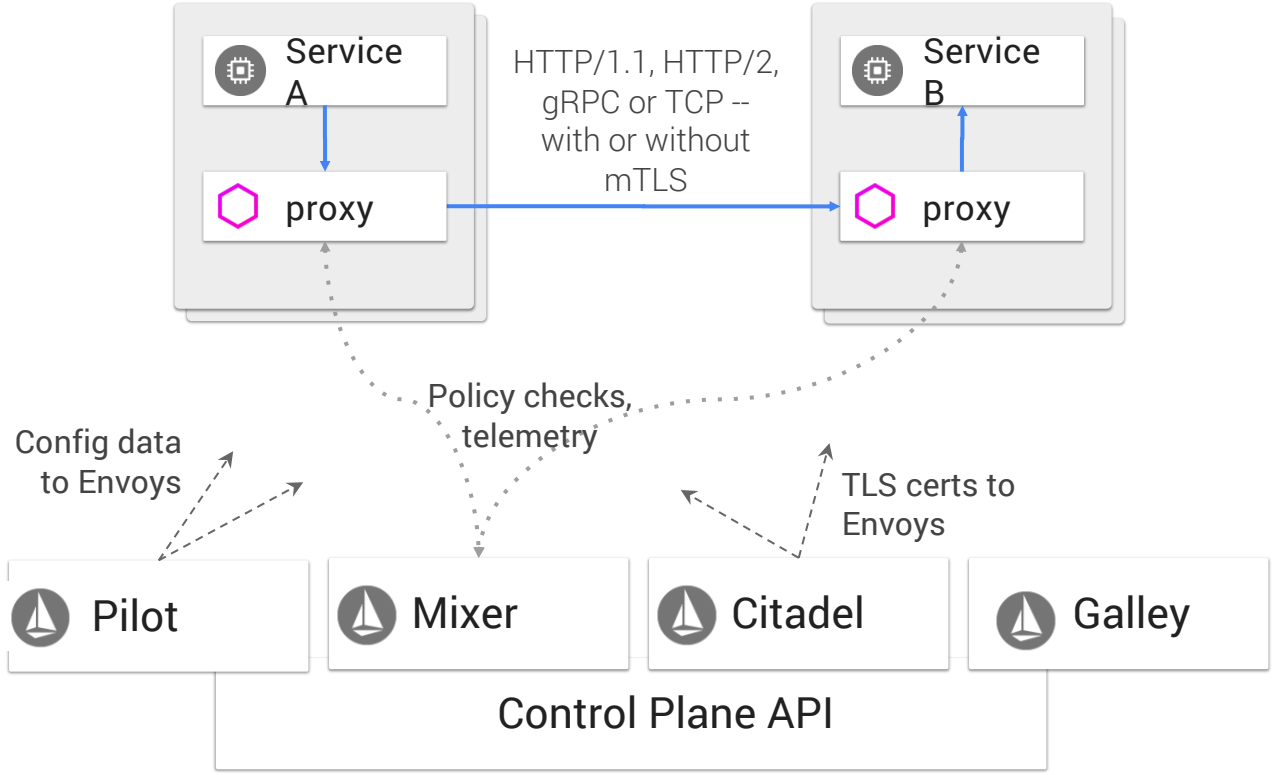
Istio Building Blocks



Istio Architecture



Istio Architecture



Pilot: Control plane to configure and push service communication policies.

Envoy: Network proxy to intercept communication and apply policies.

Mixer: Policy enforcement with a flexible plugin model for providers for a policy.

Citadel: Service-to-service auth[n,z] using mutual TLS, with built-in identity and credential management.

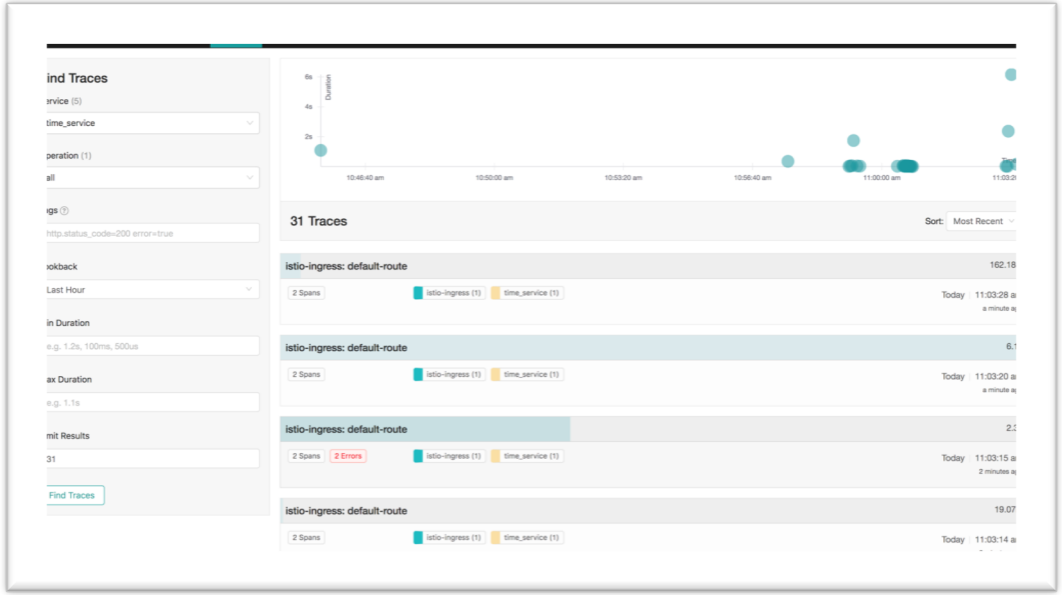
Galley: Configuration validation, distribution

Key Concepts of Istio Traffic Management

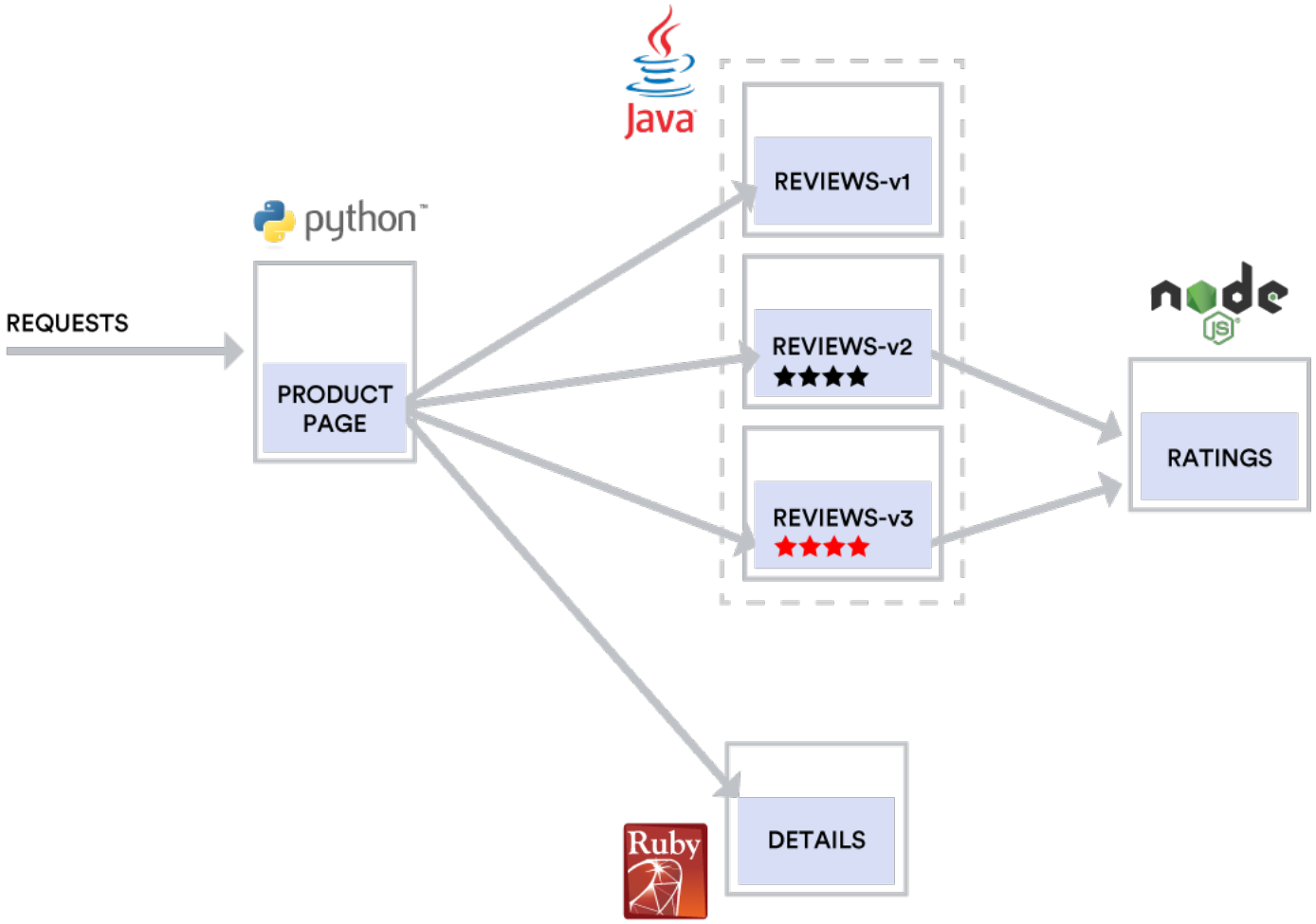
- **VirtualService** defines the rules that control how requests for a service are routed within an Istio service mesh.
- **DestinationRule** configures the set of policies to be applied to a request after VirtualService routing has occurred.
- **ServiceEntry** is commonly used to enable requests to services outside of an Istio service mesh.
- **Gateway** configures a load balancer for HTTP/TCP traffic operating at the edge of the mesh, most commonly to enable ingress traffic for an application.
- **Sidecar** configures one or more sidecar proxies attached to application workloads running inside the mesh.

Istio Telemetry

- Metrics
- Logs
- Tracing
- Visualization



What are we deploying?



DEMO

- Installing Istio
- Configuring Traffic Rules
- Visualizing Telemetry

Summary

- Automatic load balancing for HTTP, gRPC, WebSocket, and TCP traffic
- Fine-grained control of traffic behavior with rich routing rules, retries, failovers, and fault injection
- A pluggable policy layer and configuration API supporting access controls, rate limits and quotas
- Automatic metrics, logs, and traces for all traffic within a cluster, including cluster ingress and egress
- Secure service-to-service communication in a cluster with strong identity-based authentication and authorization

MI2 Sponsors

Next Webinar

Running Applications at the Edge with AWS Greengrass

AWS IoT Greengrass is software that lets you run local compute, messaging, data caching, sync, and ML inference capabilities on connected devices in a secure way. With AWS IoT Greengrass, connected devices can run AWS Lambda functions, execute predictions based on machine learning models, keep device data in sync, and communicate with other devices. This session will cover the fundamentals of AWS Greengrass.

Thursday, May 16th, 2019
9:00 AM PST / 9:30 PM IST

Register at http://mi2.live