ISTIO SERVICE MESH: PAST , PRESENT AND FUTURE

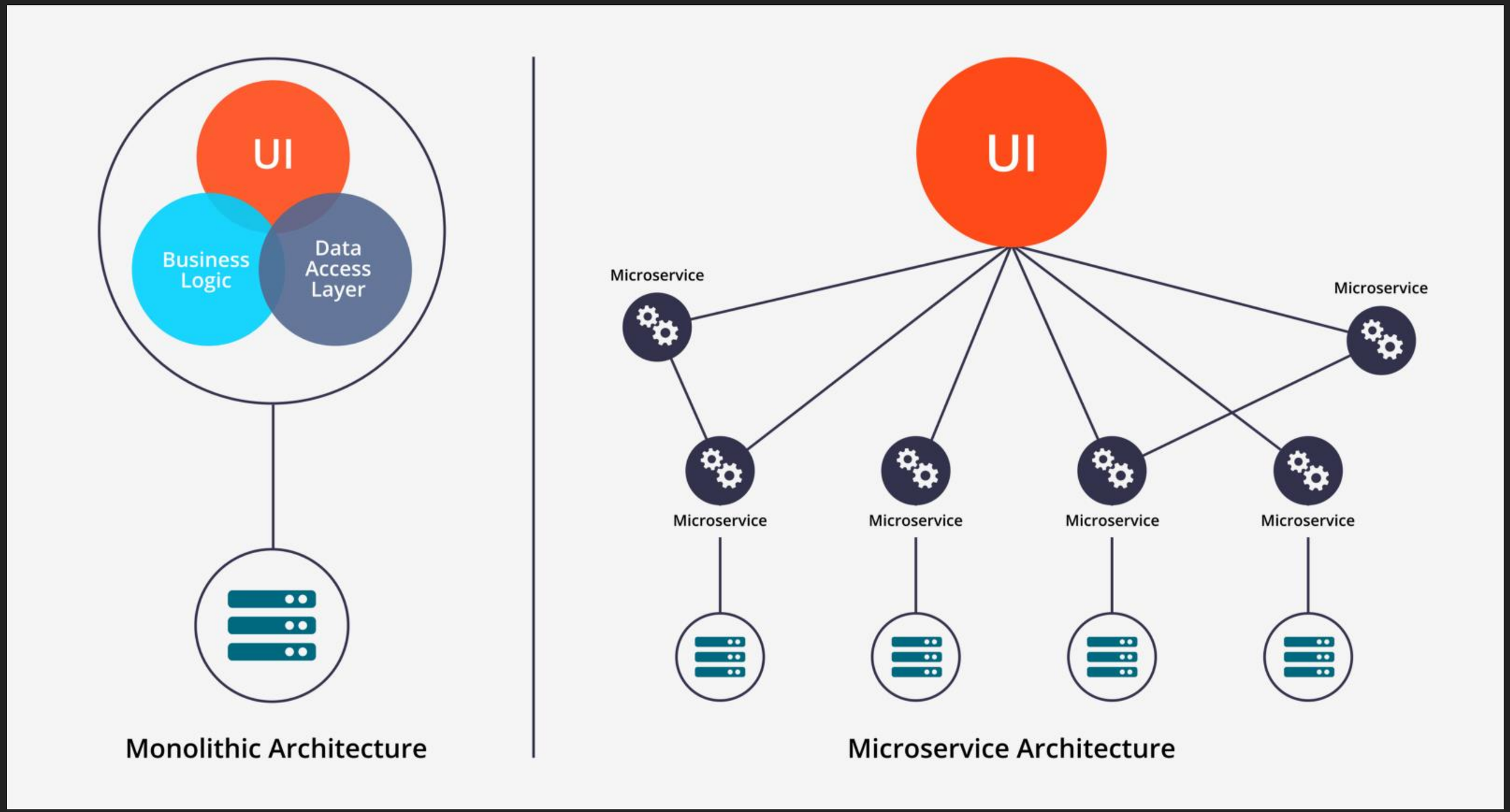
ETAI LEV RAN

ETAIL@IL.IBM.COM

INTRODUCTION

- whoami?
- Outline:
  - Background, service mesh
  - Istio: current and future
- Q&A during talk or find me afterwards

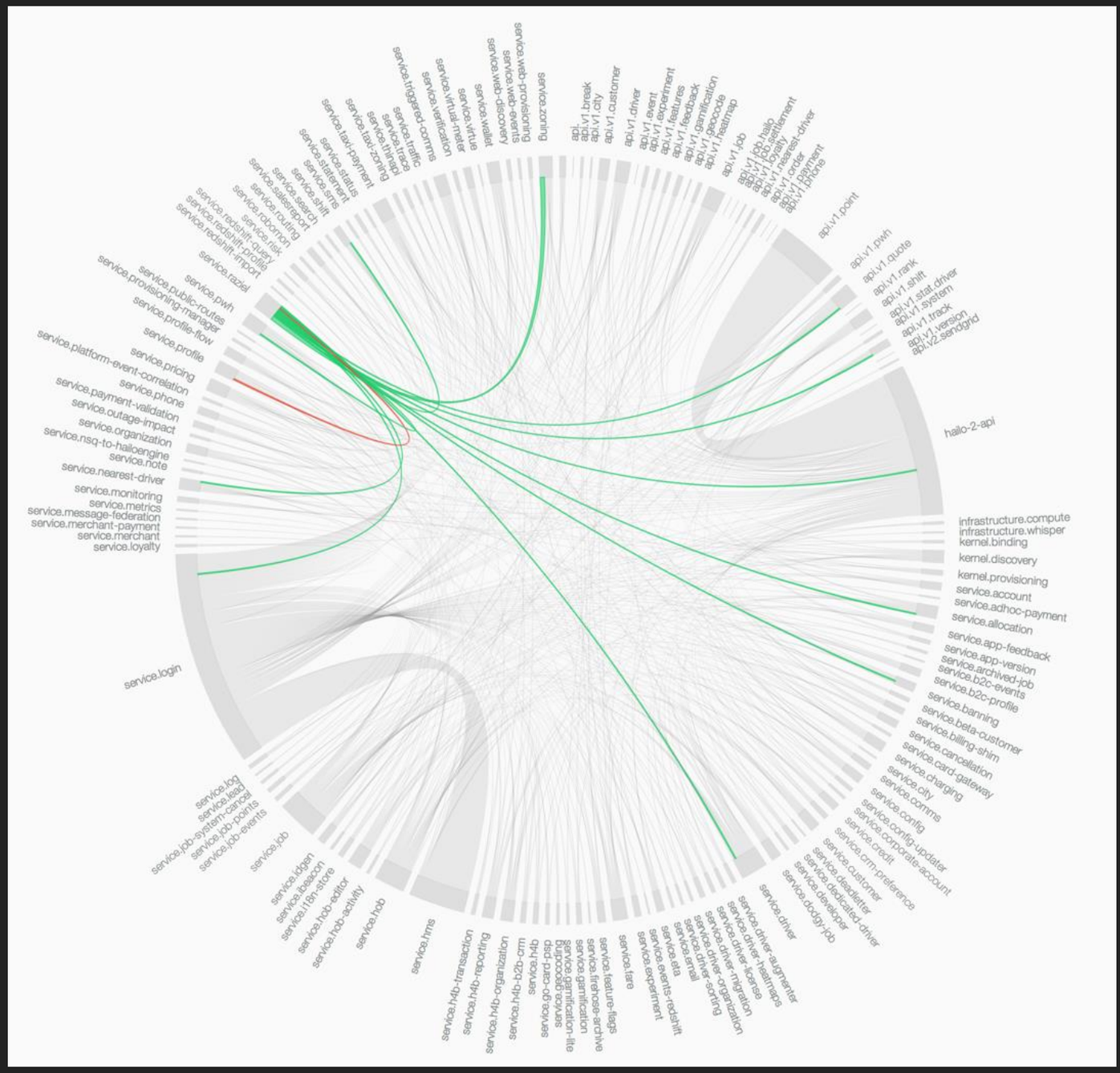
MICROSERVICE ARCHITECTURE - THEORY



MICROSERVICE ARCHITECTURE - PRACTICE

Improved delivery velocity in exchange for increased operational complexity

- Applications aren’t running in green-field environments
- Network layer is hard to manage
- Tooling is nascent



Hailo microservices

THINGS TO CONSIDER

- Security
- Canary deployments
- A/B testing
- Circuit breaking
- Rate limiting
- Fault injection
- Policy management
- Many more….

It’s doable, but…

It’s doable, but…

Requires a lot of coding

Service Mesh

A dedicated infrastructure layer to make service-to-service communication fast, safe and reliable

HISTORICAL CONTEXT

1st

- Library based mesh (~2010)
- Netflix Eureka, Ribbon, Hystrix

2nd

- Externalize load balancing (~2015)
- AirBnB Nerve and Syanpse

3rd

- Comprehensive, centralized control
- Istio

ISTIO

A service mesh designed to connect, manage and secure micro services

ISTIO

A service mesh designed to connect, manage and secure micro services

Open Source

ISTIO

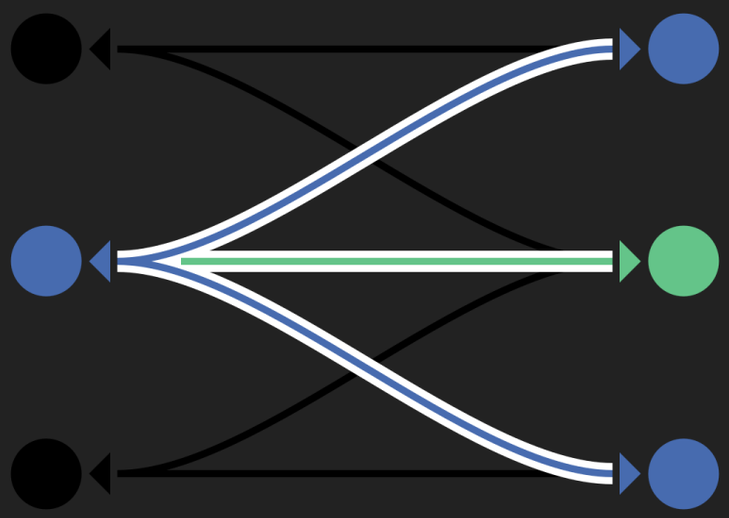
A service mesh designed to connect, manage and secure micro services

Open Source

Zero Code Changes *

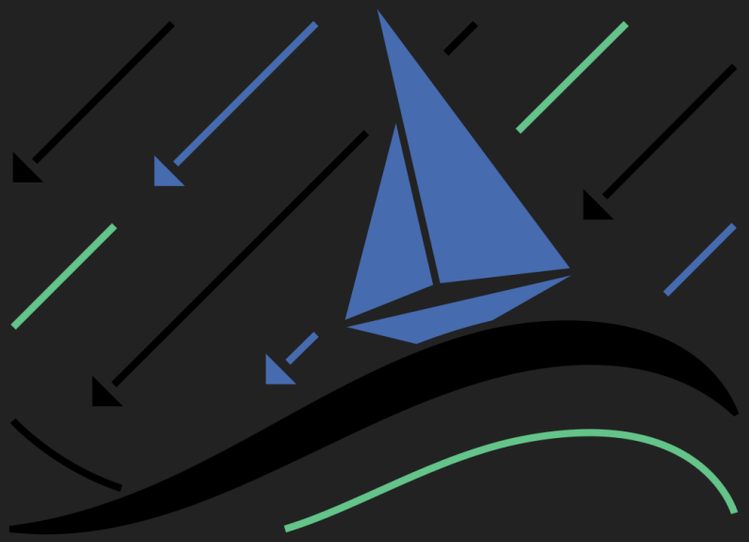
FEATURES

INTELLIGENT ROUTING AND LOAD BALANCING



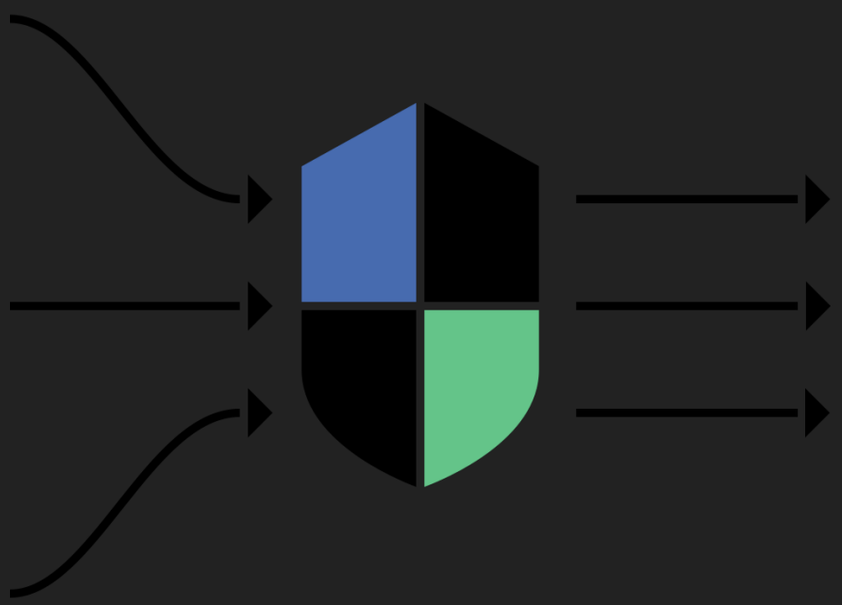
- Conduct traffic between services with dynamic route configuration
- A/B tests
- Canary releases
- Red/Black deployments

RESILIENCE ACROSS LANGUAGES AND PLATFORMS



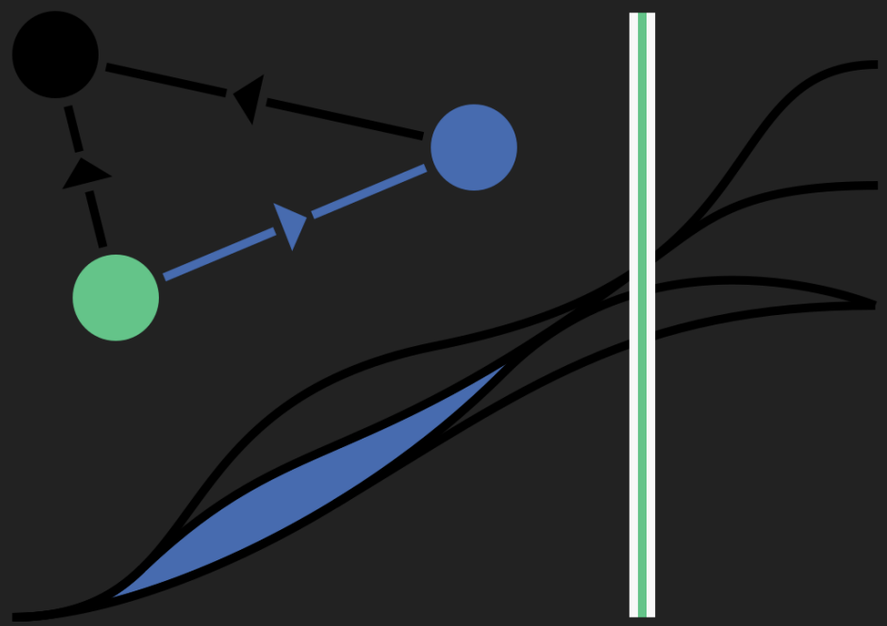
- Increase reliability by shielding applications from flaky networks and cascading failures in adverse conditions

FLEET-WIDE POLICY ENFORCEMENT



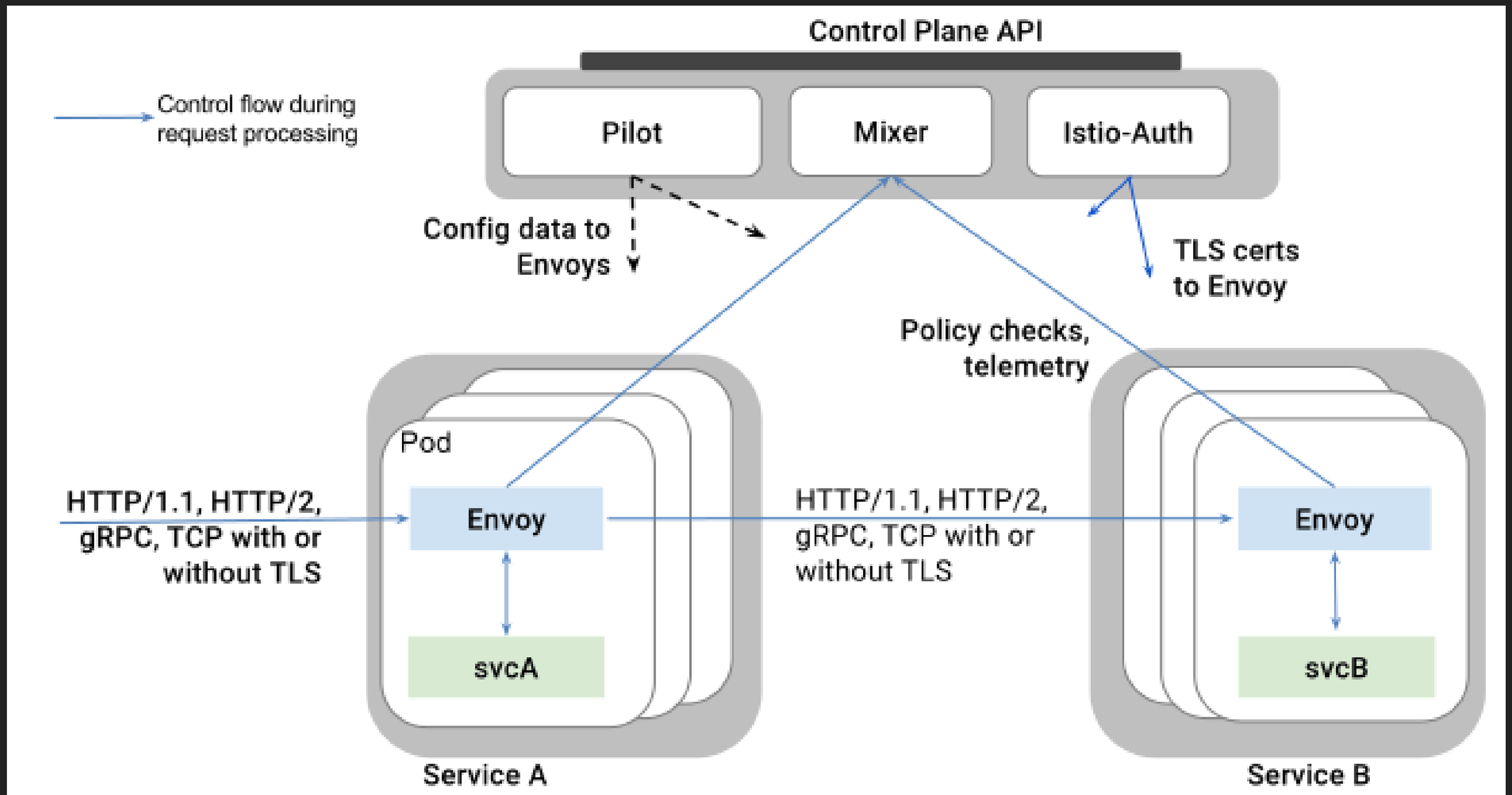
- Apply organizational policy to the interaction between services
- Ensure access policies are enforced
- Make sure resources are fairly distributed among consumers.

IN-DEPTH TELEMETRY AND REPORTING



- Understand the dependencies between services, the nature and flow of traffic between them, and quickly identify issues with distributed tracing.

ARCHITECTURE



Istio

Current status

- 8.4K+starts on Github
- Vibrant open source community
- Full house events at conferences
- Contribution from RedHat, VMware, Cisco, Tigera and many more
- Launched Istio 0.8

Customer feedback

- Telemetry out-of-the box saves a lot of time
- No code change is awesome
- Security as easy as it can get
- Traffic routing without involving SREs

Customer challenges

- It’s hard to understand, a lot of moving components
- Don’t clearly know how to get started
- We’ll wait until we move to microservices
- It’s not GA yet

ISTIO 1.0 PLAN

- Istio GA planned for July'18
  - Focus on productization (hardening, documentation, install and operations…)
- Follow k8s feature maturity model:
  - alpha => beta => stable
- Recommended adoption model:
  - Enable usage with limited scope (e.g., single service), expand to additional features and services when comfortable
  - Gradual adoption (Istio a-la-carte)

WHAT'S NEXT?

- Beyond 1.0, looking to make Istio ambient
  - Works well on single k8s cluster for services using REST/gRPC
- Multi-cluster
- Mesh expansion to VMs
- Multi-protocol
- Multiple programming models
- Just another part of the cloud native infrastructure

Roadmap (community)

- Cloud Foundry support
- Multi-cluster mesh
- Increment adoption use cases
- HA and multi-cluster for CA
- Event-based communication pattern
- Support toolkit

CONTACT ME

Etai Lev Ran: etail@il.ibm.com

USEFUL LINKS

- Web istio.io
- Twitter: @Istiomesh
- Traffic management using Istio: https://ibm.co/2F7xSnf
- Resiliency and fault-tolerance using Istio: https://bit.ly/2qStF2B
- Reliable application roll out and operations using Istio: https://bit.ly/2K9IRQX

THANK YOU!