

# Istio



Simplifying your microservices life

# Yours truly...



**Juan Pablo Genovese**  
(aka J.P.)

Cloud Architect / DevOps Practitioner

## A bit about me

- 22 years in the IT business in different industries.
  - From S/390 to Cloud environments.
  - From COBOL to microservices.
  - Loves working on Cloud Native & Distributed architectures, enabling companies in the Cloud World and teaching..
- 
- Absolutely passionate about teaching and dancing Tango
  - Plays guitar and sings.
  - Have two dogs (a Giant schnauzer and a Boxer)
  - Loves meeting people and talking about pretty much everything

## Keep in touch!



[www.jpgenovese.com](http://www.jpgenovese.com)

# The African Lion Philosophy for Solving Complex Problems

How do you eat an elephant?

One bite at a time.



# Microservices are great!

- Independent development
- Independent deployment
- Independent scalability
- Reusability

## ... but microservices are also hard.

- Increased complexity in development cycle
- Increased complexity in deployment cycle
- Have to work with distributed transactions (oh... crap!)
- Increased technical complexity
- Experts needed!
- Boundaries madness
- Versioning is HARD
- Communication hardships are often minimized

# The Artisan Way

- Find the service. Where's the address?
- Make sure it is alive.
- Make sure it is free to be used.
- Figure out if you have permissions to use it.
- Identify myself to the service. Who am I.
- Actually use that wretched service!
- Report the usage stats.
- Did it fail? Damn it. Do this all over again!



# Istio

To the rescue!

# A bit of background

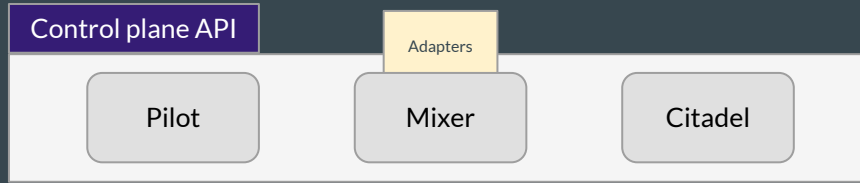
- A service mesh for your microservices
- Backed by Google, IBM and Lyft
- Currently in version 0.8
- In heavy development now, most features are still alpha, but functional
- It needs a container orchestrator to be deployed. Currently doable in K8s
- Integrates with Consul & Eureka
- Soon support for Cloud Foundry and Apache Mesos



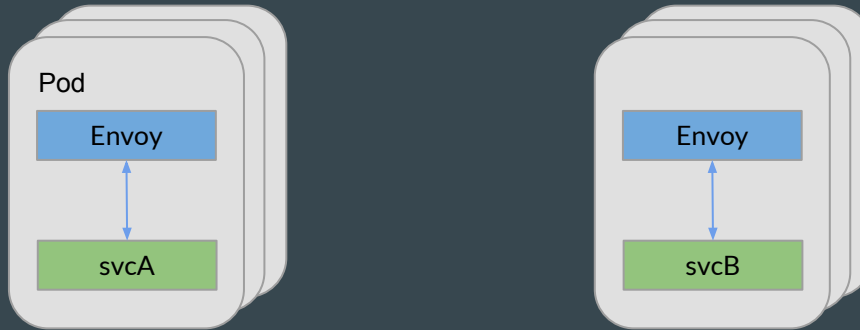
# What problems Istio solves?

- Traffic Management
  - Load balancing, routing, service discovery, failure recovery.
- Service Identity and Security
  - Service authentication and authorization and encryption.
- Policy Enforcement
  - Resources distributed fairly across services. Centralized and pluggable configuration without modifying the application code.
- Telemetry
  - Have easy access to understand the dependency between services, the traffic flow and identification of issues.

# The architecture



Control Plane



Data Plane

# Envoy



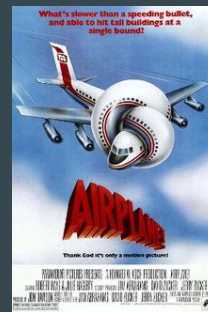
Kinner C-7 Envoy

- Dynamic service discovery
- Load balancing
- TLS termination
- HTTP/2 & gRPC proxying
- Circuit breakers
- Health checks
- Staged rollouts with %-based traffic split
- Fault injection
- Rich metrics

# Pilot



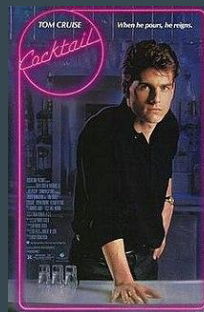
- Service discovery for the Envoy sidecars
- Traffic management capabilities for intelligent routing (e.g., A/B tests, canary deployments, etc.),
- Resiliency (timeouts, retries, circuit breakers, etc.).



# Mixer



- Enforces access control and usage policies across the service mesh
- Collects telemetry data from the Envoy proxy and other services.
- Use adapters to connect to many metrics, authorization, quota and logging backends.



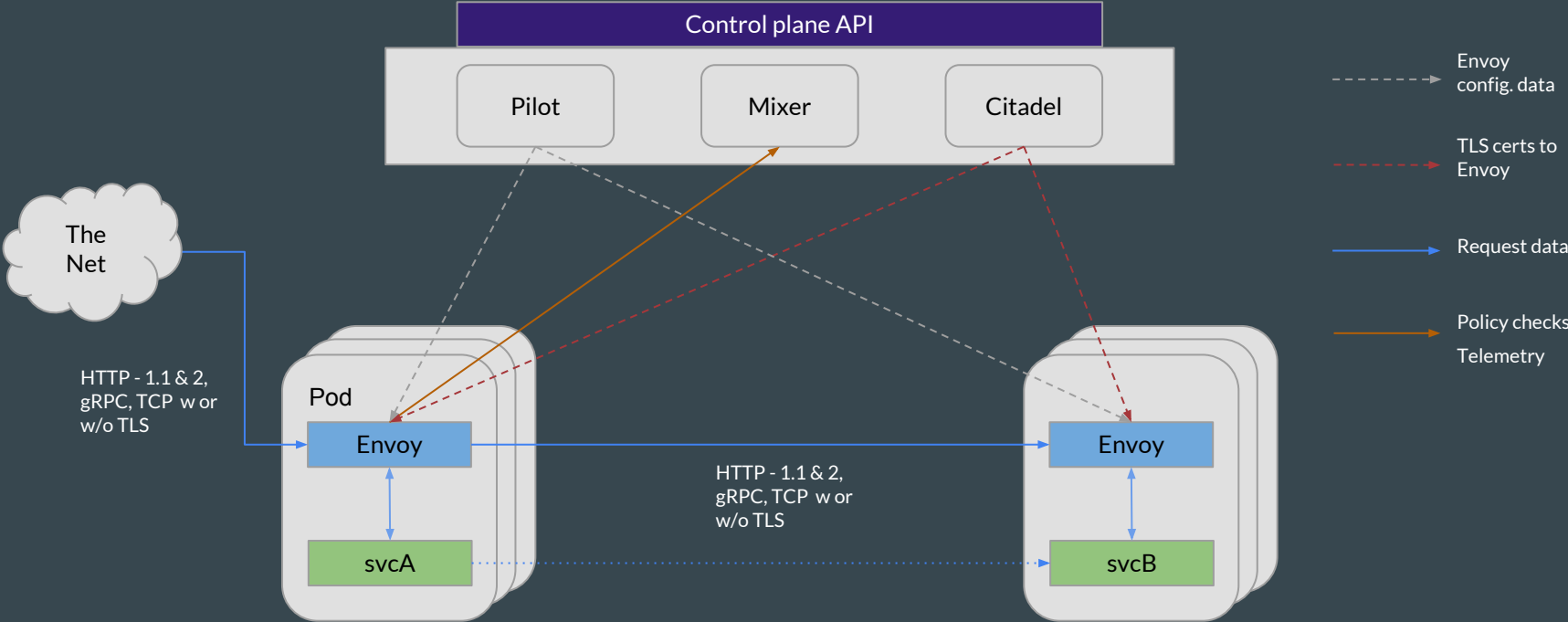
# Citadel

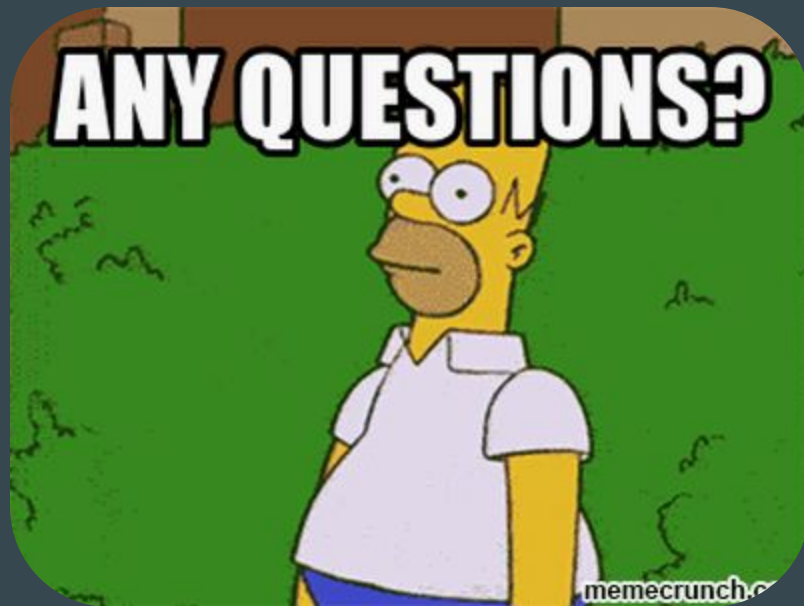


- Provides strong service-to-service and end-user authentication
- Built-in identity and credential management.
- Provides operators the ability to enforce policy based on service identity rather than network controls.



# How does Istio work?







# Thank you!!

