KubeCon CloudNativeCon
North America 2019

From Brownfield to Greenfield: Freddie Mac's Service Mesh Journey

Lixun Qi
@lixunqi
Sr Tech Lead

Shriram Rajagopalan
@rshriram
Unprincipled Engineer

About Freddie Mac

- Chartered by Congress in 1970 to provide liquidity, stability and affordability to the U.S. housing finance market
- One of the two government sponsored enterprises participating in the secondary mortgage market
- $2 Trillion in assets

What this talk is NOT about ☺

- Mortgage rates
- Subprime mortgage crisis

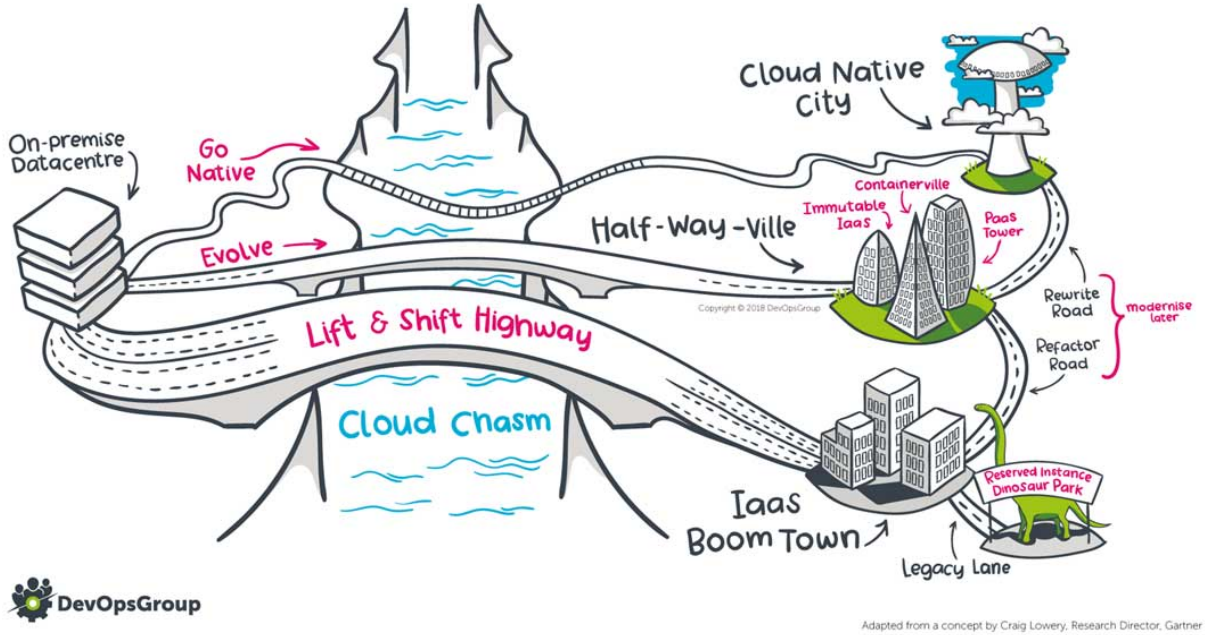
Major IT Initiatives

Application Modernization

- Monolith to Microservice
- DevOps and CI/CD for cloud native applications

Cloud Transformation

- Lift and Shift to VMware cloud on AWS
- Cloud Native Infrastructure on Managed Kubernetes

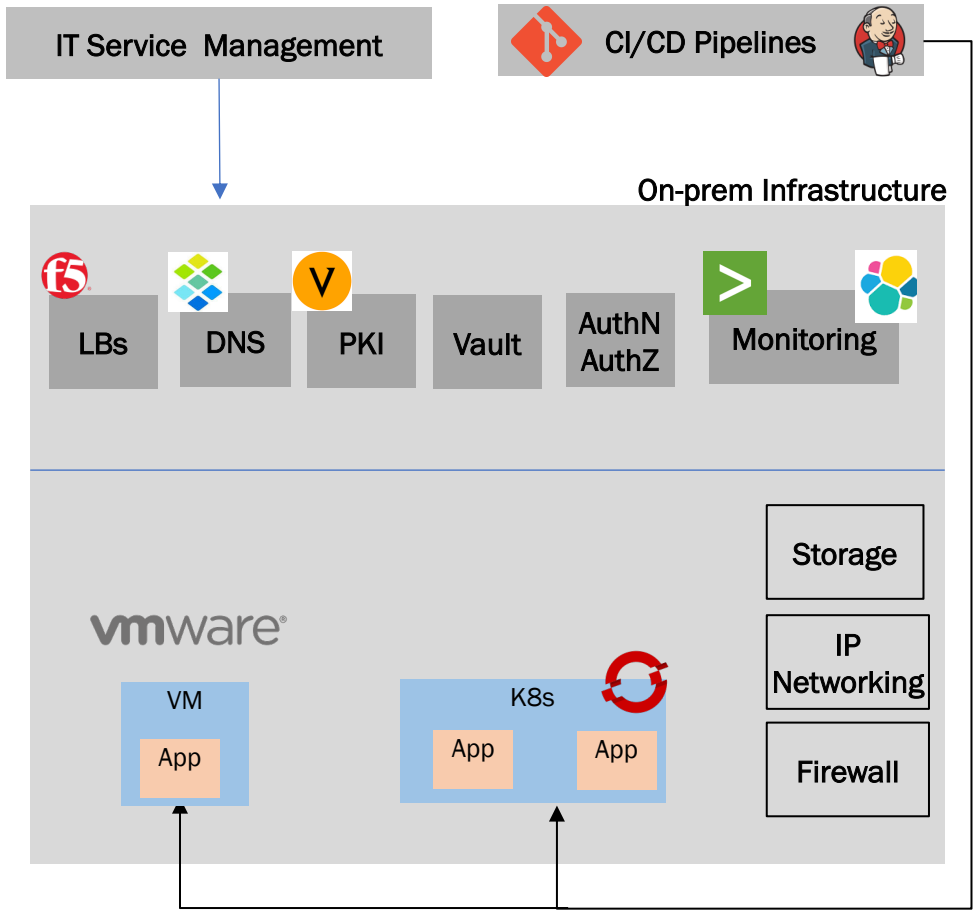


Infrastructure Footprint

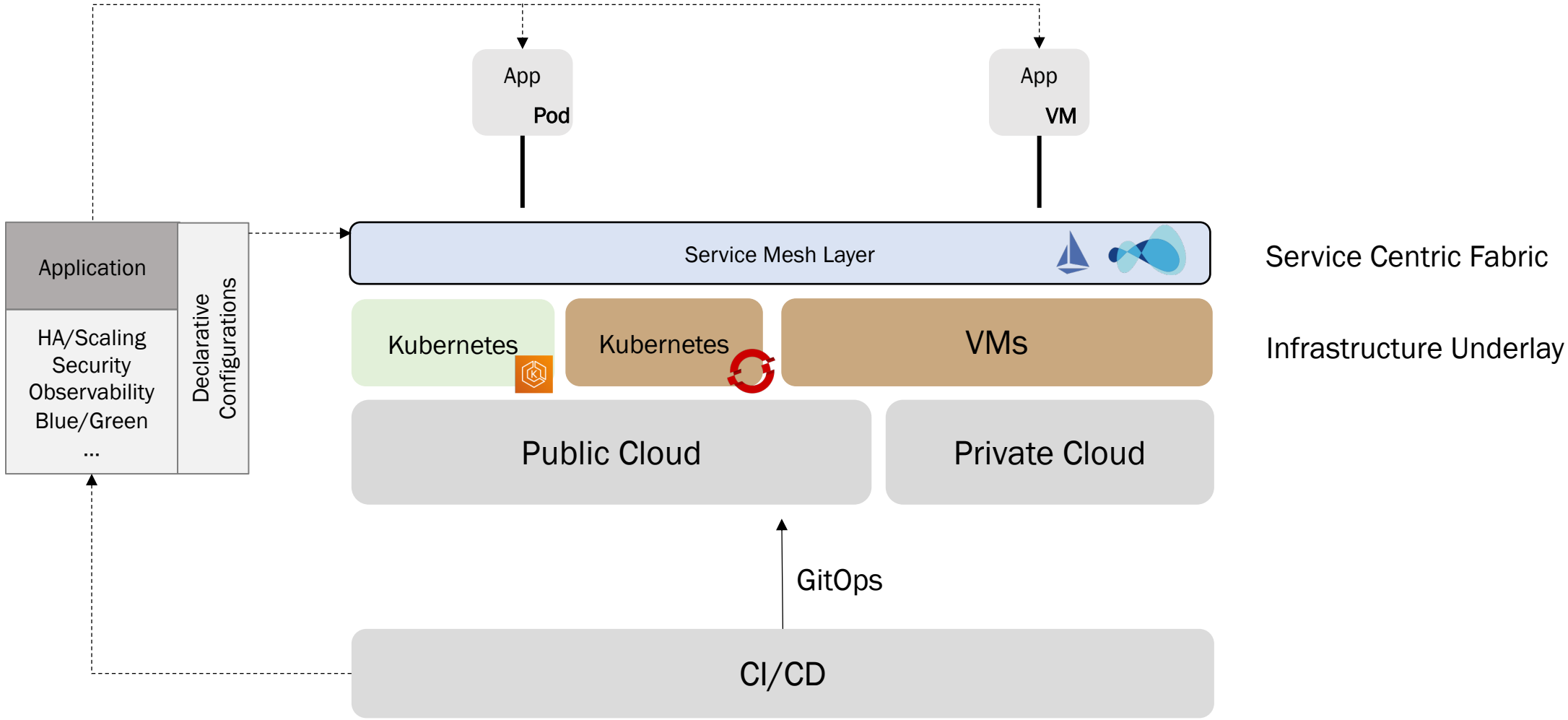
- *600+ Applications*
- VMware VMs
- Kubernetes
  - Multiple vendors
- Hardware load balancers
- API Management platforms
- Large SQL/NoSQL databases
- ESBs
- Java EE App Servers
- Cloud Services

From Legacy Infrastructure

- Application teams have many touch points for infrastructure and platforms
- Consumption of infrastructure and platform services requires manual ticket requests, posing challenges to CI/CD pipelines
- Tight coupling between Development and Operation
- Hybrid cloud challenges



To Software Defined Infrastructure



To Software Defined Infrastructure…

- The mesh abstracts common application support features into the service platform for both brownfield and greenfield applications
  - PKI management
  - Observability
  - Security in transit
  - HA/Autoscaling
  - Canary deployment
  - Chaos engineering
  - Corse grained authentication & authorization

CI/CD and GitOps

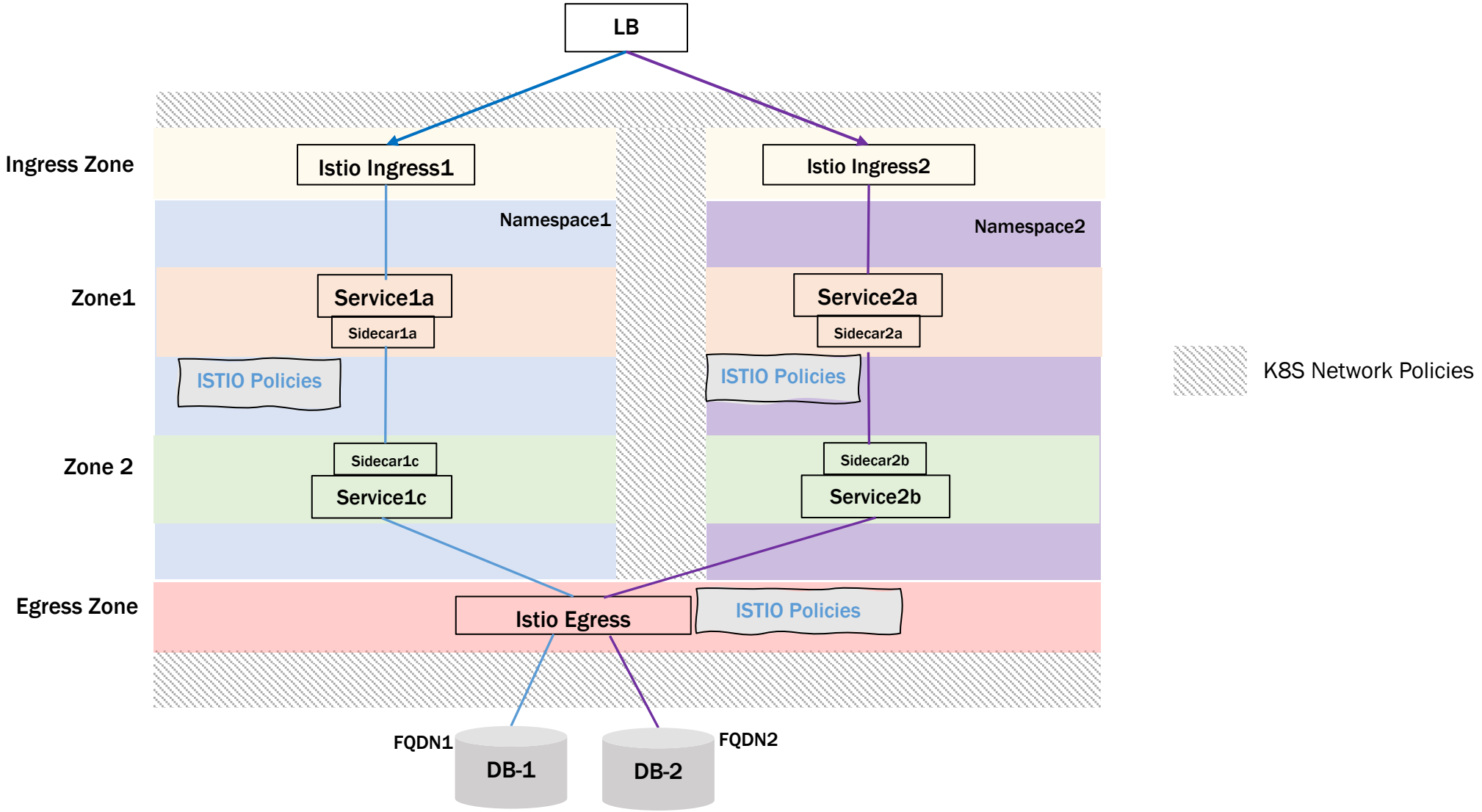
- Configurations as code
- Central, declarative policies for better compliance audit and governance
- Decouple Development from Operations
- Faster release cycles

Security Use Cases

- Challenges
  - Modern applications are dynamic and distributed
  - Insider threats
  - Traditional L3/L4 and perimeter FW models not in microservice data path

- Istio is critical component of our next gen FW design
  - Zero trust and micro-segmentation
  - Distributed and closer to applications
  - Identity based
  - DNS-aware policy
  - Mutual TLS
  - Cert/Key rotation
  - Security as Code
  - Centralized policy management and compliance audit

Micro Segmentation w/ Kubernetes & Istio

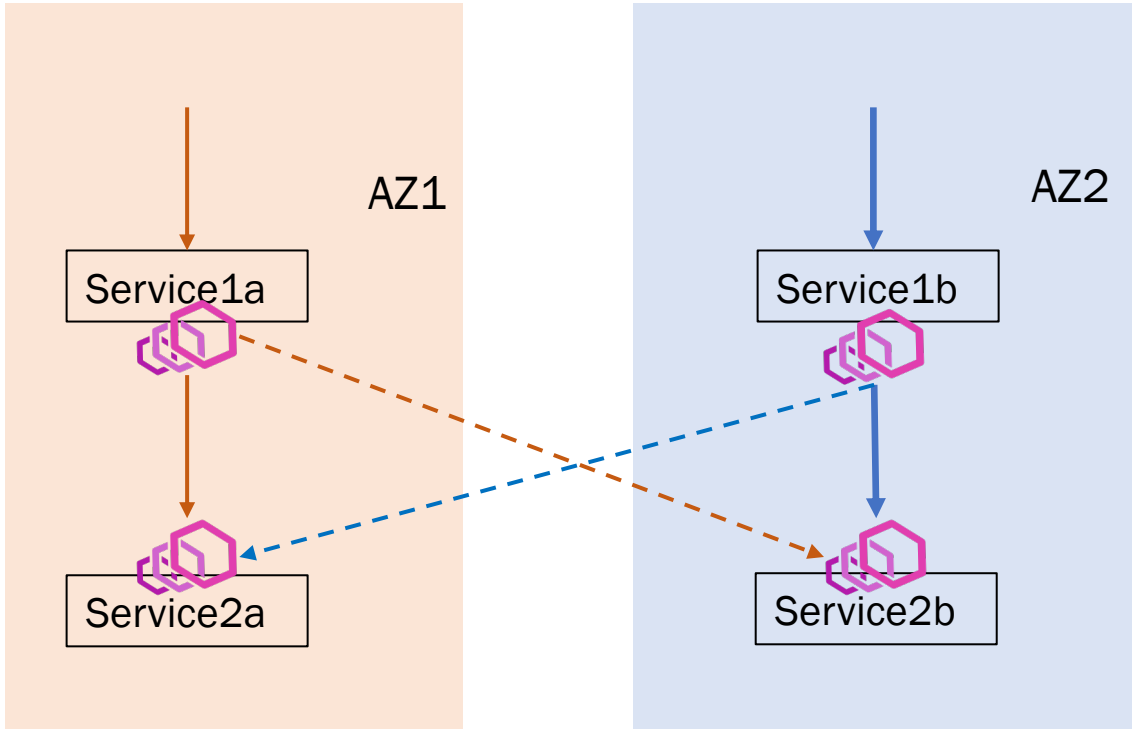


Availability Use Cases

- Handling failures
  - Locality-aware load balancing in a multi AZ Kubernetes cluster
- Distributed Load Balancer
  - Migrate from central hardware load balancer to multiple Cloud LBs
  - Move central traffic/route rules to Istio Ingress GW/Sidecar
- Handling load
  - Autoscaling based on app metrics
- Zero-downtime updates
  - App updates using version-aware routing
  - Seamless certificate refresh at ingress gateway and sidecars

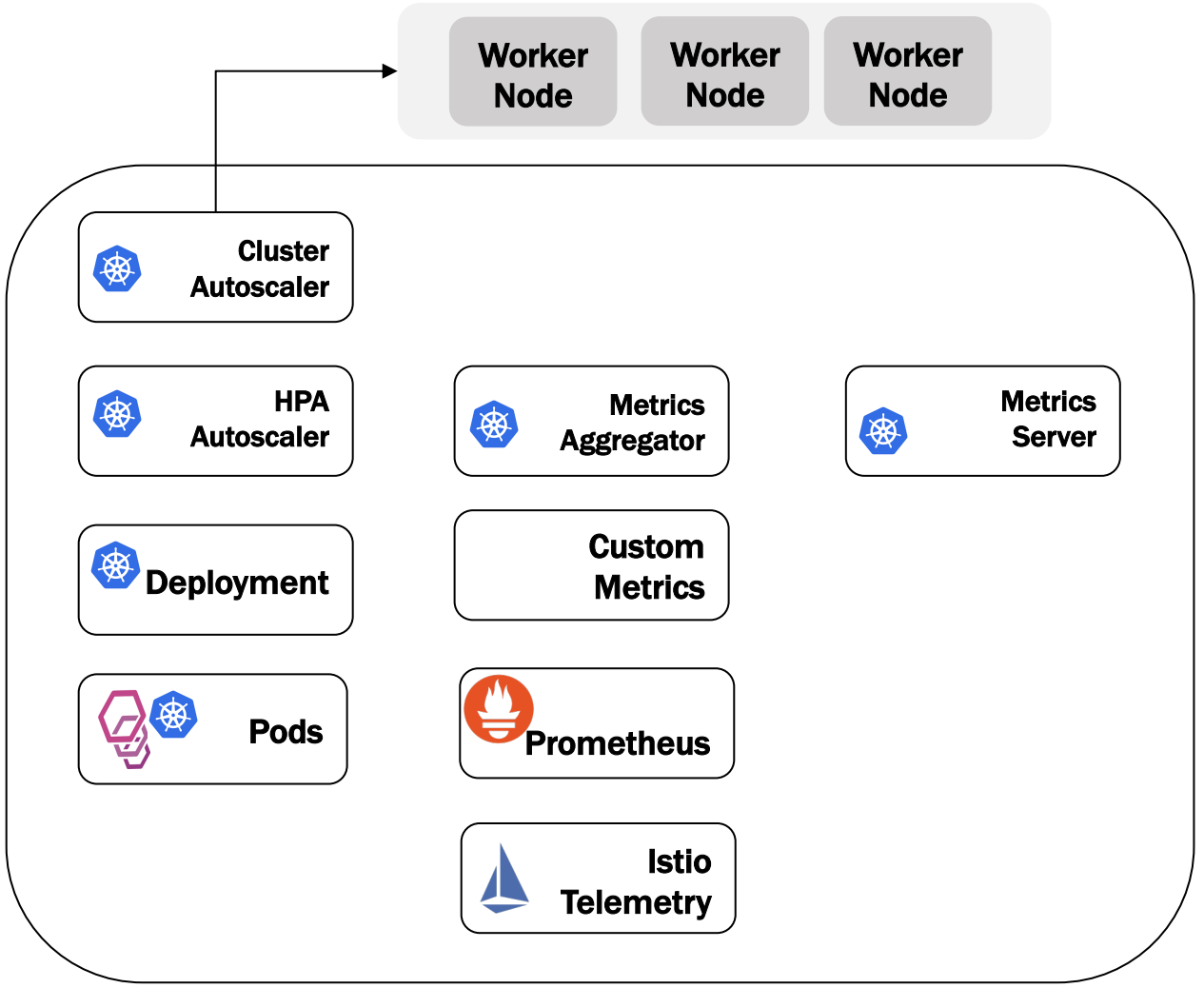
Availability: Locality-Aware Load Balancing

- Locality load balancing for performance and cost optimization
- Kubernetes doesn't provide locality-based load balancing
- Istio's locality-prioritized and fail-over LB feature offer additional HA design options



Availability: Autoscaling

- Scaling on CPU and memory doesn't have direct relationship to SLA
- Istio offers additional metrics to auto scale, i.e., request volume, response time and status codes



Brownfield Integration - Goals

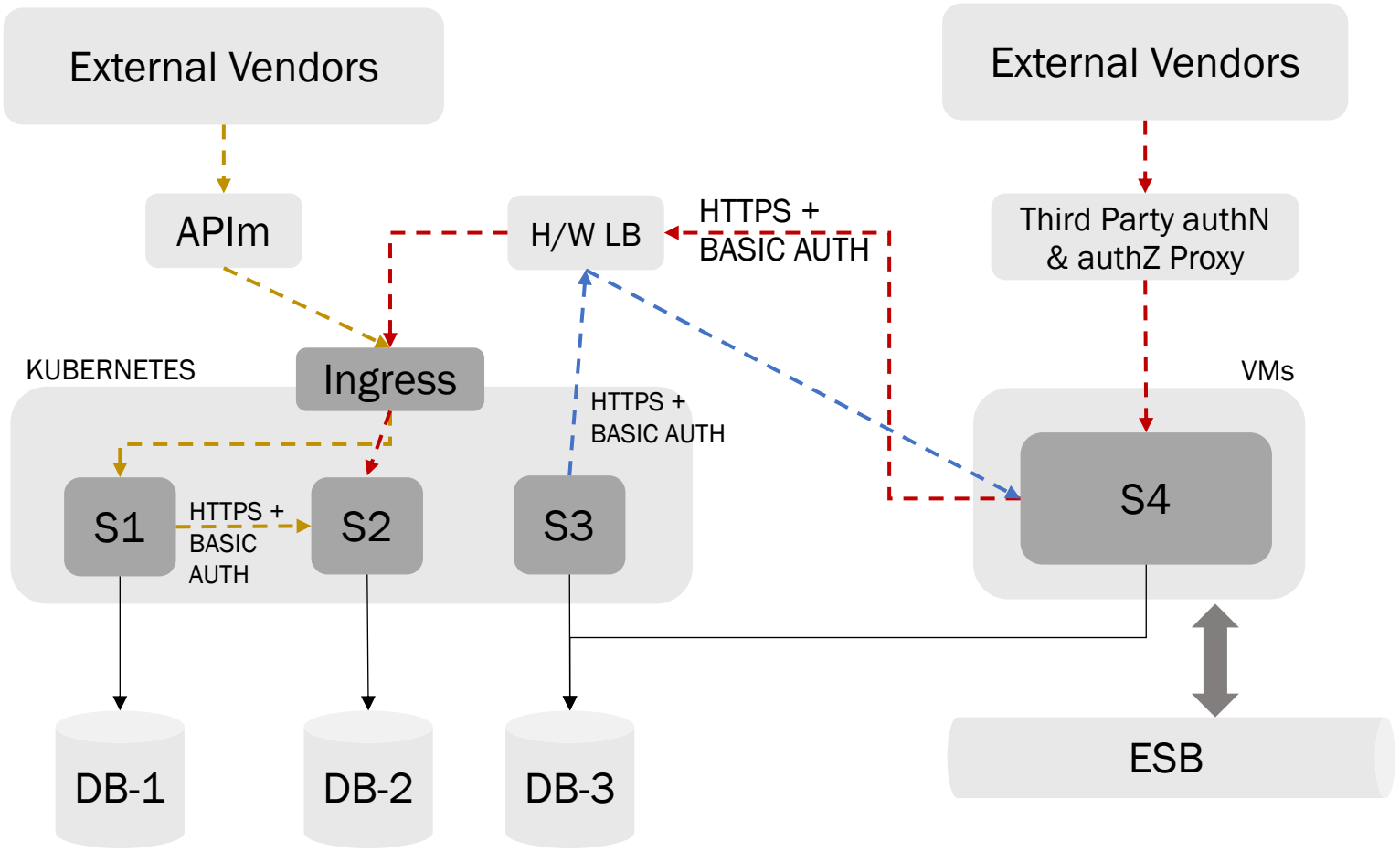
- Mesh first, containerize next!

- Traffic flows:
  - VM → Kubernetes Ingress
  - Kubernetes pods → VM
  - LB → VM or Kubernetes service
  - VM → VM
- Uniform security
  - Mutual TLS, AuthN & AuthZ policies applied uniformly
- Uniform telemetry

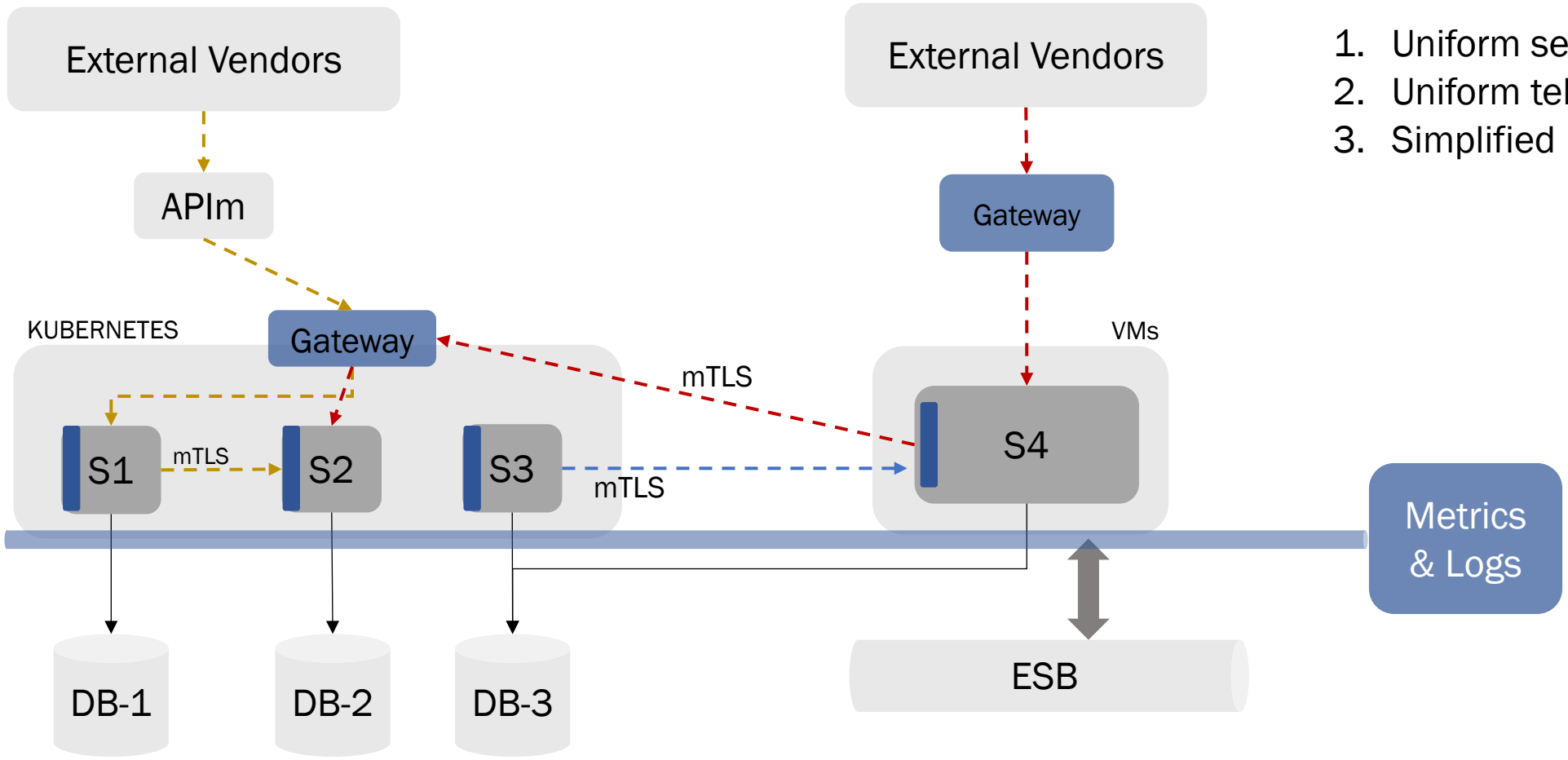
Brownfield Integration - Challenges

- Traffic flow:
  - Asymmetric network connectivity between VMs and Kubernetes
- Security
  - Allow for migration from app-centric certs to workload-centric certs
  - Allow for mesh and non-mesh traffic to VMs
  - Allow for integration with in-house CA / Trust Store
- Traffic interception
  - IP tables not feasible due to Java middleware issues
- Service & endpoint registration
  - Scattered across h/w load balancers and VMware management services

Example Use Case

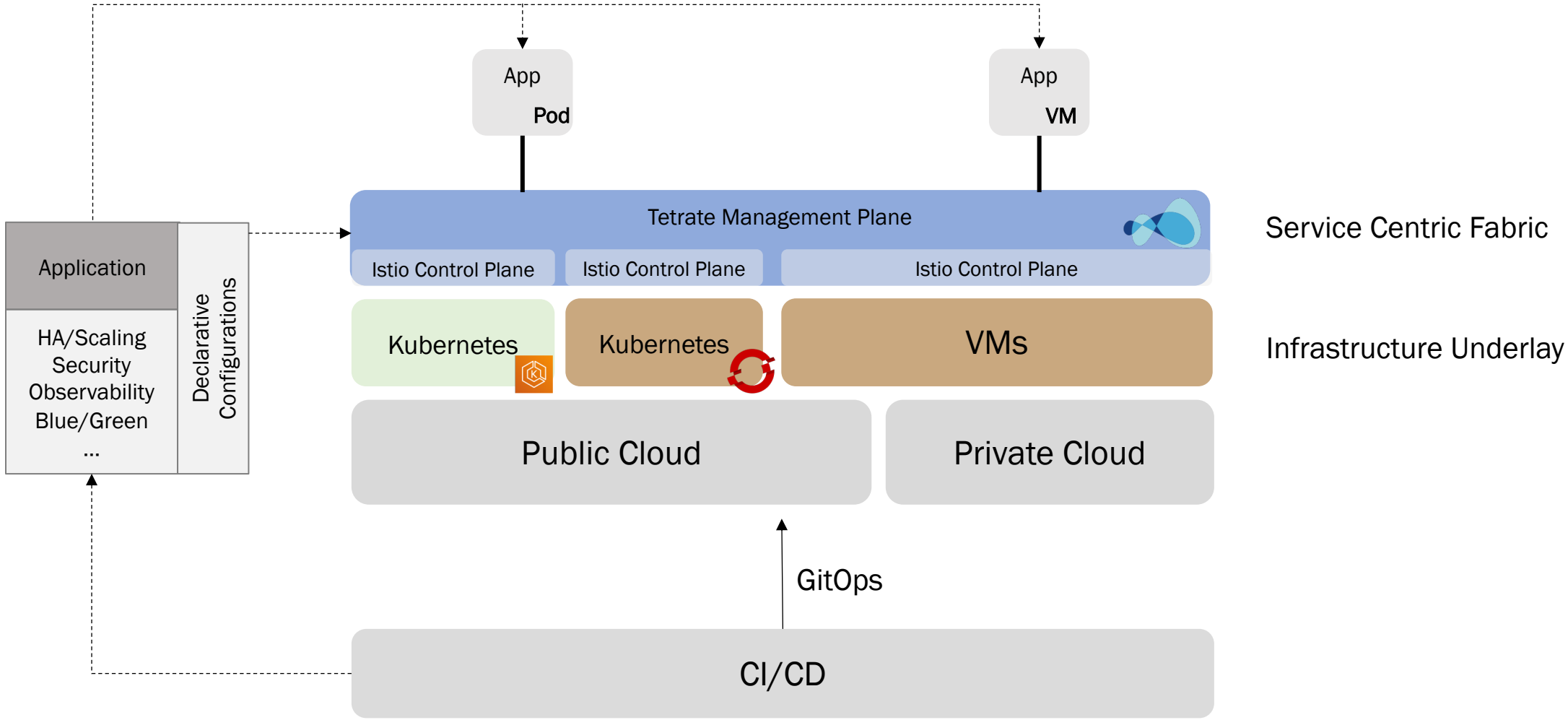


Example Use Case w/ Mesh



1. Uniform security
2. Uniform telemetry
3. Simplified routing

Software Defined Infrastructure



Lessons Learned

- Incremental adoption
  - Don’t boil the ocean by getting rid of all legacy tech.
  - Let people get comfortable and opt into features over time
  - Focus on things that require minimal code change at start
    - E.g., starting with observability, then mutual TLS, followed by routing and policy enforcement

- Do not use org Root CA for the mesh
  - Intermediate CA issued per citadel needs to be stored in FIPS compliant hardware and not in memory
    - An HSM over the network creates SPOF

Lessons Learned

- Mesh first, containerize next
  - Use the mesh to migrate from brownfield to greenfield
    - You simply need to get the sidecar into the VM to get started
  - Enables seamless fallback to VMs during migration
  - Provides observability over all traffic in the system

Lessons Learned

- As the platform owner, focus on Developer Experience for best buy-in
  - Abstract Istio APIs from developers
    - Istio APIs are infrastructure APIs – compositional in nature, providing building blocks for a variety of things
    - You don't want developers to learn yet another technology on top of Kubernetes – increases friction
    - You don't want your developers routing/securing traffic in ways you don't understand
  - Plan for CI/CD integration with the mesh from the start

- If you own the platform initiative, figure out your daily support strategy
  - Your developers will come to you first before going to the Vendor
  - Harder than you think :)

Lessons Learned

- Trusted Partners & Community
  - Consult on architecture
  - Leverage expertise to build brownfield integrations