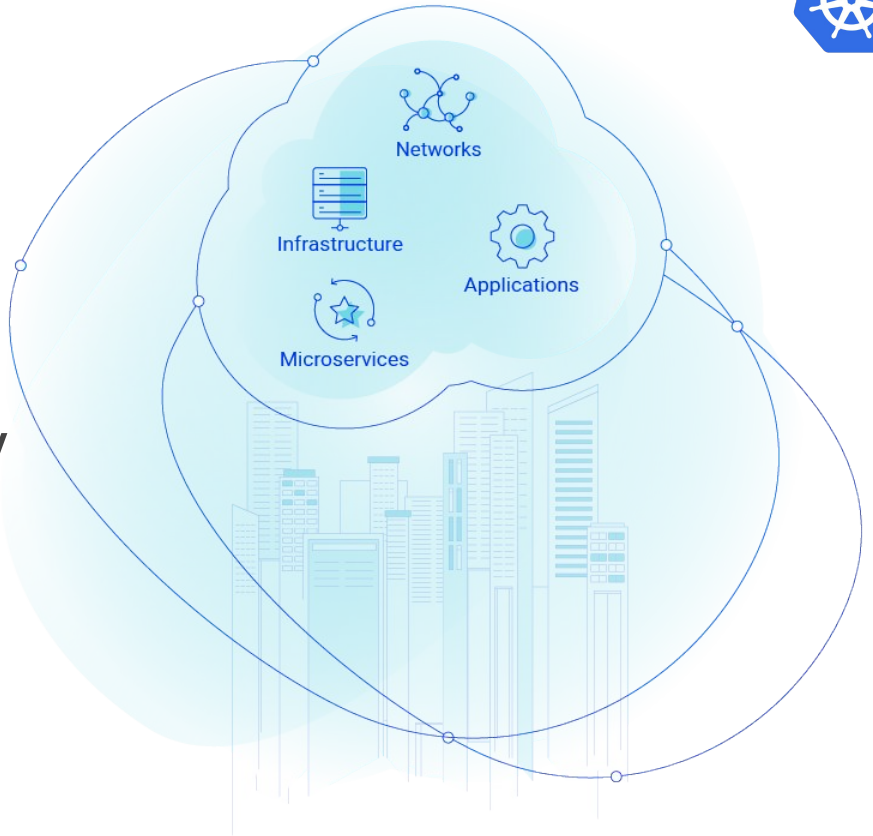
alcide



Security of the mesh and in-mesh security

Alcide x CNCF, March 2019

Gadi Naor, CTO

CLOUD NATIVE
COMPUTING FOUNDATION

```
about# gadi.naor
  --from tel_aviv
  --enjoy skateboarding
  --engineering @check_point
  --engineering @altor_networks
  --engineering @juniper_networks
  --cto alcide
  --since 07.16
  run
```

alcide

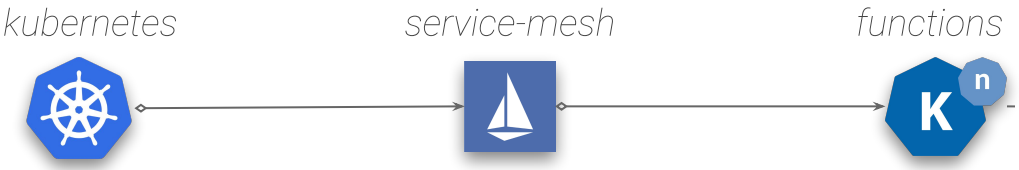
In This Webinar

In-Mesh Security & Off-Mesh Security

- Cloud Native Application - The New Stack
- Security & Operations of Service Mesh
- Beyond Zero Trust - Security Challenges In Cloud Native Applications
- Defense & Segmentation Strategies
- Understanding Data Leak Vectors

alcide

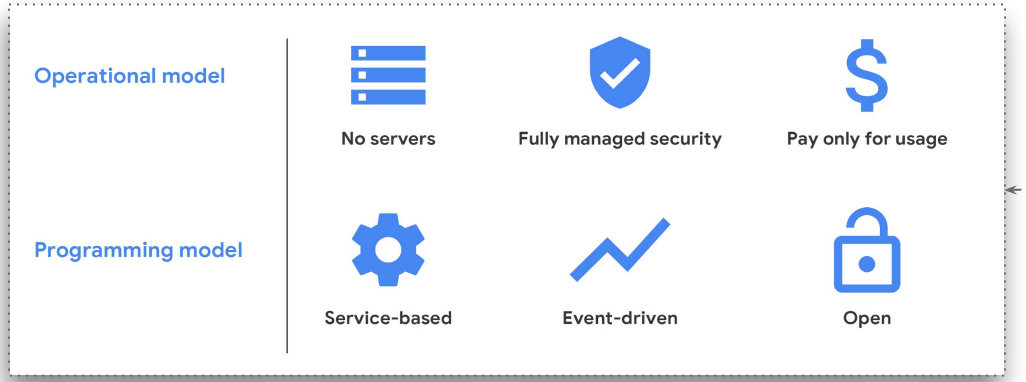
Cloud Native Applications - The New Stack



tri•fect•a

/trīˈfektə/

a bet in which the person betting forecasts the first three finishers in a race in the correct order.



alcide

The Cloud Native App Stack

```
$ Primitives for deploying and managing container workloads at scale
```

```
$ How workloads interact with each other over the network in a secure,reliable, observable way
```

```
$ “PaaS” abstractions for specifying, running, and modifying applications
```


alcide

Istio - The Promised Land

Traffic control | Observability | Security | Fault-injection | Hybrid cloud

Connect, manage and secure microservices

Connect

- Layer 7 path-based routing
- Traffic shaping
- Load balancing - A/B testing, canarying

Manage

- Telemetry
- Fleet-wide Visibility - Zipkin, Prometheus & Grafana

Secure

- Identity based service access control
- Service authorization - API level access control
- Service-Service encryption with TLS (mTLS)

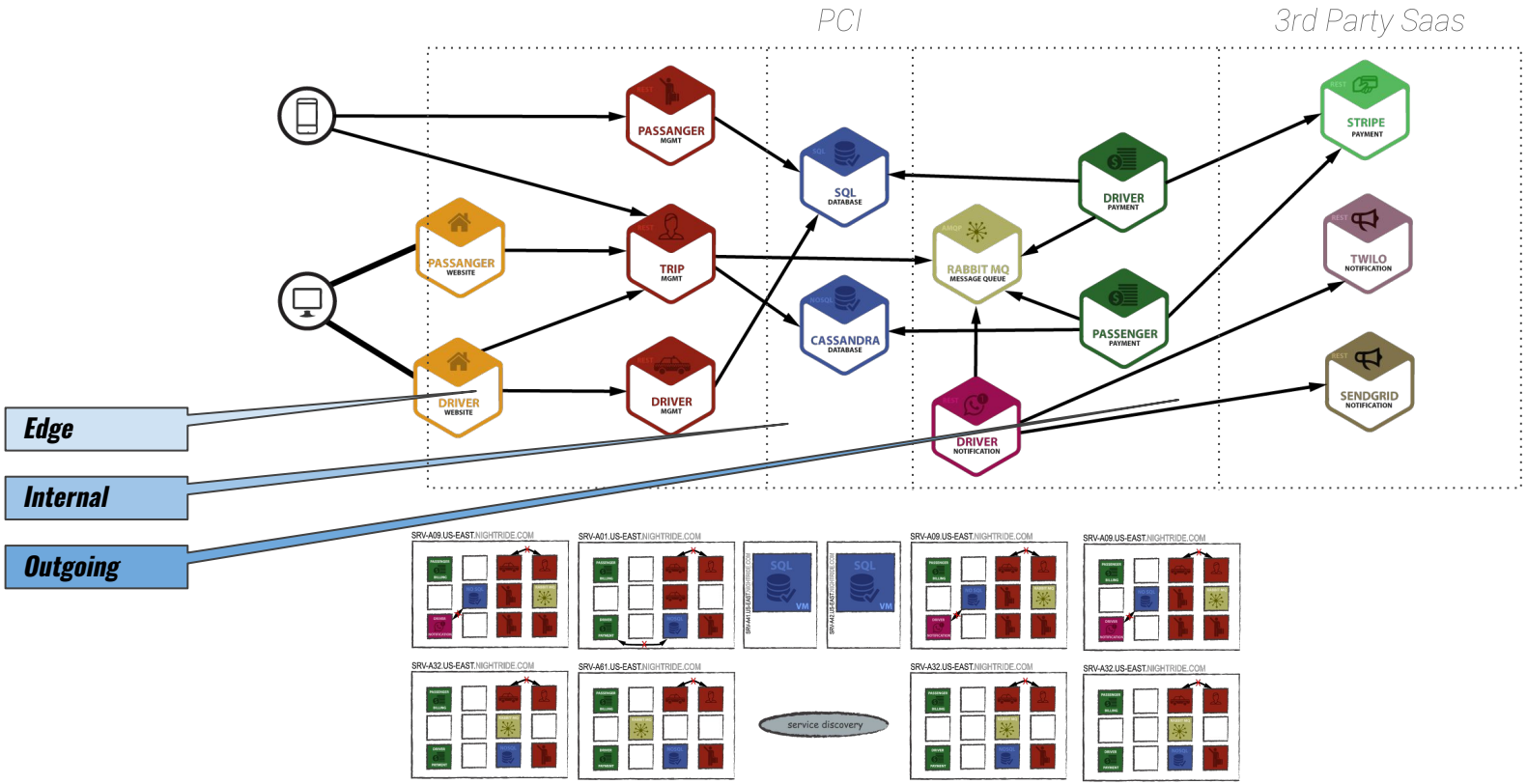
alcide

The Service Mesh

Building Blocks

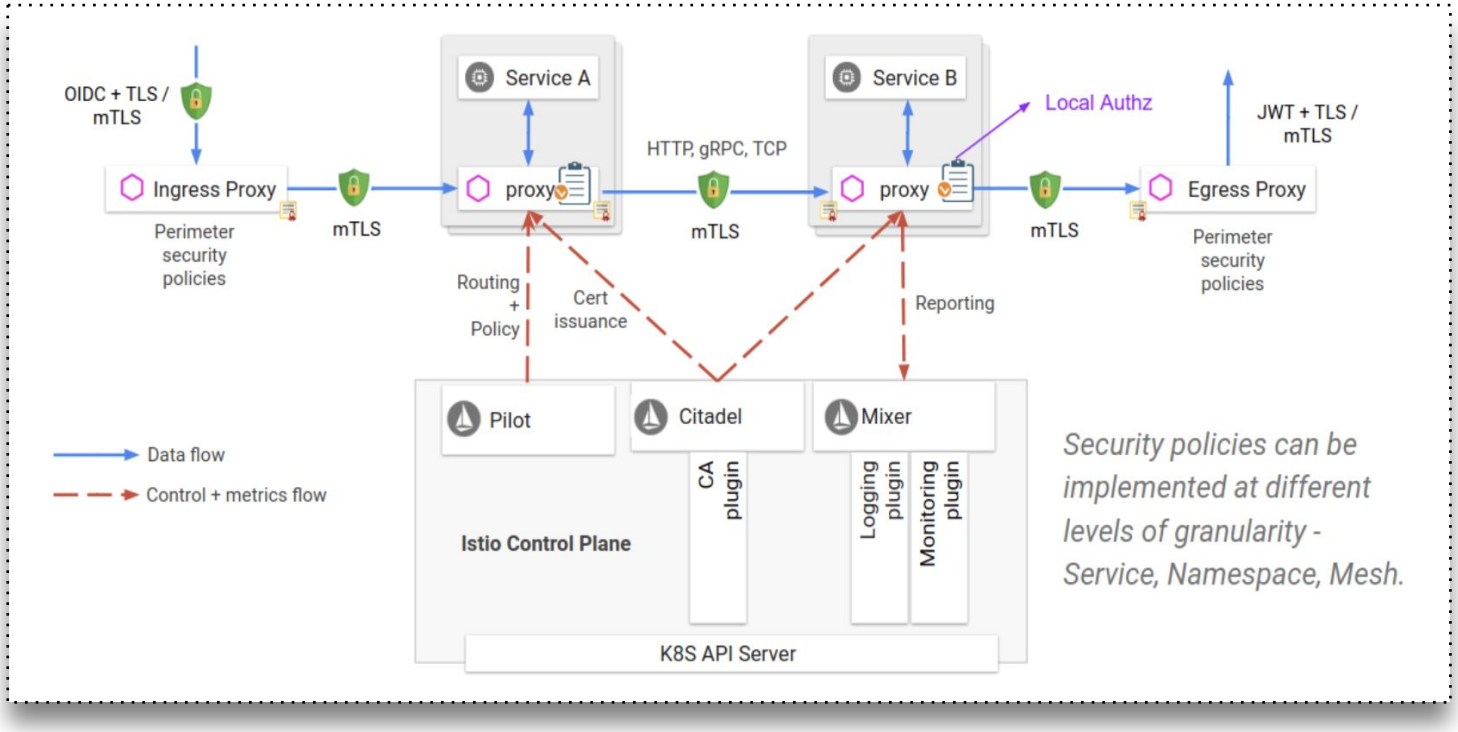
alcide

Ride The {{ Microservices }} Lightning



alcide

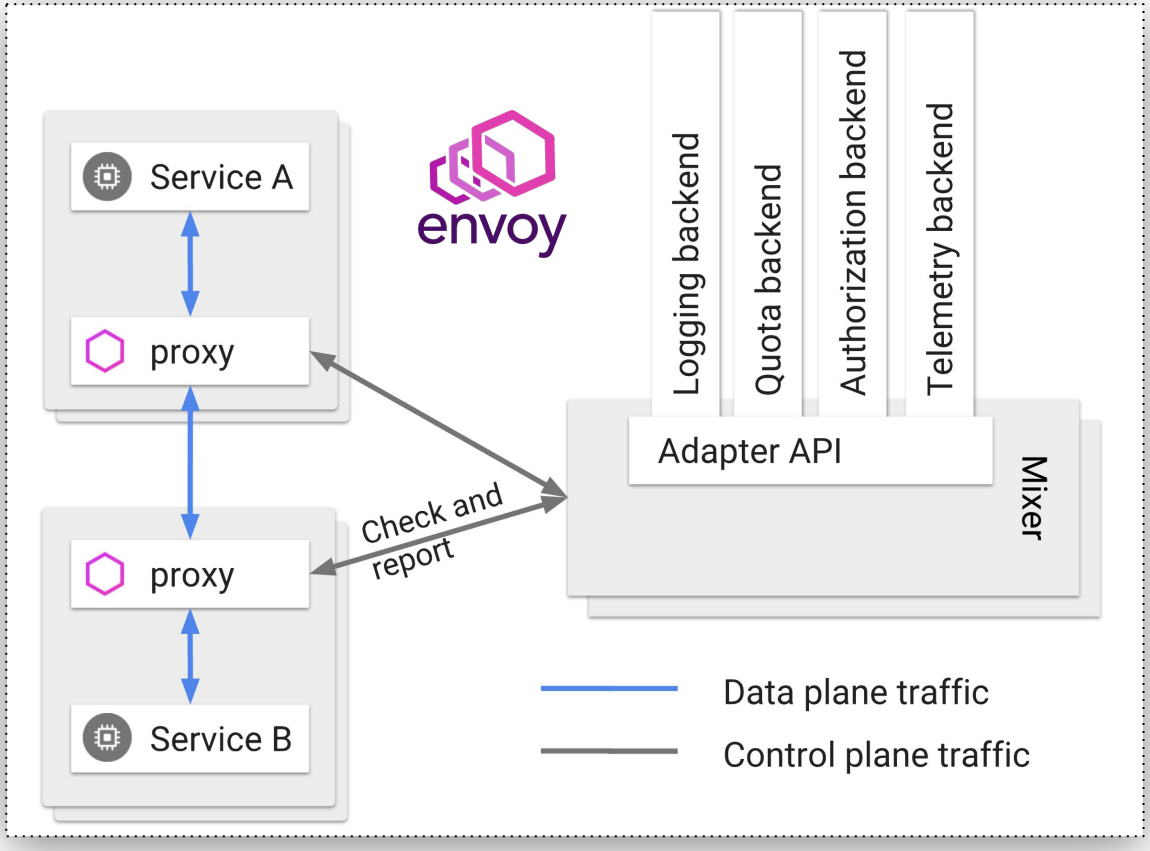
ATLS - “Istio”



https://cloud.google.com/security/encryption-in-transit/application-layer-transport-security/

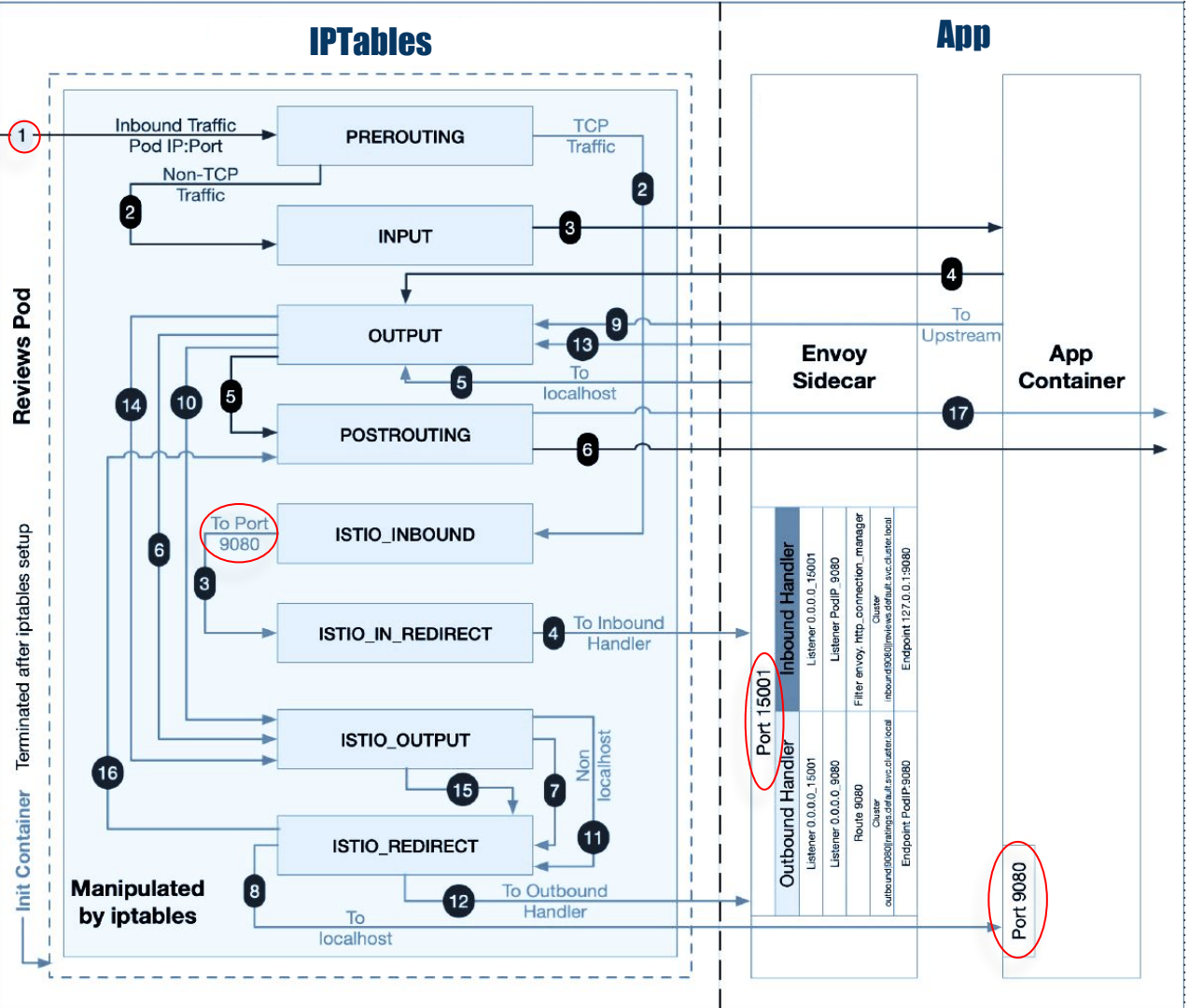
alcide

Controlling Envoy Dataplane



- Logging
- Quota
- Authz (more later)
- Telemetry

alcide



Service Mesh Dataplane

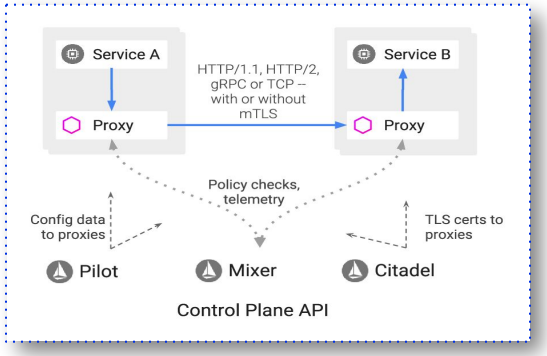
- Auto Inject @ Admission Controller
- xDS API
- Observability Pantheon
- Rich Protocol Support
  - H2, gRPC, MongoDB, DynamoDB,..
- Side Car Tax
  - Network Latency
  - CPU
  - Memory

Mesh Escape 101

```
$ kubectl  -n default exec -it ratings-v1-77f657f55d-zq248 -c istio-proxy -- bash -c "id -u"
1337
```

```
$ useradd envoyuser -u 1337 and su - envoyuser
bye bye mesh
```

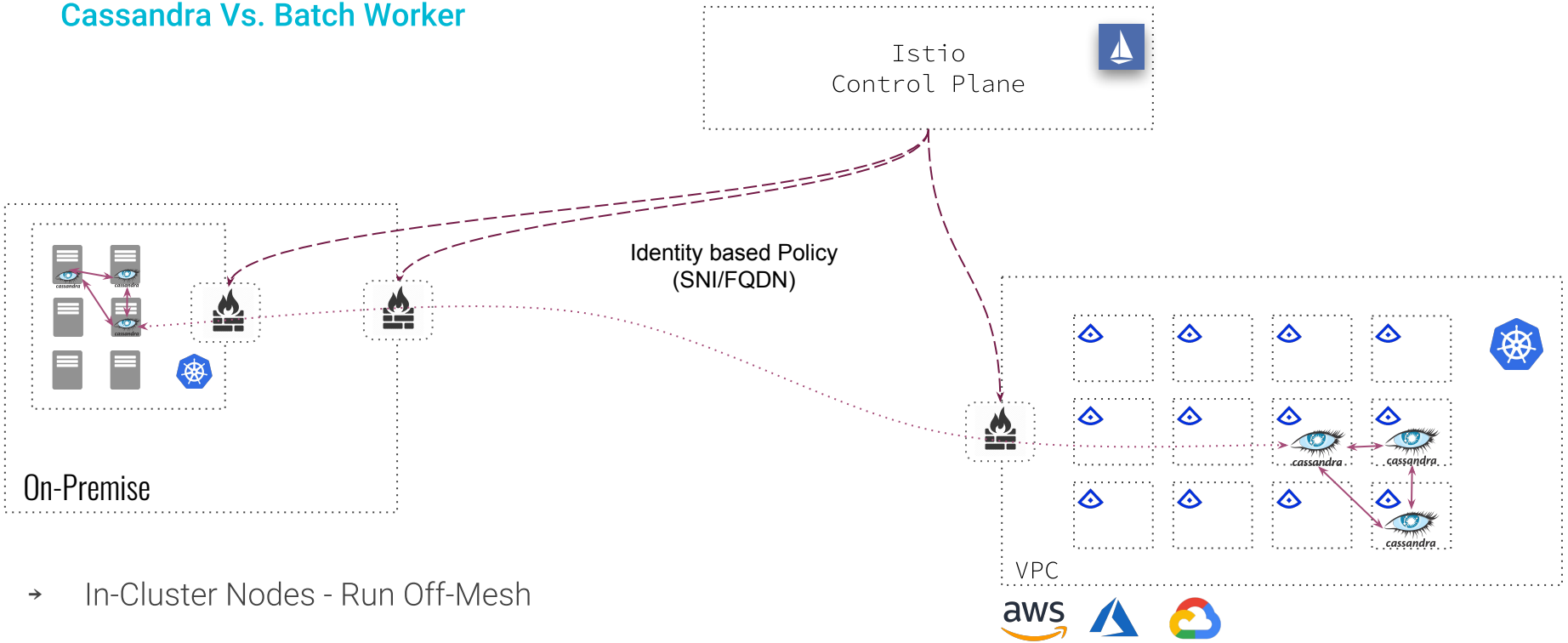
- IPv6 - east-west communications
- Tunneling (more later)
- 'Do not inject' annotation
- Does not cover infra services, control plane, ...
- Use Pod Security Policy and admission controllers for better control



alcide

Performance // Not All Services Are Born Equal

Cassandra Vs. Batch Worker



- In-Cluster Nodes - Run Off-Mesh
- Cassandra - DC Aware datastore
- Spirit: use the right tool for the right task

alcide

In-Mesh & Off-Mesh

Service Mesh Security Building Blocks

alcide

Application Segmentation aka Zero Trust

- Bring security into the heart of both network and application mesh
- Enable protection at the workload level, as well as service level
- Wrapping granular security policies around an individual workload (or group of workloads).
- Identify and prevent threats moving laterally through the network
- Add east-west protection to the traditional perimeter security model.
- Contain and quarantine threats, within the micro-segment - prevent propagation.

Good Start … Not Enough

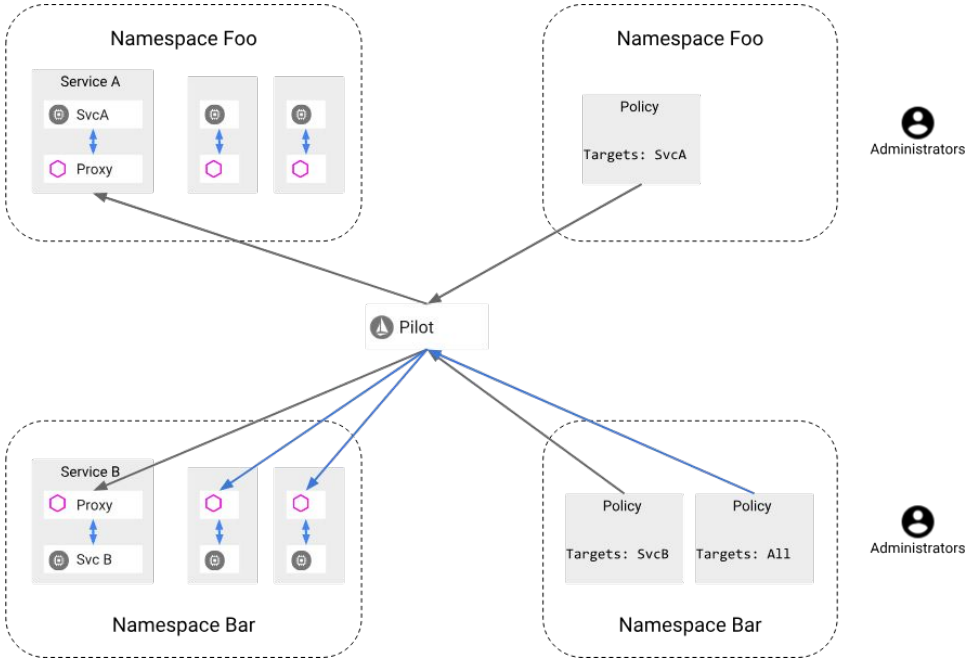
- DevSecOps & CI/CD
- Application & Infra convergence → Have an Isolation Game Plan?!
- Threat Protection - AppSec, Scans, Exploits (cpu,...), Container Escape

alcide

Istio Service Authentication

You need to:

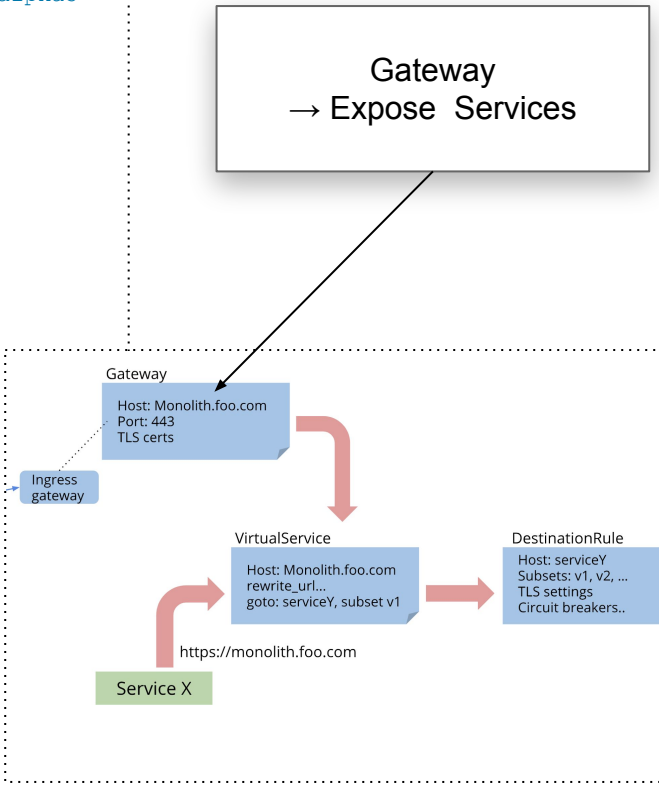
- Istio Mixer
  - Policy decision APIs
  - Service-to-Service Authentication
  - mTLS - identity provisioning
  - key life-cycle management
- End-user Authentication
  - Origin Authentication
  - Request level JWT (Auth0, Google Auth,...)



Istio Ingress & Egress Policy

Example: Grant egress access https://www.google.com

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: google
spec:
  hosts:
  - www.google.com
  tls:
  - match:
    - port: 443
      sni_hosts:
      - www.google.com
    route:
    - destination:
        host: www.google.com
        port:
          number: 443
      weight: 100
```




```yaml
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: google
spec:
  hosts:
  - www.google.com
  ports:
  - number: 443
    name: https
    protocol: HTTPS
  resolution: DNS
  location: MESH_EXTERNAL
```


alcide

Istio Service Authorization (RBAC)

Example: Grant authenticated users with read access to version v1,v2 of products service

```yaml
apiVersion:
"rbac.istio.io/v1alpha1"
kind: RbacConfig
metadata:
  name: default
spec:
  mode: 'ON_WITH_INCLUSION'
  inclusion:
    namespaces: ["default"]
```

```yaml
apiVersion: "rbac.istio.io/v1alpha1"
kind: ServiceRole
metadata:
  name: products-viewer-version
  namespace: default
spec:
  rules:
  - services: ["products.default.svc.cluster.local"]
    methods: ["GET", "HEAD"]
    constraints:
    - key: request.headers[version]
      values: ["v1", "v2"]
```

read access

condition

```yaml
apiVersion: "rbac.istio.io/v1alpha1"
kind: ServiceRoleBinding
metadata:
  name: binding-products-all-authenticated-users
  namespace: default
spec:
  subjects:
  - properties:
      source.principal: "*"
  roleRef:
    kind: ServiceRole
    name: "products-viewer-version"
```

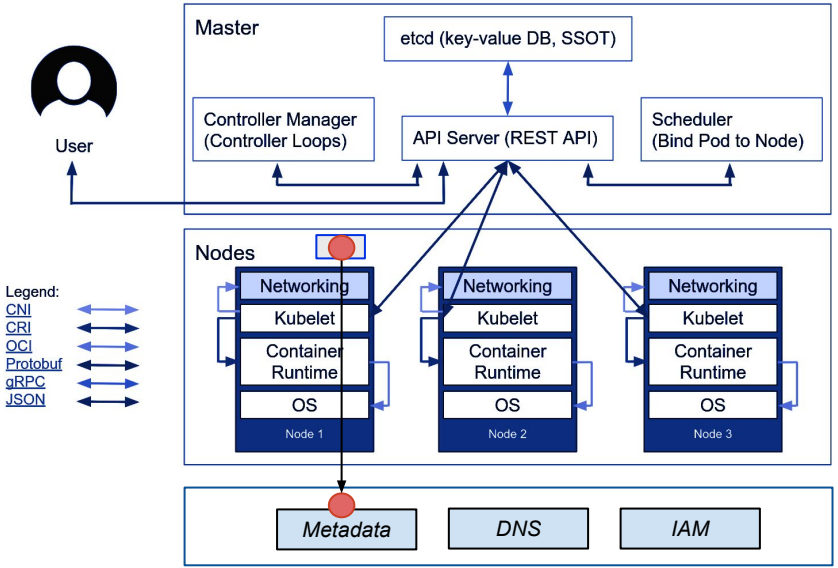
authenticated users

alcide

Exploit Chain Example

Cloud Metadata Service

1. Server Side Request Forgery (SSRF) App vulnerability in *screenshotting functionality of* ***Shopify Exchange***
2. Gain access from pod to Google Cloud Metadata
3. Obtain kube-env from cloud metadata
4. Obtain kubelet certificates from kube-env
5. Run arbitrary command with kubectl using the certs from kube-env



shopify

alcide

Off Mesh //Kubernetes Network Policies 101

With security, no news … is good news

History

pre-work Q4'15 → alpha v1.2 (Mar'16) → stable v1.7 (June'17)

Anatomy & Spirit

- Pods are not-isolated by default
- Require CNI policy support (calico, canal, weavenet, cilium)
- Multiple policies attached to Pods
- Policies "associated" to pods based on labels
- Policy rules for ingress, egress or both
  - Ingress : "Who can connect to this Pod?"
  - Egress : "Who this Pod can connect to?"

alcide

Off Mesh // Kubernetes Network Policies 101

Rules Of Engagement

- Rules are allowing traffic (whitelist)
- Rules are additive
- Container Ports only (not service ports)
- Empty groups [] implement deny
- API access level control → Istio/Api Gateway
- Policy management - hard over time
- Separate from Istio network policies

YAML Traps

```
...
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          user: alice
      podSelector:
        matchLabels:
          role: client
  ...
```

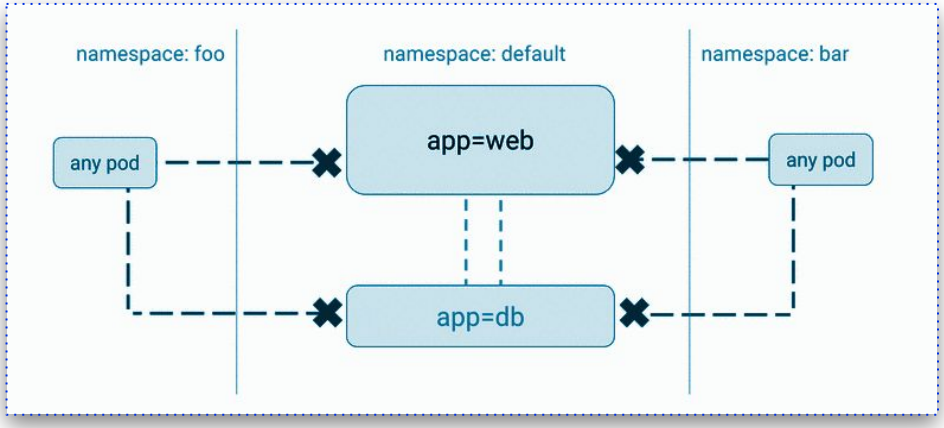
```
...
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          user: alice
    - podSelector:
        matchLabels:
          role: client
  ...
```

alcide

Off Mesh // Microservice Per Namespace

Namespace Isolation

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  namespace: my-namespace
  name: deny-from-other-namespaces
spec:
  podSelector:
    matchLabels:
  ingress:
  - from:
    - podSelector: {}
```



- Whitelist ingress traffic from namespace pods
  - Deny the rest
- Traffic dropped at the destination
  - Noisy/Hostile Neighbour

alcide

Off Mesh // Segment & Conquer

Node Taints & Pod Tolerations

- Manipulate Kubernetes Scheduler
- Force Workload placement on certain nodes

```
$ kubectl taint nodes es-node elasticsearch=false:NoExecute
```

```
…
tolerations:
     -    key: elasticsearch
          operator: Equal
          value: false
          effect: NoExecute
```

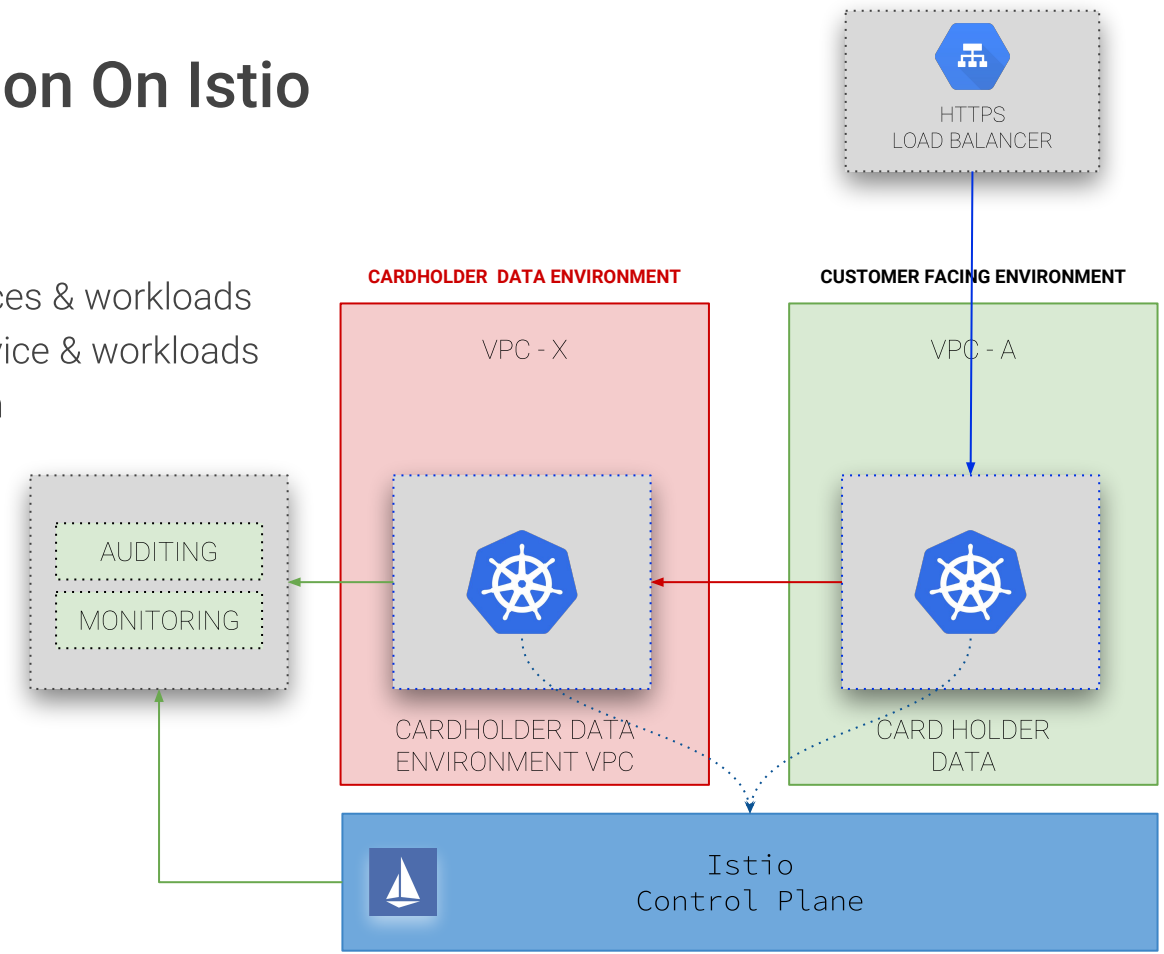

alcide

Commerce Application On Istio

Use Case: PCI DSS Compliance

- Cluster X: PCI Cardholder services & workloads
- Cluster A: Customer facing service & workloads
- In-Cluster special segmentation
- Cross Clusters & VPCs policies
- Istio Network Policies

- DevSecOps overhead
- Compute Resource overhead



alcide

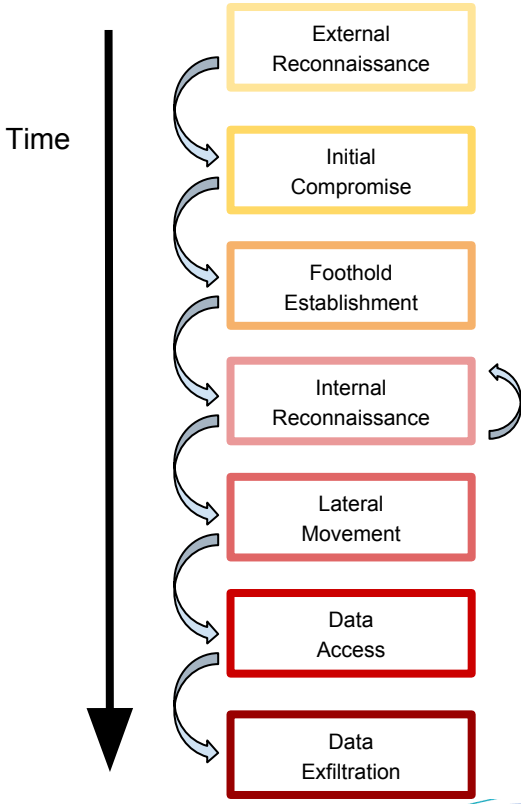
Attack Model

Istio & Service Discovery

alcide

Model of Attack on Istio & Kubernetes

1. Collect information to be used in preparing and executing the attack
2. Initial access to a deployed component
    - compromise externally-exposed vulnerable workload; deploy rogue workload
3. Prepare the compromised infiltration point as long term proxy for the attack
    - connect to Command and Control (C&C); persist
4. Identify targets, enablers or obstacles to expanding the scope of the infiltration
    - internal scanning; infrastructure API (ab)use
5. Lateral movement: infiltrate additional resources from the established foothold or from other previously compromised resources
    - pivoting, escalation; bypass security policies, firewalls…
6. Often, the target of the attack is to access sensitive information stored in data repositories
    - DBs, message-queues, files, APIs; within cluster or accessible from cluster
7. Often, data extraction is the end goal of an attack.
    - This phase is centered around perimeter-facing network connections.

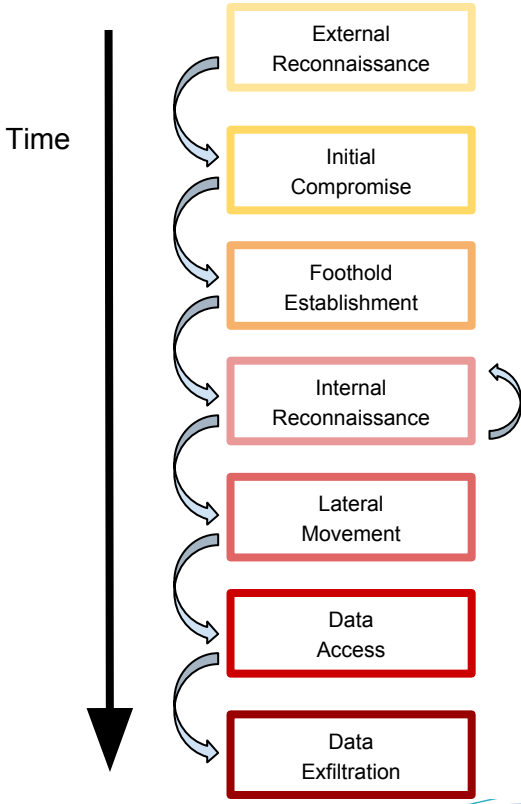


alcide

Model of Attack on Istio & Kubernetes

1. Collect information to be used in preparing and executing the attack
2. Initial access to a d
   - compromise externally-exposed vulnerable workload; deploy rogue workload
3. Prepare the compromised infiltration point as long term proxy for the attack
   - connect to Command and Control (C&C); persist
4. Identify targets, enablers or obstacles to expanding the scope of the infiltration
   - internal scanning; infrastructure API (ab)use
5. Lateral movement: infiltrate additional resources from the established foothold or from other previously compromised resources
   - pivoting, escalation; bypass security policies, firewalls…
6. Often, the target of the attack is to access sensitive information stored in data repositories
   - DBs, message-queues, files, APIs; within cluster or accessible from cluster
7. Often, data extraction is the end goal of an attack.
   - This phase is centered around perimeter-facing network connections.

Using DNS Tunneling



alcide

DNS Tunneling

- *Tunneling*: embedding messages of one network protocol within messages of a different network protocol.
- In Security Context: bypass security boundaries
  - Encapsulating malicious traffic within permitted, ubiquitous traffic
  - e.g. bypass perimeter firewall, Kubernetes/Istio network security policies
- Attacker sets up a tunnel from Internet to a compromised Pod (Workload)
  - Pass commands to it
  - Exfiltrate data harvested in the cluster through its service discover infra

alcide

DNS Tunneling // Attack and Protection

- DNS tunneling may be used by attacker to connect compromised WL in a Kubernetes cluster to Internet
  - covert channel for C&C and data exfiltration
  - bypassing perimeter firewall, cloud/k8s/istio network security policies
- Challenges in detection and mitigation
  - high-resolution monitoring: deep packet inspection per WL
  - scalable and fast analysis
  - detection using WL context
  - automatically adapt to changes of WLs and WLs activity in cluster
  - fine-grained mitigation

alcide

Zero Trust API Access Control

Definition

- Each API is protected (God Bless Envoy & Istio)
- East-West API Calls
- Authentication API Keys, Bearer Tokens, TLS Client Certificates, or HTTP Basic Authorization
- Authorization - Attribute Based Access Control

Challenges

- GraphQL, Non HTTP protocols (Kafka, C*, ...)
- Egress Controls - Workload Segmentation
- Off service mesh
- API Sprawl
- Detect Anomaly & Drifts

alcide

Istio Security

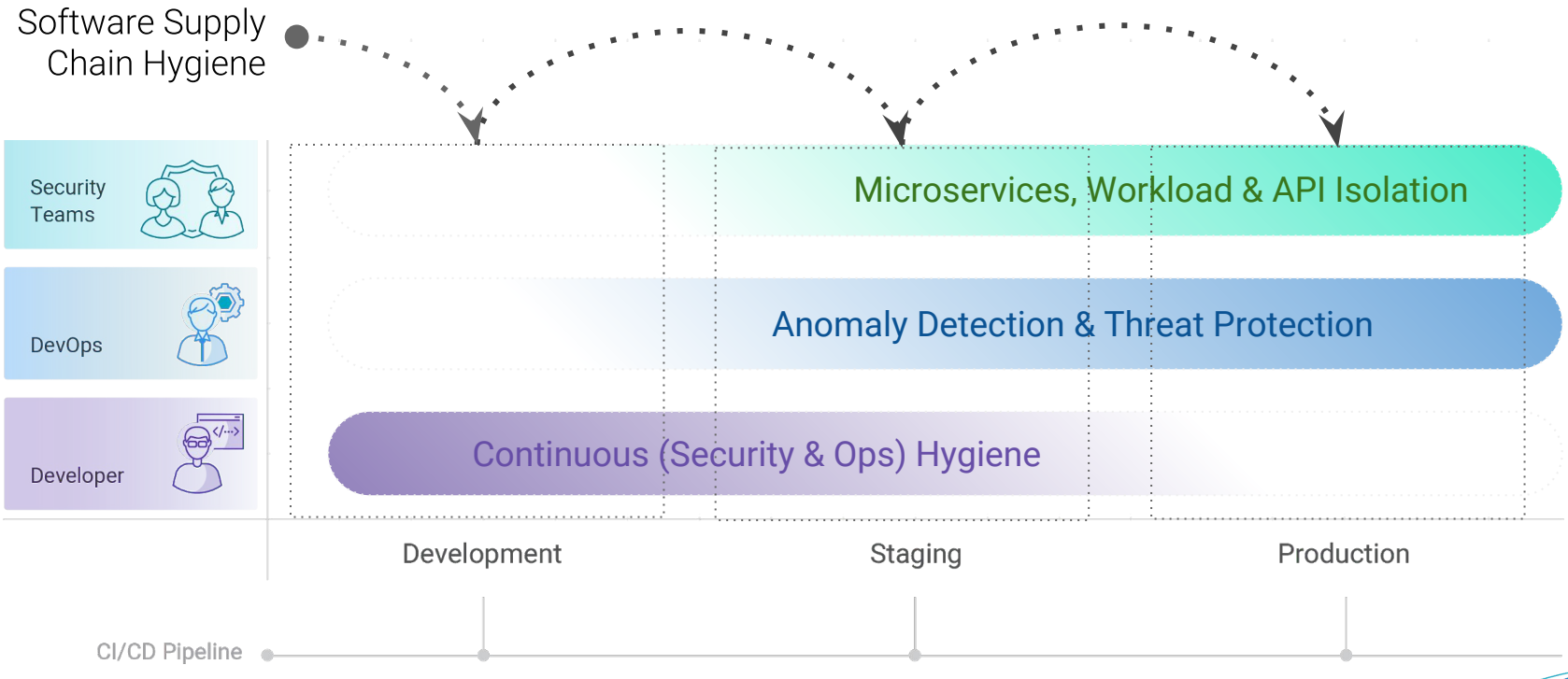
The Good, Bad & Ugly

- Pods are not-isolated by default
- Basic east-west workload segmentation between cluster workloads
- Decouple microservices and policy rules - with healthy flexibility
- Cluster nodes network policies covered by the underlying cloud security groups
- Policy management - extremely challenging over time.
- You can’t embed policy into workloads/microservices
- No explicit way for drop rules
- Dropped traffic is not logged anywhere
- Full network security stack (cryptomining, scans, tunneling, spoofing, application attacks,...)

alcide

Complexity Is The Worst Enemy For Security

SMeshOps - Mitigate Challenges Collectively



alcide

Q & A

gadi@alcide.io

get.alcide.io/request-demo

CLOUD NATIVE COMPUTING FOUNDATION

Thank You

gadi@alcide.io

get.alcide.io/request-demo

CLOUD NATIVE
COMPUTING FOUNDATION