



# ISTIO

Here's another fine  
mesh you've got me  
into!

Karthik Prabhakar



kubernetes

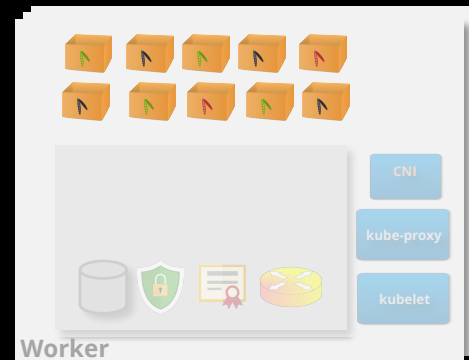
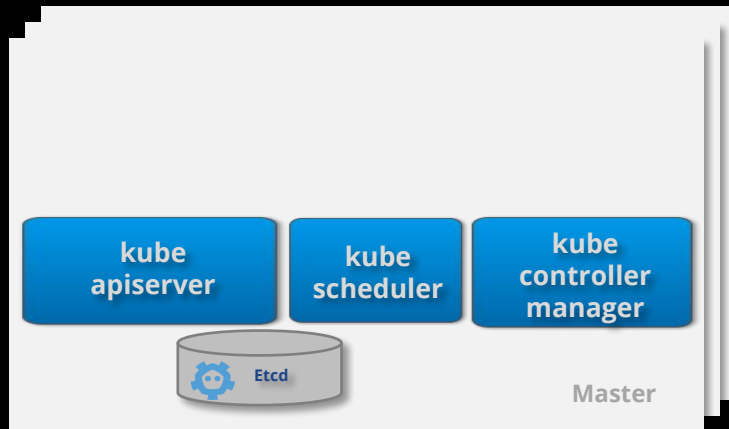
#SFK8s



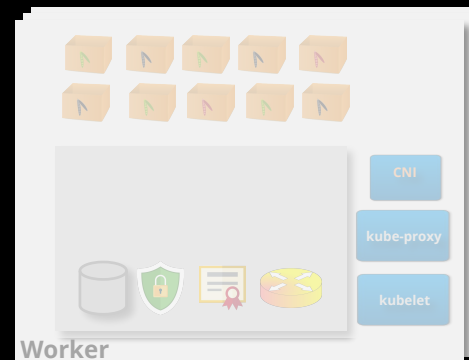
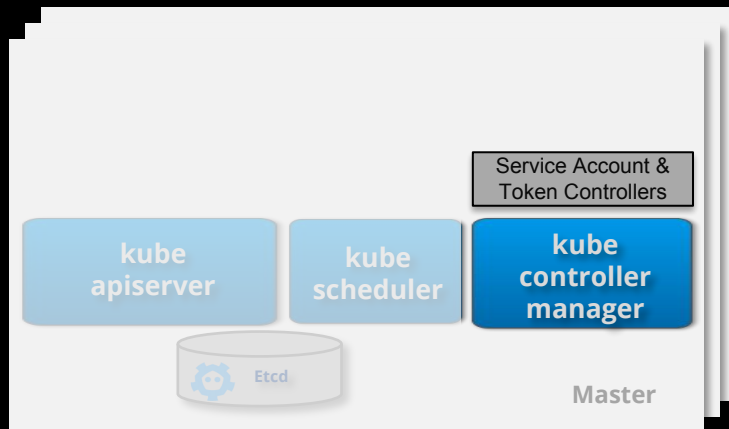
# Assumed: Kubernetes Connectivity Concepts



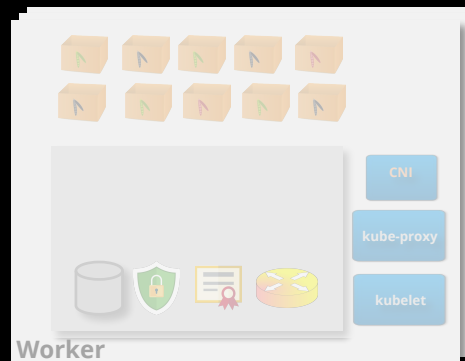
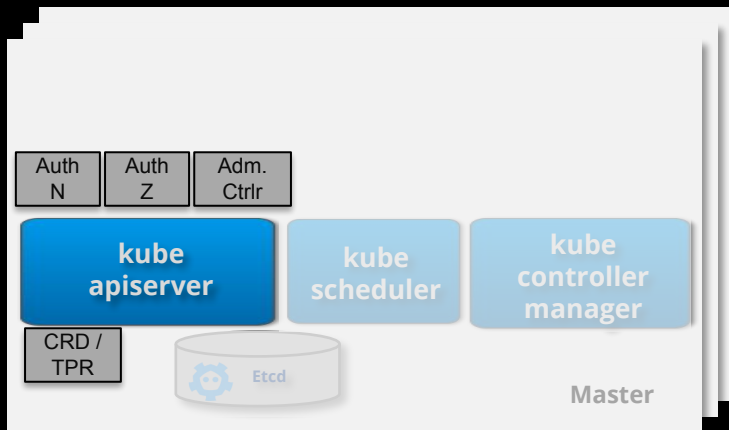
# Assumed: Other Important Kubernetes Concepts



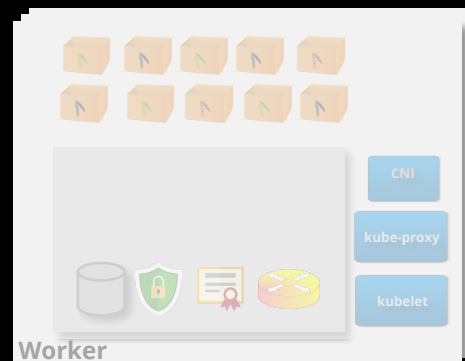
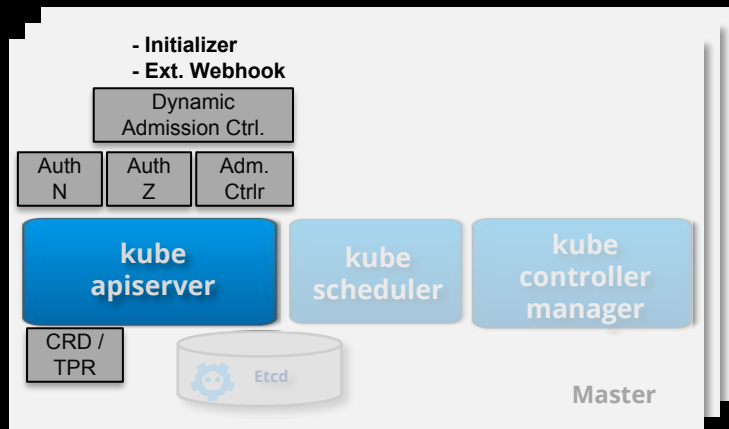
# Other Important Kubernetes Concepts



# Other Important Kubernetes Concepts

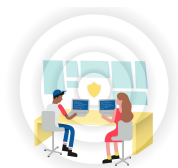


# Other Important Kubernetes Concepts

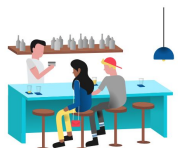




Traffic Management, Resilience and Load Balancing



Security, Identity, Identity-linked encryption



Fleet-wide Policy enforcement



Telemetry and Reporting



# Benefits to Microservices Deployments

## Traffic Management

Circuit Breakers

Retries

Timeouts

Rate Limits

Outlier Detection

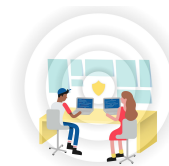


## Observability

Stats

Tracing

Logging



## Security, Policy

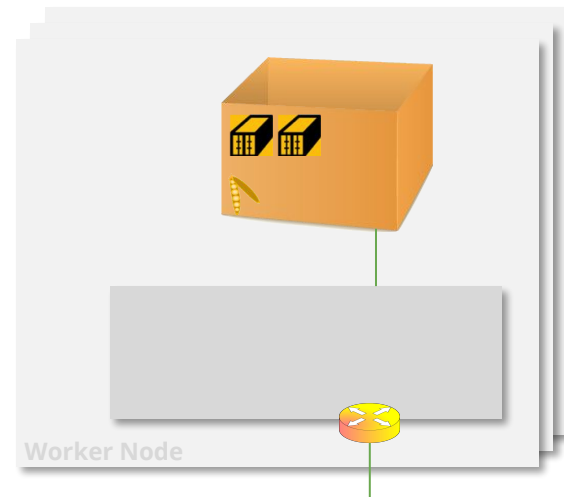
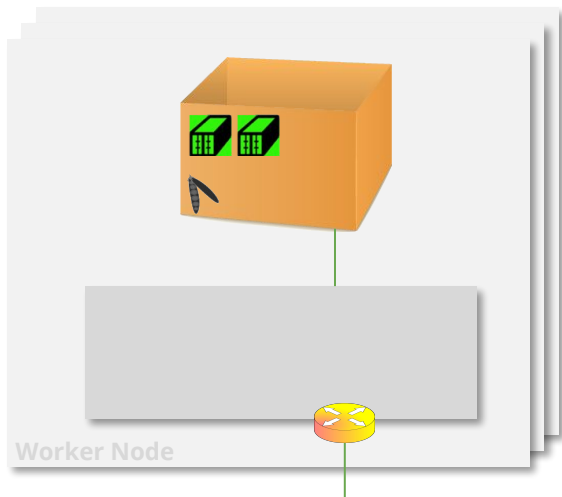
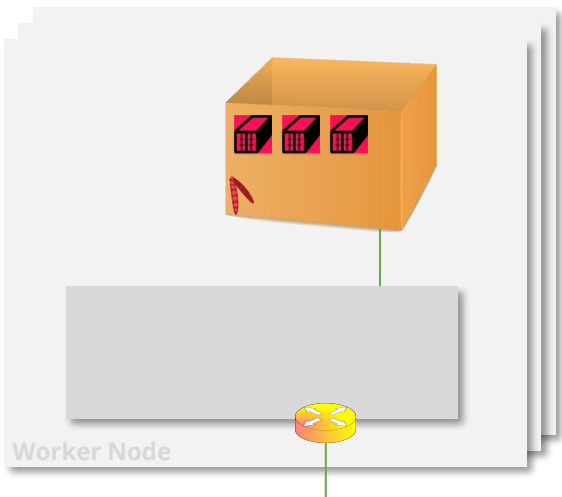
TLS

Routing

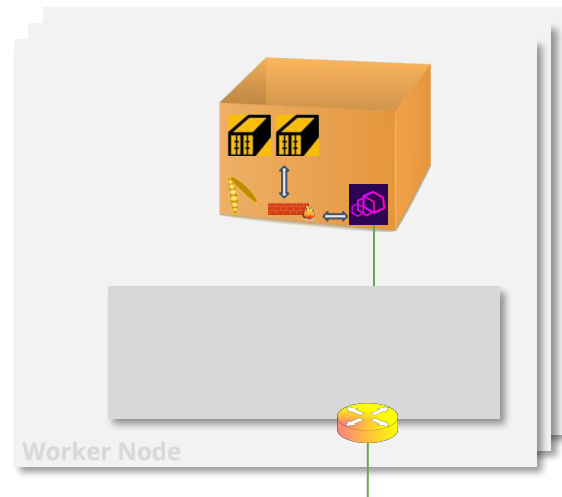
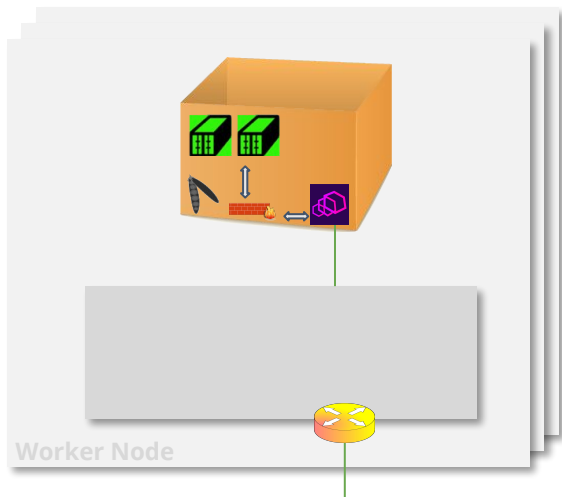
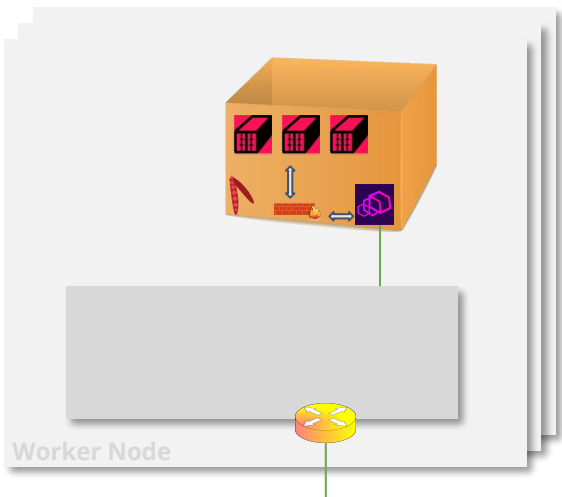




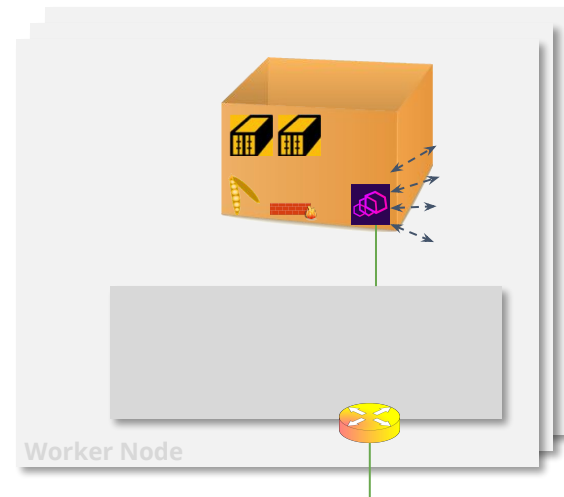
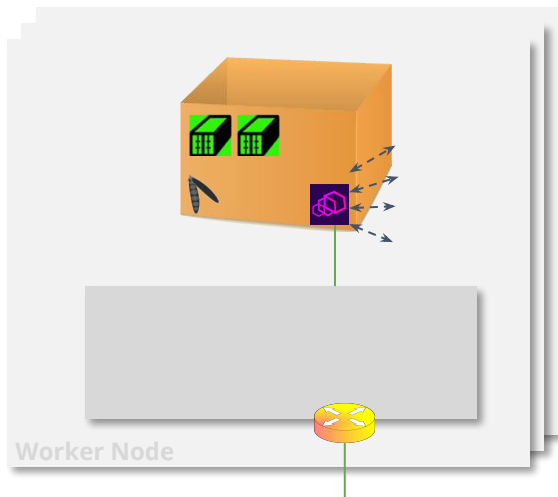
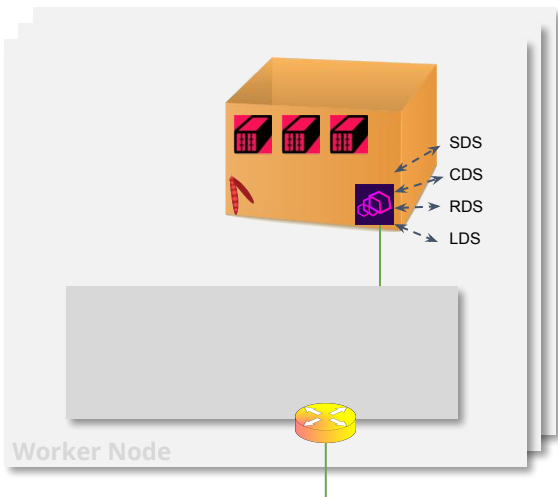
## Data Plane: Istio-Proxy (Envoy)

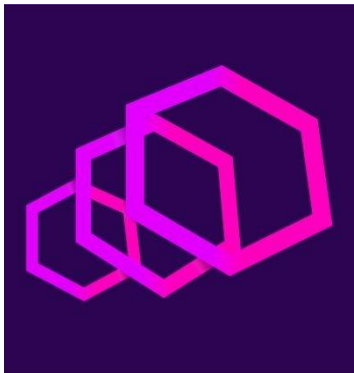


## Data Plane: Istio-Proxy (Envoy)



# Data Plane: Istio-Proxy (Envoy)



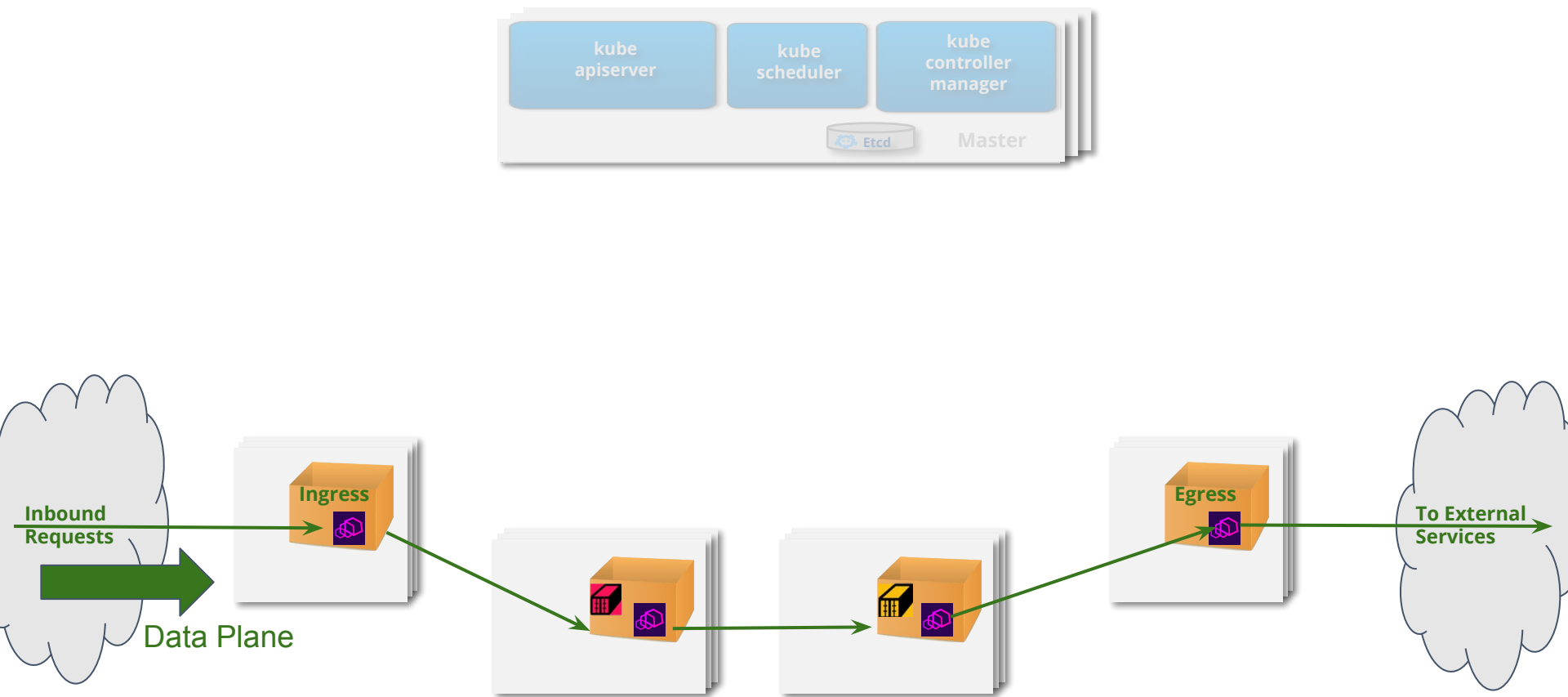


- Lightweight
- Fast
- Modular & Extensible
- Control Plane integration via JSON/REST API's
- Service Discovery + **Active**/Passive Health Checks
- HTTP/2 (+ gRPC), HTTP/1.1
- TCP, MongoDB, etc.
- Hot Restarts and/or Rolling Updates
- Hardened from deployments at scale
- CNCF (?)
- ... and fits Kubernetes architecture nicely!



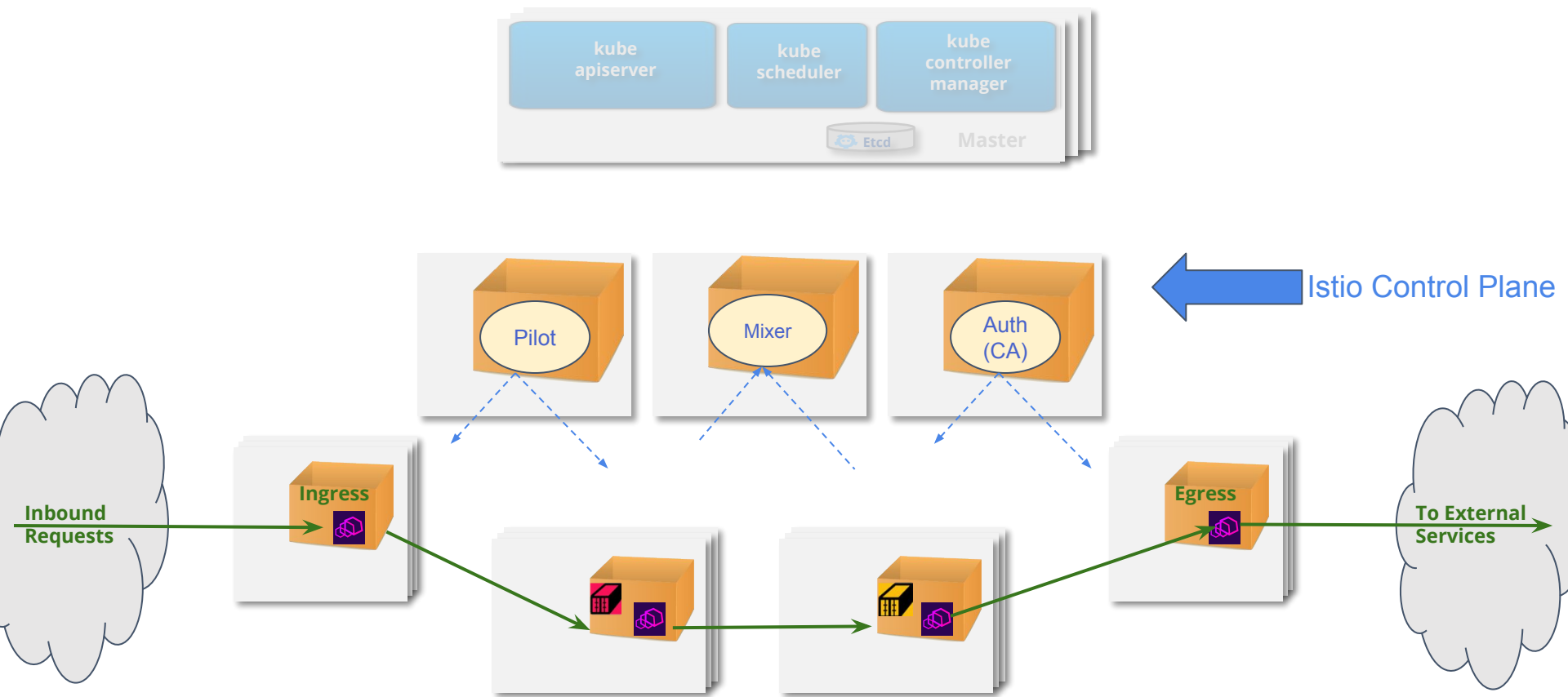


# Istio Control Plane



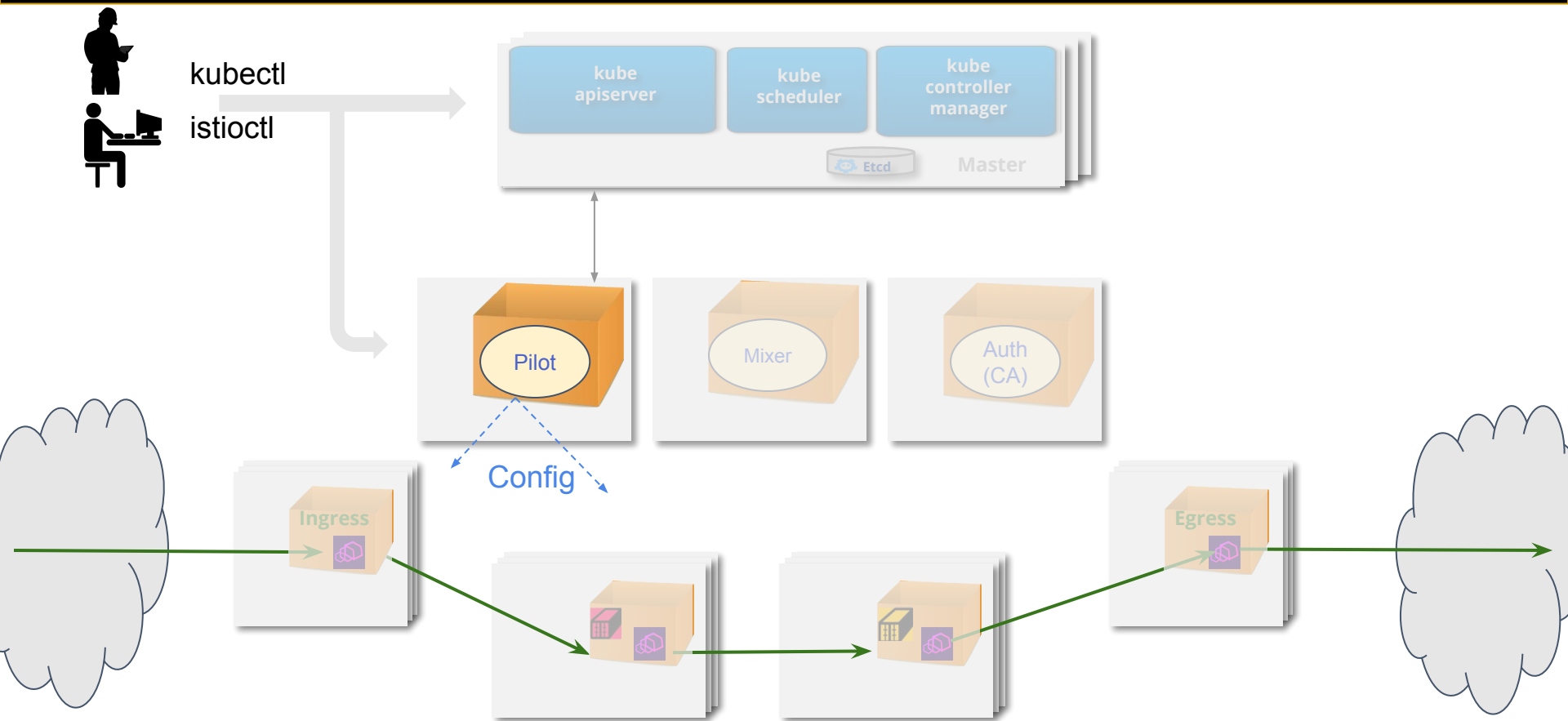


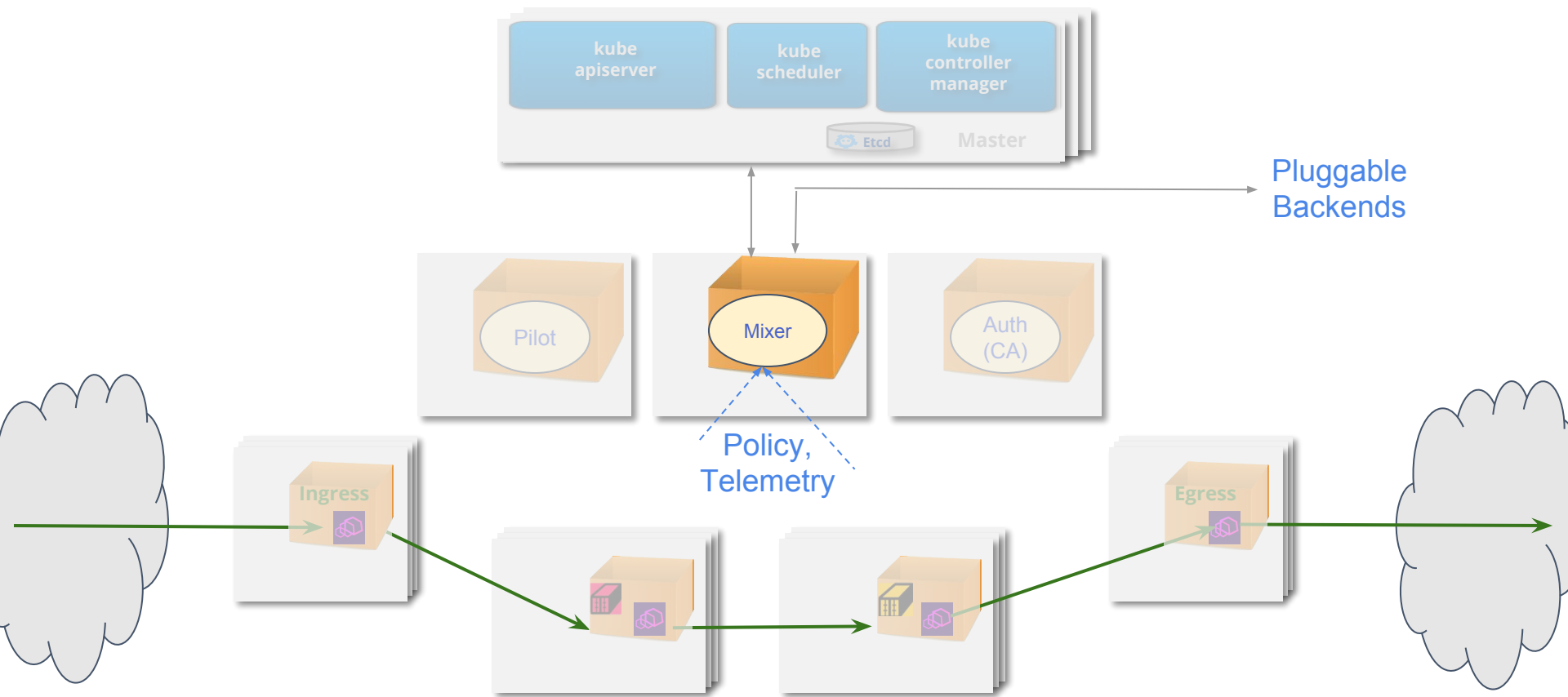
# Istio Control Plane



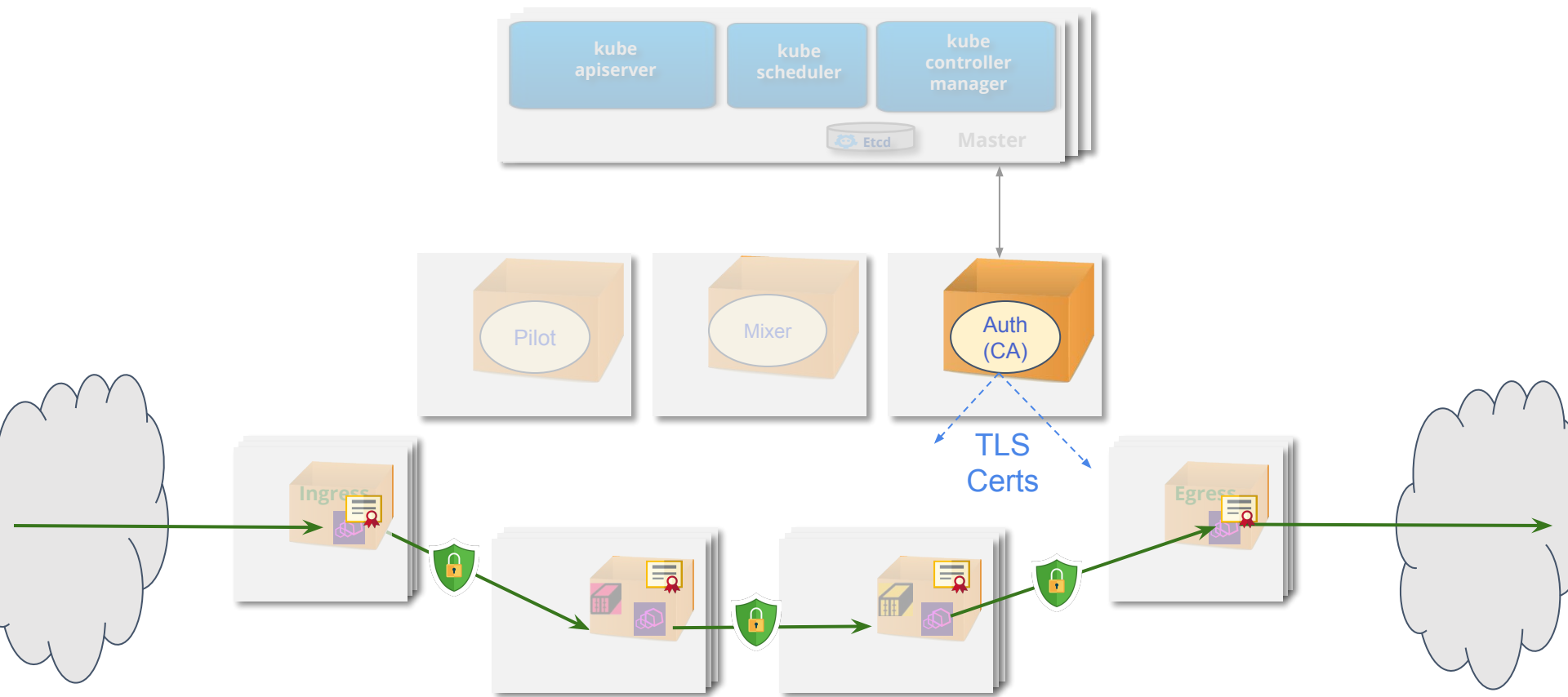


# Istio Pilot









## Current Istio Development Themes



- Hybrid Istio (Multiple/Heterogenous Container|VM clusters)
- AuthN / AuthZ / Policy
- Performance, Scale
- Multi-Namespaces
- More Integrations
- Consistent Configuration
- Session Persistence, Caching
- Operational Patterns (Sharding, Anomaly Detection, etc.)
- Better Edge Integration
- Overall system robustness



Demo Time:

- 1. Deploy Kubernetes with Calico**
- 2. Deploy Istio**
- 3. Deploy microservice application**
- 4. Operate service mesh**
  - a. Observability Demo**
  - b. Traffic Management Demo**
  - c. Security (mTLS, AuthZ) Demo**



**NEW!** *(Announcing Next Week)*

Hands-on Class (1-day):

# Intro to Service Mesh with Istio

<https://tigera.io/contact>





[slack.projectcalico.org](https://slack.projectcalico.org)



[istio.io](https://istio.io)