Istio: zero trust communication security for production services

Samrat Ray

Tao Li

Mak Ahmad

Emerging trends driven by need for agility, speed, cost, data volumes …

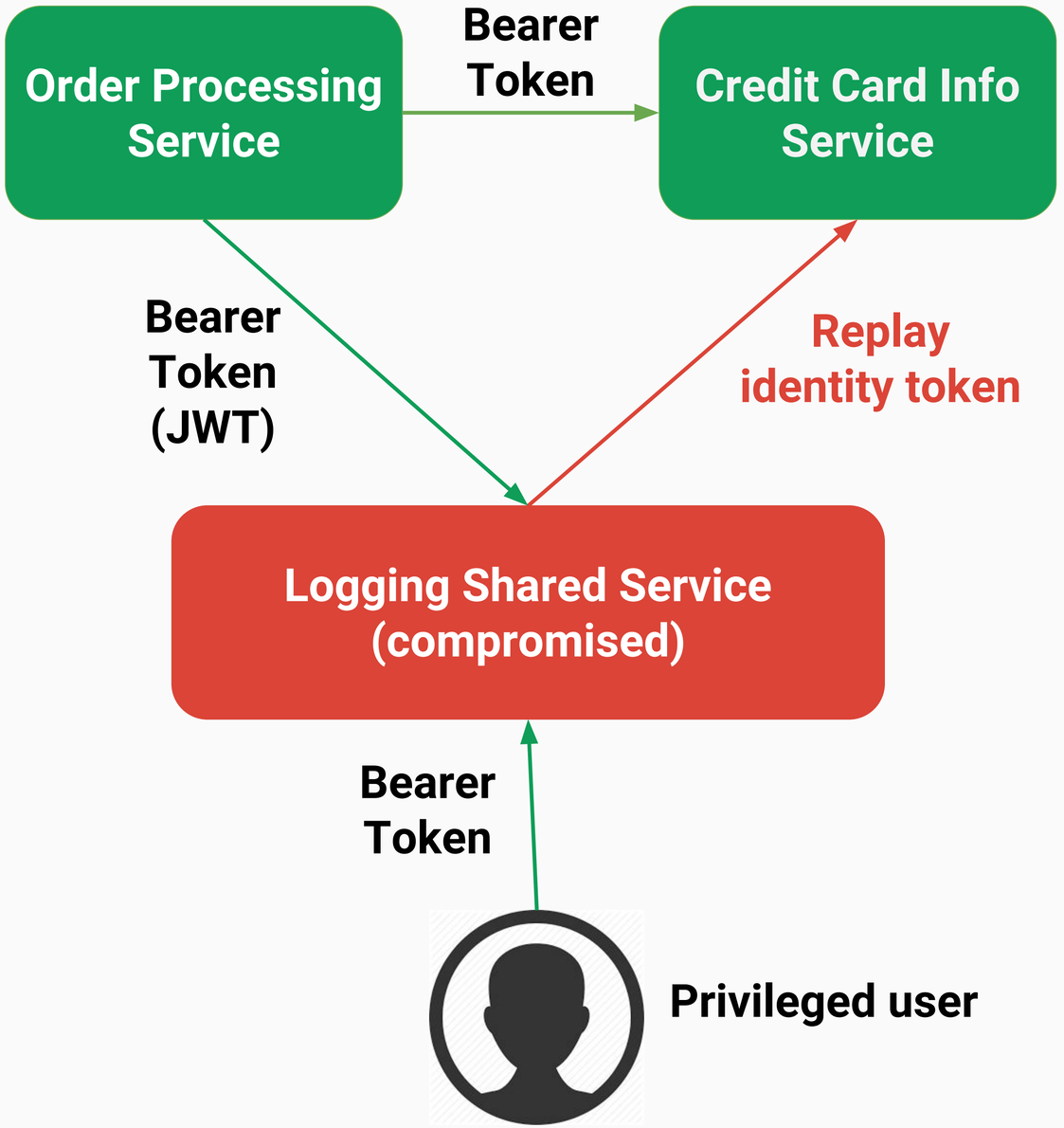
Containers + microservices

Public cloud, multi-cloud and hybrid architectures

Heterogeneous app stacks - languages, PaaS, OSS

How does it impact security?

- Significantly **greater surface area** to attack within the network if the perimeter is breached.
- Compromise of shared services enables **replay attacks**.
- Higher risk of indirect unauthorized access using **stolen identities** or by malicious insiders.



Mitigating risk in modern production environments

The zero trust security model for communication security

1. **Assume threat already exists within the network**
   A peer (service or client) should not be trusted just because it is in the same network even if the client presents valid application level credentials.

2. **Ensure every service has its own firewall (micro-perimeter)**
   All access - human or programmatic - to production services should be strongly authenticated, authorized and encrypted.

3. **Enable central administration and monitoring**
   Security teams should be able to understand the overall security posture and manage mandatory security policies across all services.

Google has adopted such a posture for many years ...

BeyondCorp at Google

- Enterprise security model that builds upon 6 years of building zero trust networks at Google.
- Improves security with regards to how **employees and devices** access internal applications.

Application Layer Transport Security at Google

- Transport security based on strong mutual authentication of peer vs. trusting the network.
- Optimized for Google's **production services**.
- Other design principles:
  - transparent from the application
  - long lived connections

But how do you implement
a similar posture
at scale
while retaining
the **deployment agility** and
stack heterogeneity
enabled by modern production environments ?

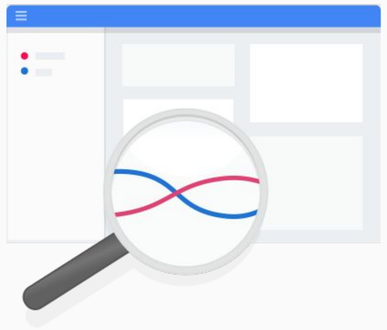
... drum roll ...

Istio is an open services platform
that incorporates the
principles and learnings
from securing Google’s production
with technologies like ALTS.

What is Istio?

An open services platform to manage service interactions across container- and VM-based workloads

What does Istio do?



Observability

Latency, errors, dependencies, tracing

Control

L7 load-balancing, rate limiting, safe rollouts of new versions

Security

Encryption, strong authentication, authorization

What can you do with Istio security?

- Enable mTLS for authentication and encryption.
- Authorize access based on service identity or any channel attribute.
- Configure finer grained RPC-level access control for REST and gRPC.

Why do we support mTLS via Istio?

Policy driven encryption in transit

with no code changes

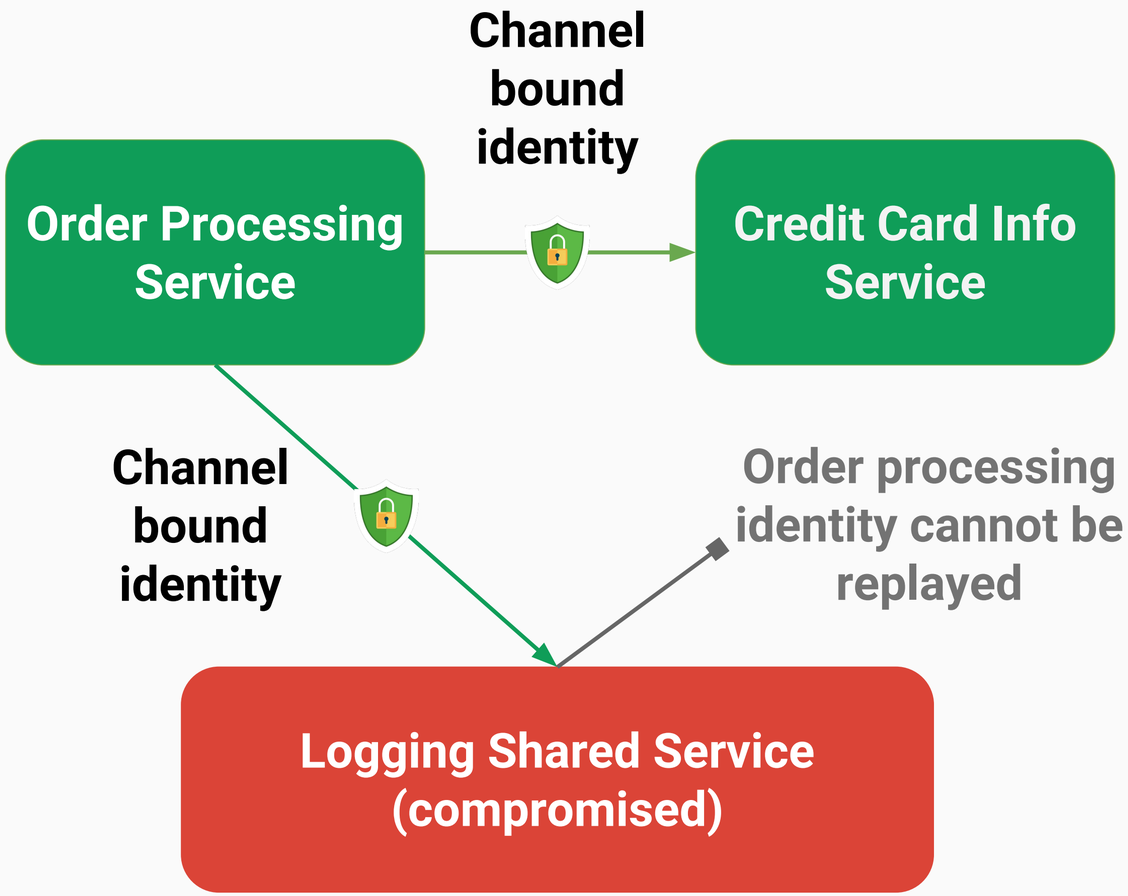
… but that’s only the obvious value ...

… the real value is strong authentication

Peers are authenticated using **non-replayable service identities** bound to the TLS channel.

Similar to ALTS, Istio strongly authenticates the **workload identity** and not the host.

End user or application level identity is propagated as a bearer token across service “hops”.



Service Authorization

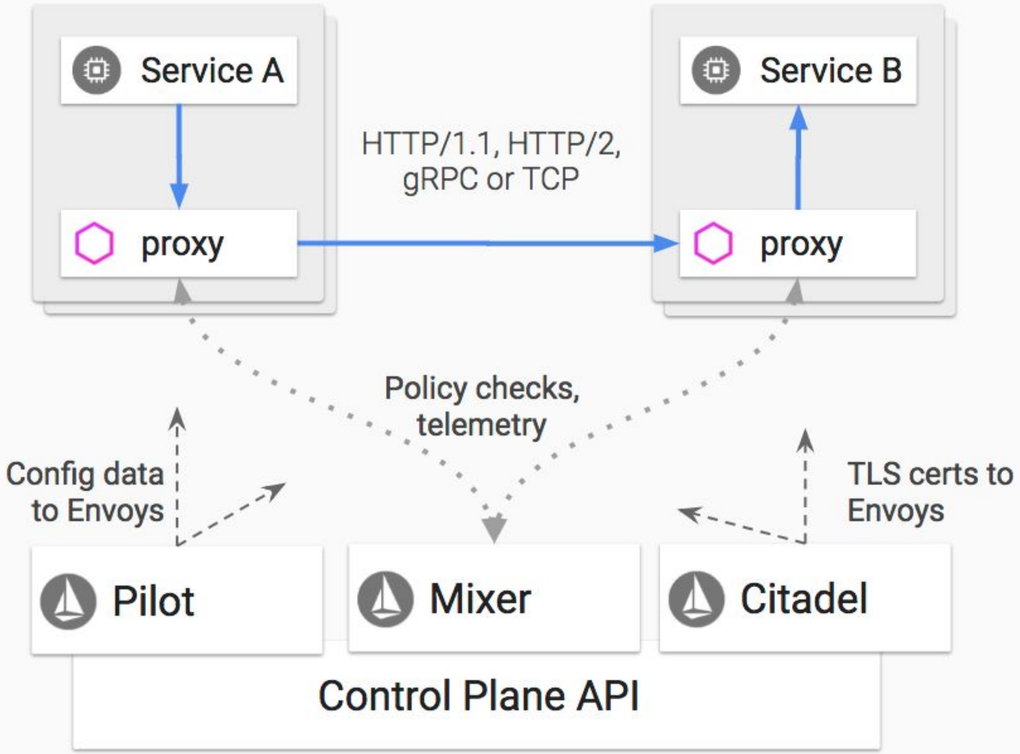
Istio enables **identity and context aware network security policy** enforcement.

Policies based on service identities are **resilient to redeployment of peers**.

REST / gRPC: Finer grained access control and ability to **reason about both peer and end user** identity.

Layer	Policy Input	Applicability
Application (future)	Declarative resource model	First party applications.
API (RPC)	Operations: URI, verbs JWT Claims: End User Id	HTTP / REST gRPC
Channel	Service identity (Cert) IP Context: K8S Namespace	TCP based protocols

Istio Component Architecture



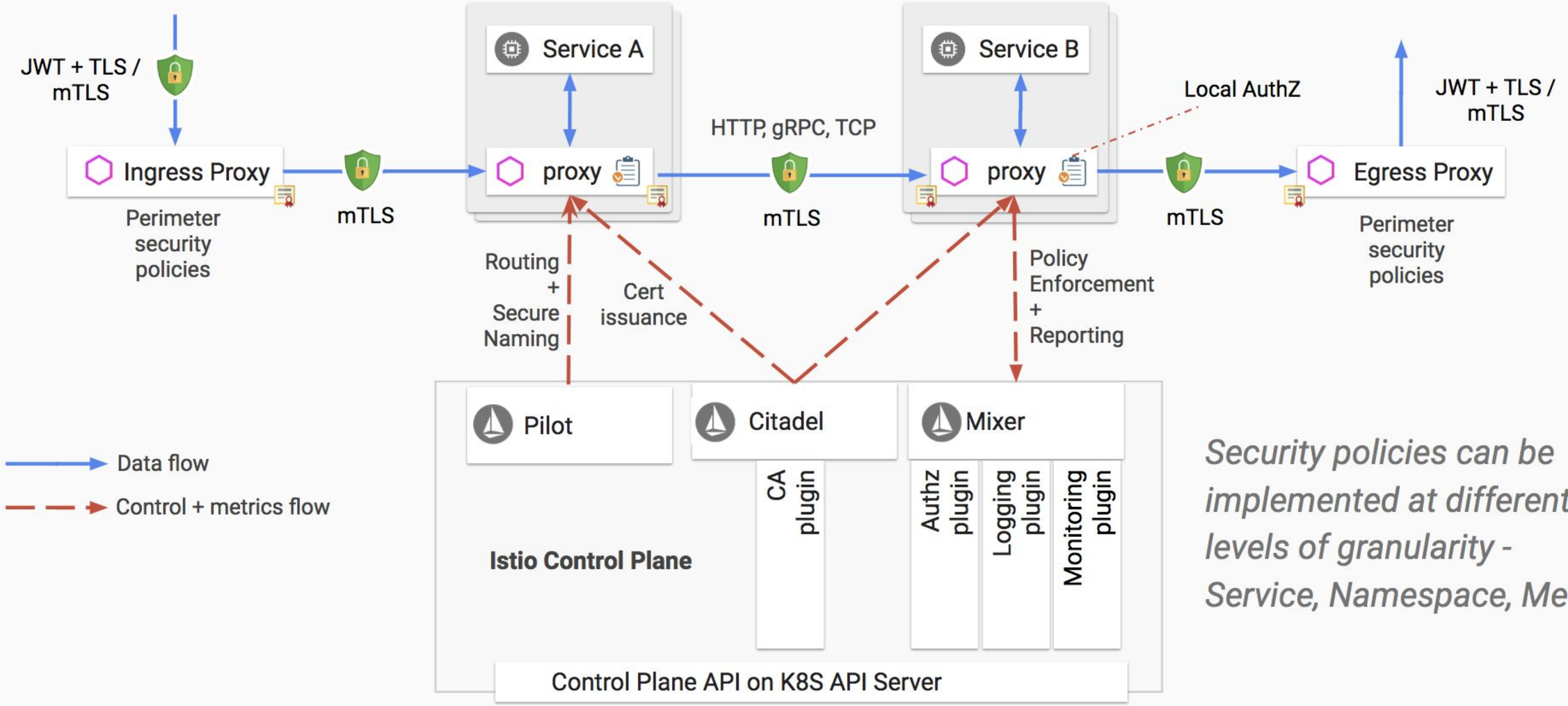
Envoy: Network proxy to intercept communication and apply policies.

Pilot: Control plane to configure and push service communication policies to Envoys.

Citadel: Service-to-service authentication using mutual TLS with built-in certificate and key management.

Mixer: Flexible model for policy enforcement and monitoring with a plugin model.

Istio Security: Life of Request



Security policies can be implemented at different levels of granularity - Service, Namespace, Mesh.

Demo Agenda

Policy based mTLS enablement

Enforce strong peer authentication + encryption.

End-user identity verification

Assert that client is acting on behalf of an authenticated end user.

Peer identity based Authorization

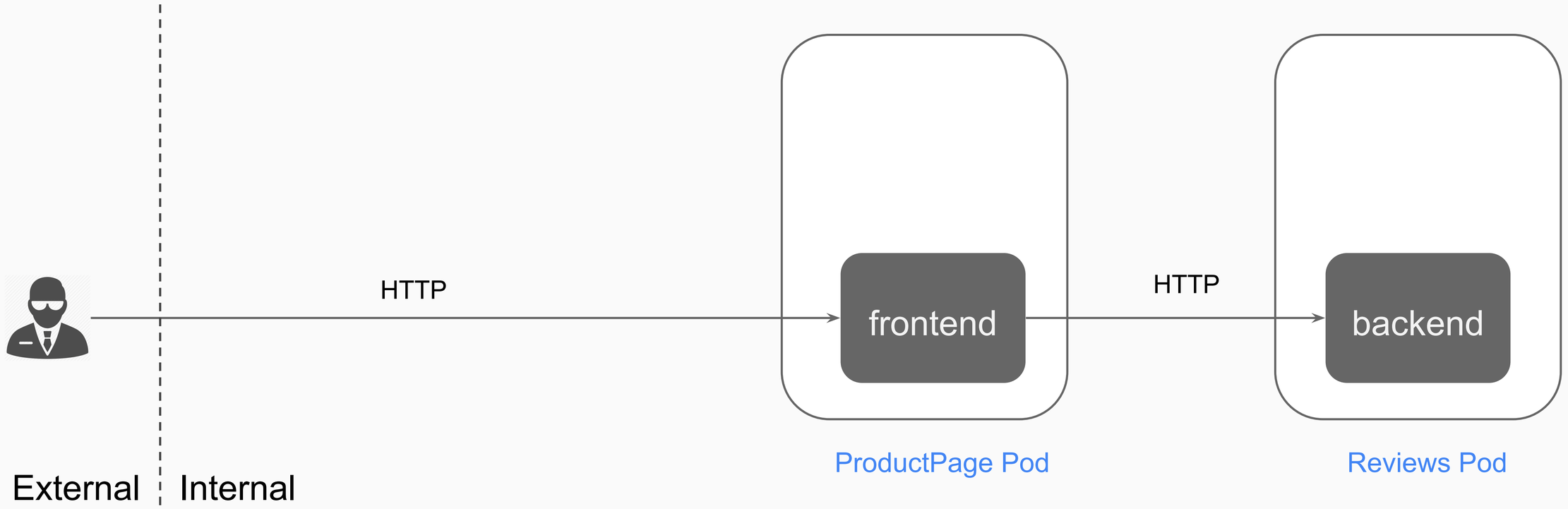
Deny non-whitelisted services even with the right application level credentials (bearer token).

Monitoring

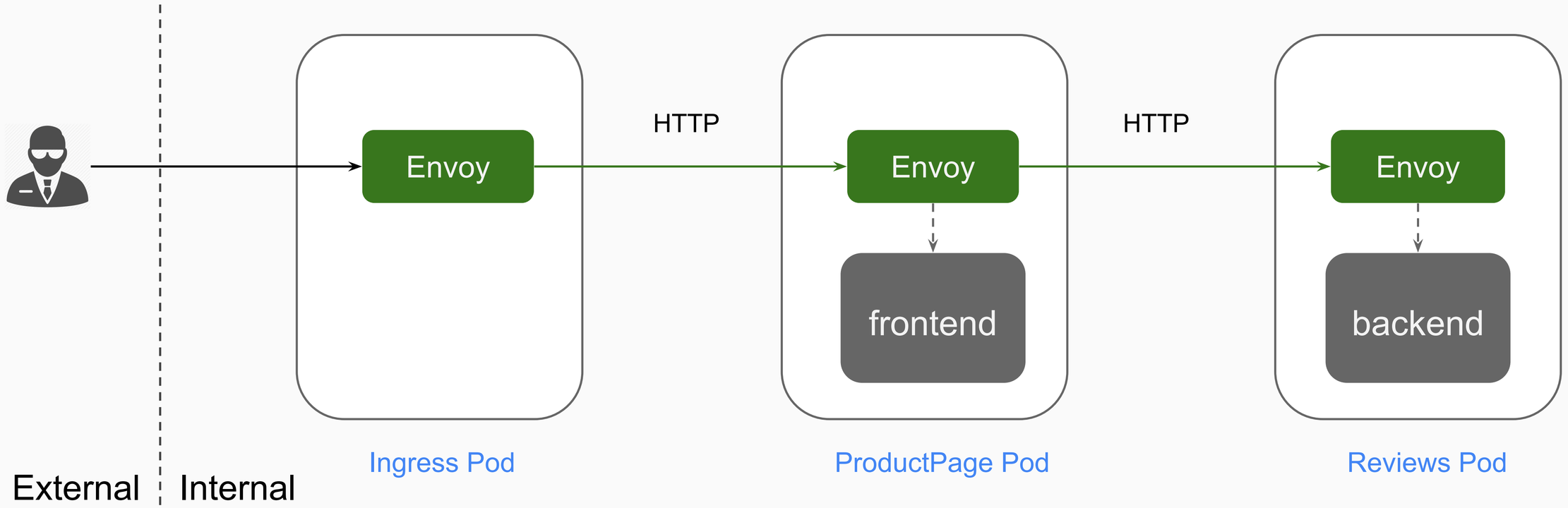
Centrally manage and monitor all communication.

Two-Tier Example

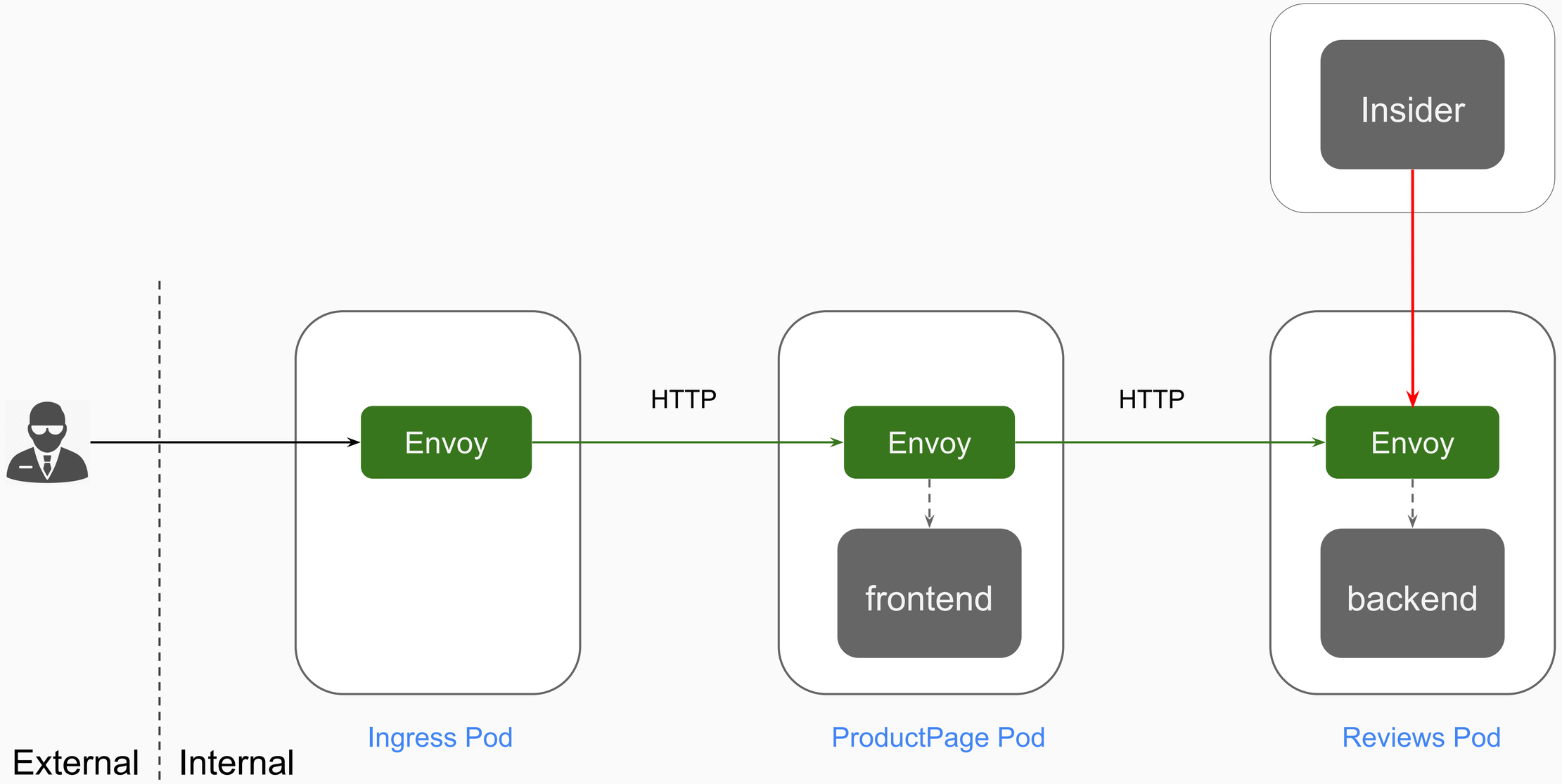
No Traffic Encryption
No Access Control
No Visibility



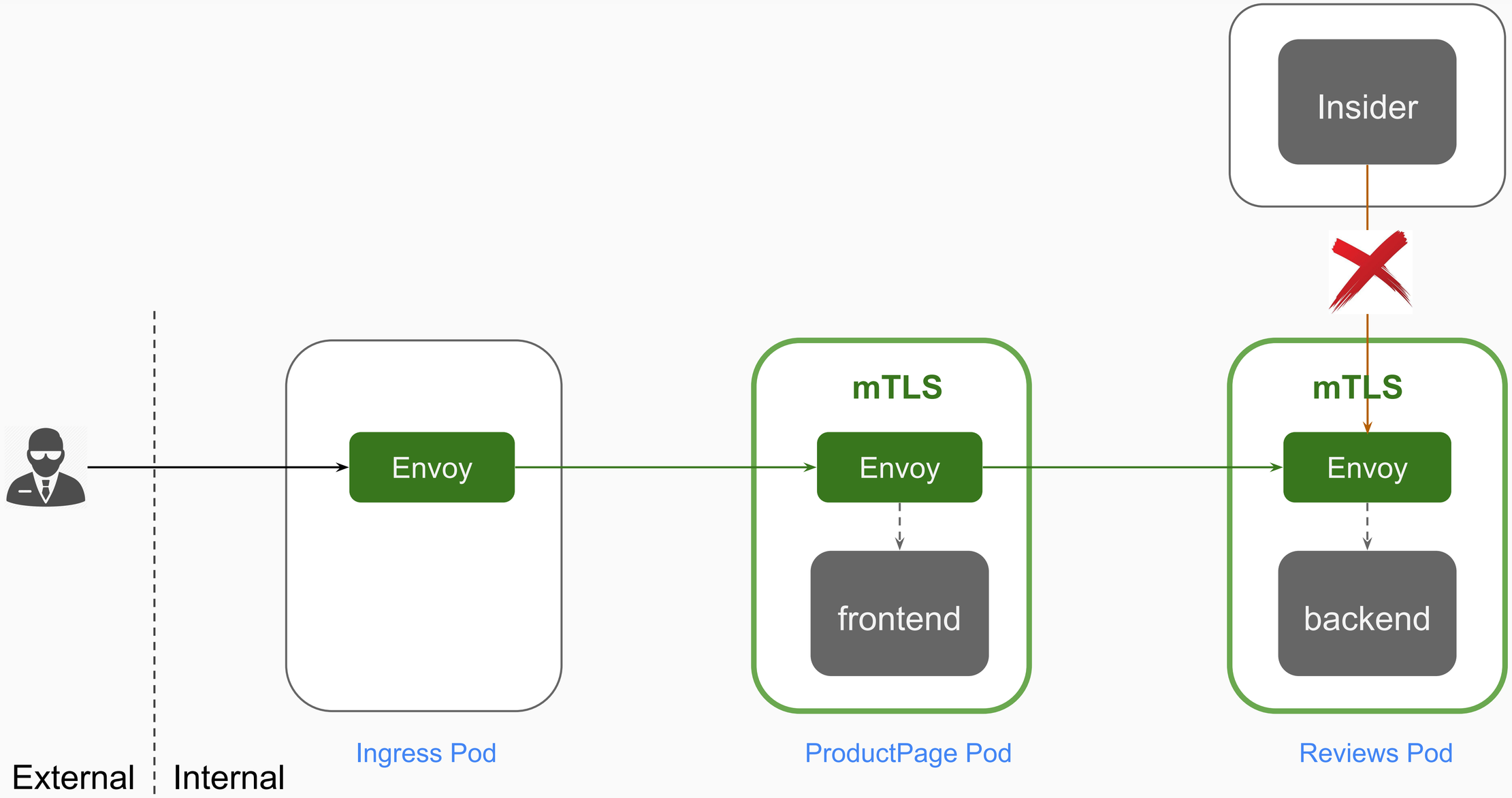
Istio **without** mTLS



Istio **without** mTLS: Insider can access with app level credentails

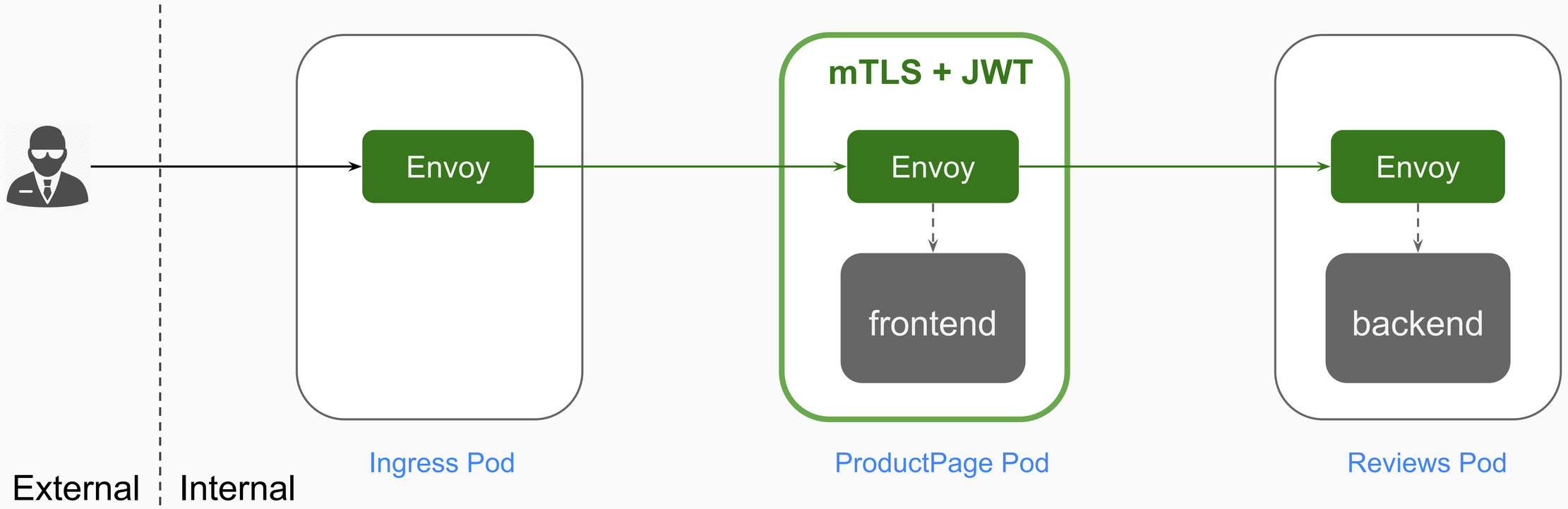


Istio with mTLS: Can only access with valid key/cert

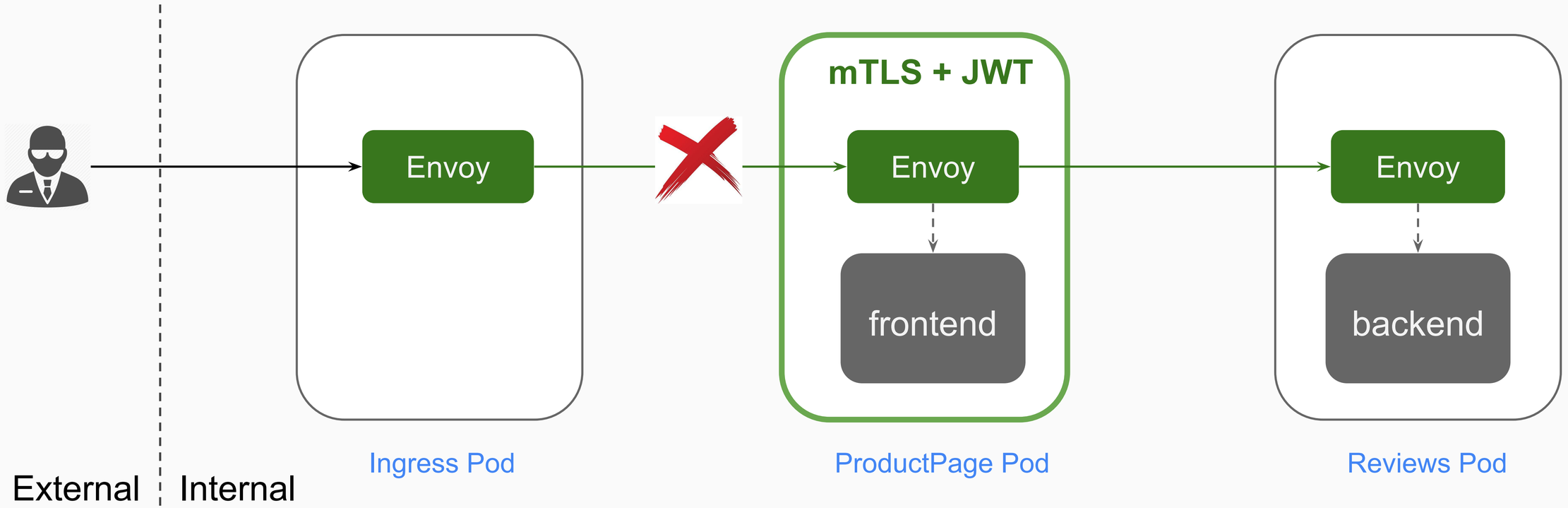


Validate that front end is acting on behalf of authenticated end user

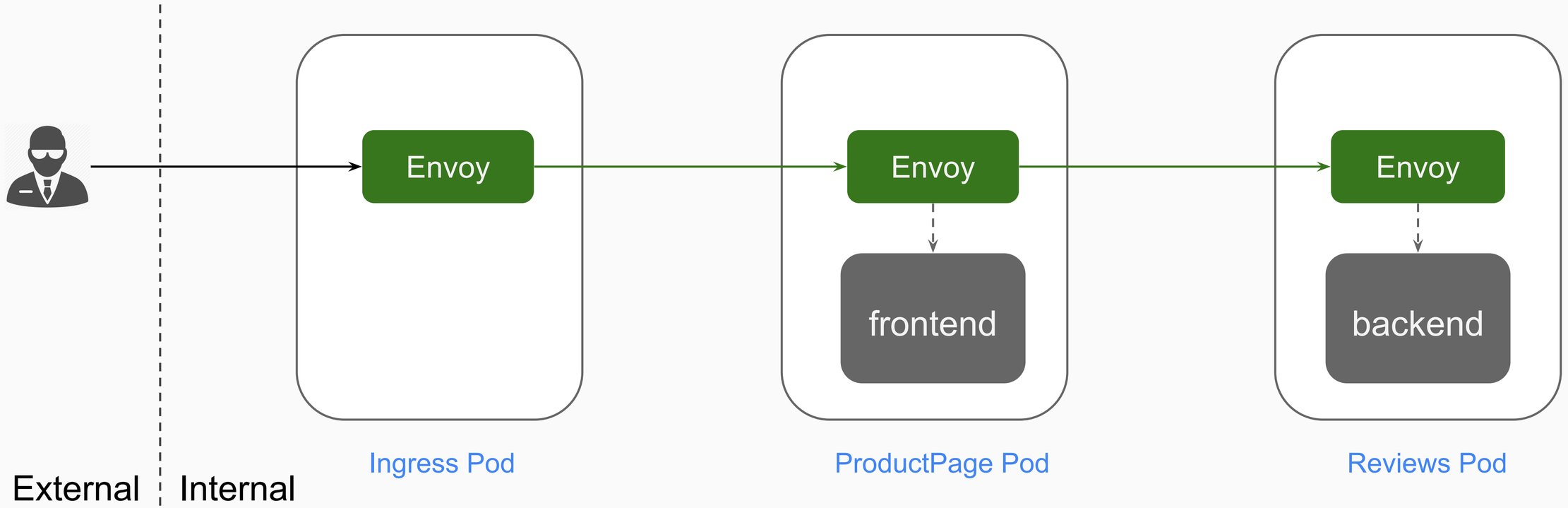
Backend expects 2 credentials



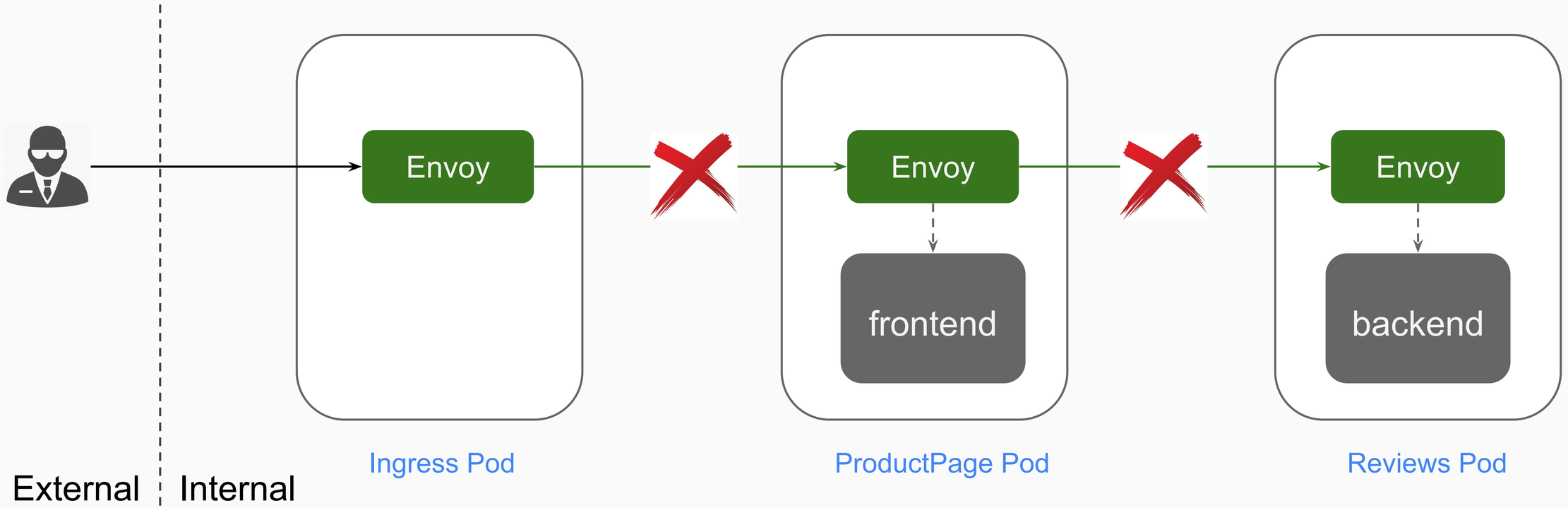
Istio ensures that request has valid user credential in additional to an authenticated peer



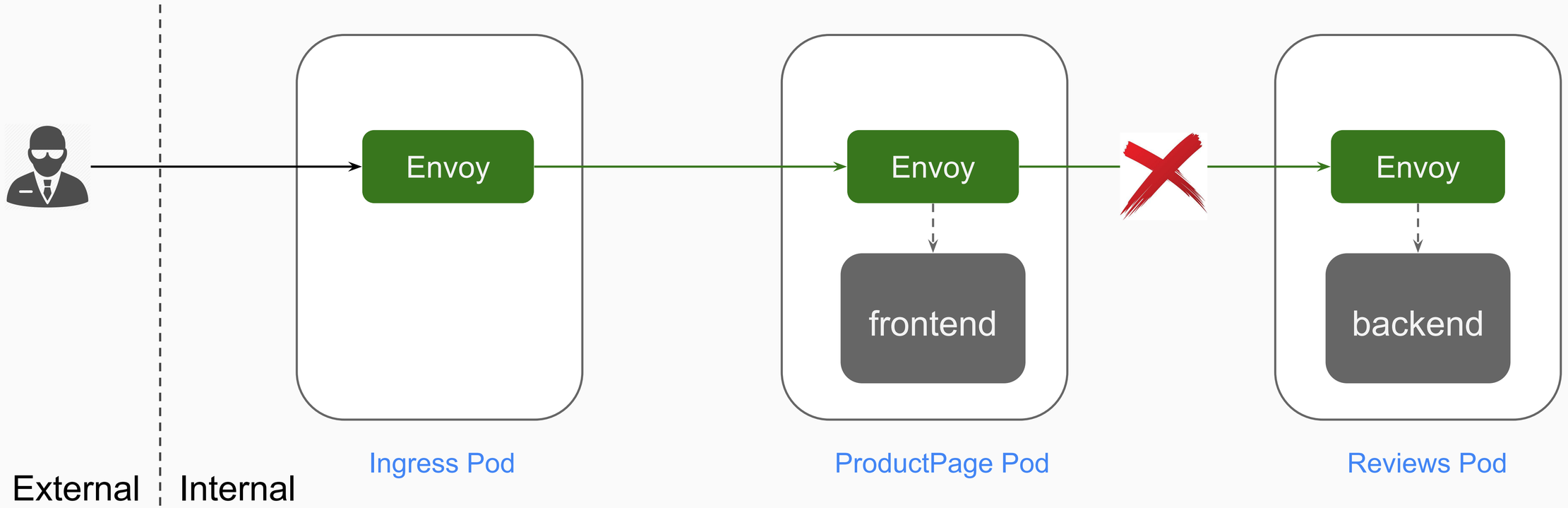
Istio Authorization



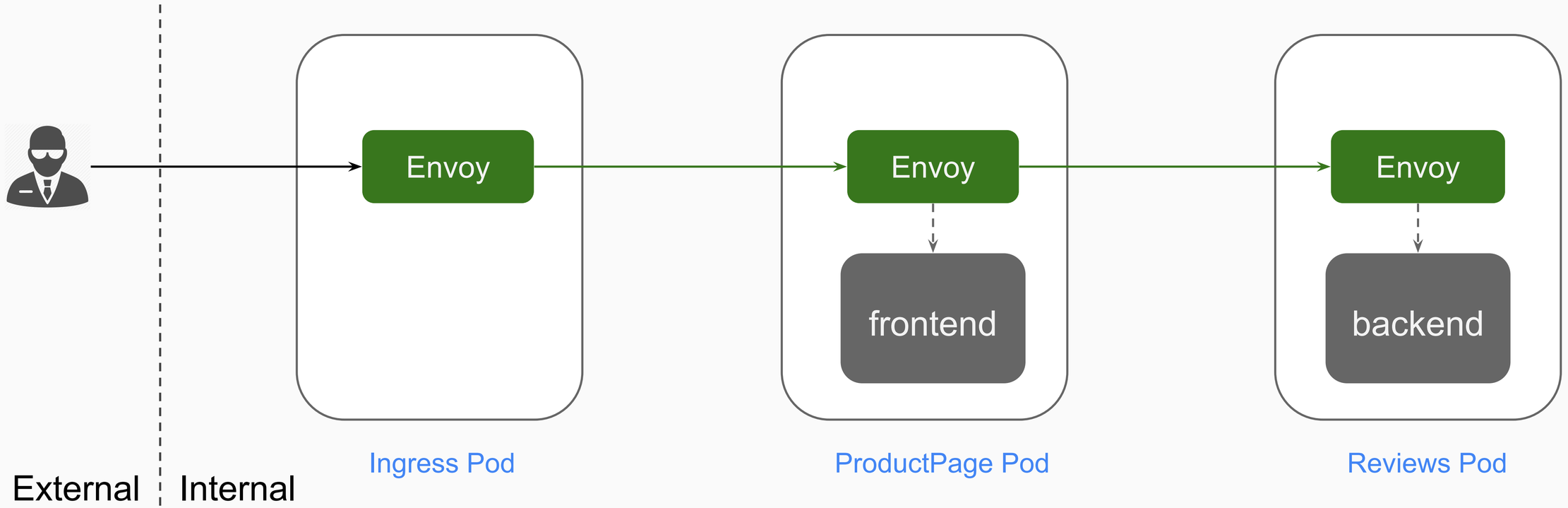
Istio Authorization



Istio Authorization



Istio Authorization



How to engage with Istio Security

Users

Try out Istio on Google Cloud

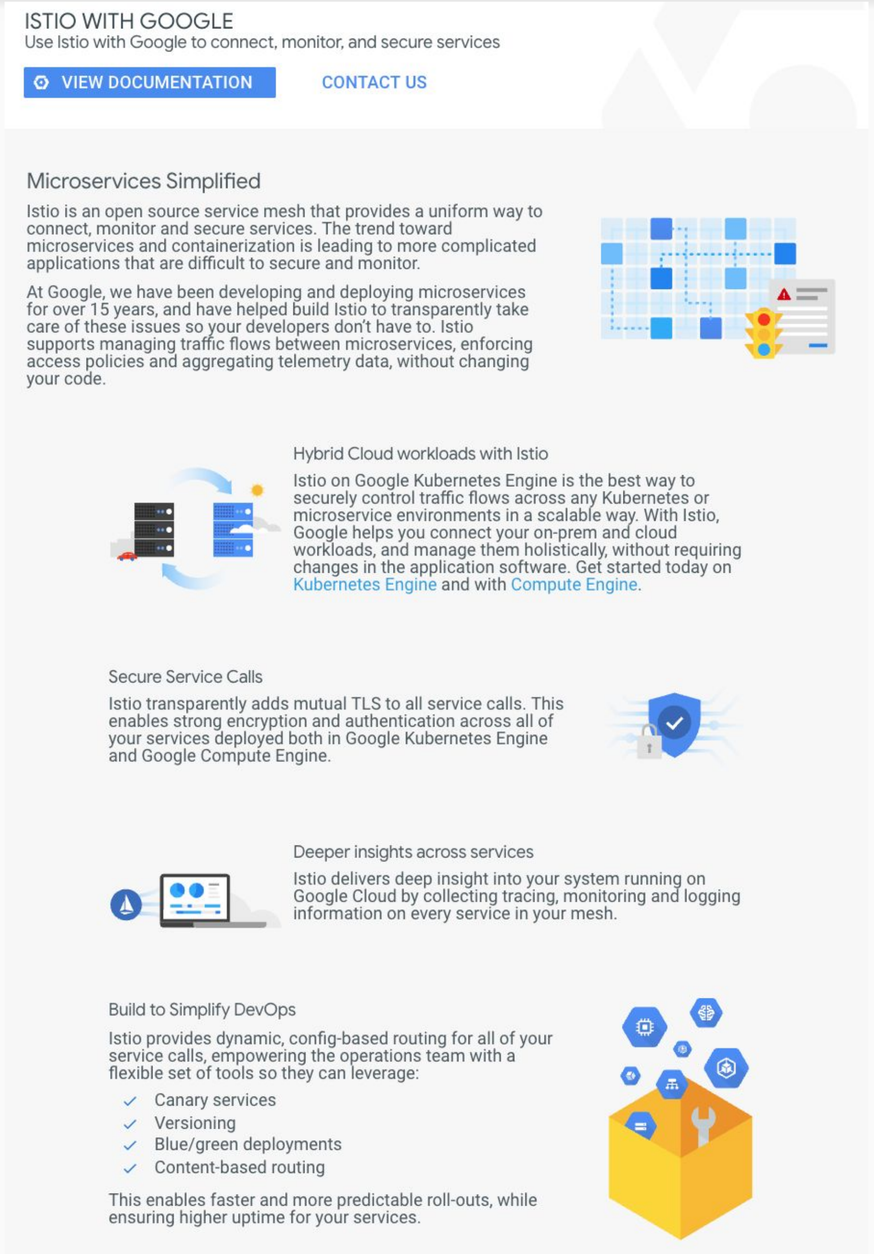
https://cloud.google.com/istio

Join the Istio user group

groups.google.com/forum/#!forum/istio-users

Participate or ask questions

istio.rocket.chat



How to engage with Istio Security

Developers

- Code contribution on github.com/istio/
- Sign up for Istio security groups.google.com/forum/#!forum/istio-security
- Join the Istio security community meetings (istio.zoom.us/j/117159906) bi-weekly on Wednesdays from 11am-12pm PST
- Sign up for Istio dev groups.google.com/forum/#!forum/istio-dev
- Join the wider Istio community meetings (zoom.us/j/986657835) bi-weekly on Thursdays from 11am-12pm PST

Questions?

Monitoring


Istio Dashboard
Last 5 minutes
Refresh every 5s
Source
All
Source Version
All
HTTP Destination
reviews.default.svc.cluster.local
TCP Destination
All
Destination Version
All
istio.io
Request Volume
0.45 ops
0.40 ops
0.35 ops
0.30 ops
0.25 ops
0.20 ops
0.15 ops
0.10 ops
0.05 ops
0 ops
17:09:00
17:09:30
17:10:00
17:10:30
17:11:00
17:11:30
17:12:00
17:12:30
17:13:00
17:13:30
All
200s
400s
All Secured Requests
All Unsecured Requests