ORACLE

Securing Cloud-Native Applications

Open research challenges and opportunities

François Gauthier
Senior Principal Researcher
Oracle Labs Australia

Cloud native practices empower organizations to develop, build, and deploy workloads in computing environments (public, private, hybrid cloud) to meet their organizational needs at scale in a programmatic and repeatable manner. It **is characterized by loosely coupled systems that interoperate in a manner that is secure, resilient, manageable, sustainable, and observable.**

CNCF Cloud Native Definition v1.1

Loosely Coupled and Secure – But Not Without Efforts!

The Shared Security Model

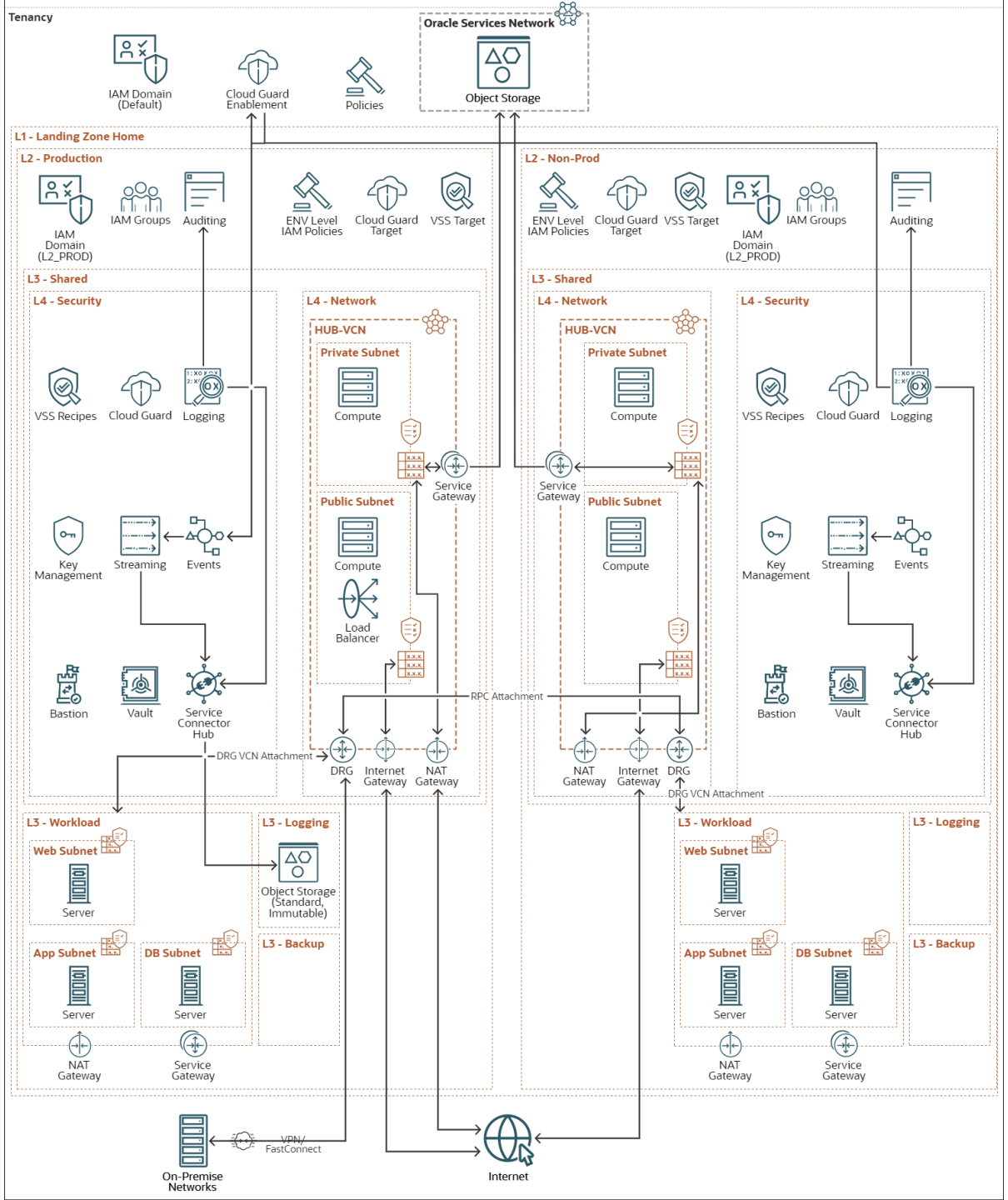
All major cloud vendors: Oracle Cloud Infrastructure (OCI), Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform (GCP) operate under a shared security model.

Details vary by vendor, but the split is typically along “security *of* the cloud” vs “security *in* the cloud”:

- Security *of* the cloud is owned by the cloud vendor.
  - Physical security of data centres
  - Hardware, software and networking that powers the vendor’s services
  - Services provided by the cloud vendor
- Security *in* the cloud is the customer’s responsibility.
  - User credential management
  - Identity and Access Management policies (IAM)
  - Network, firewall and security rules configuration
  - Data stored in the cloud

Security *In* the Cloud is Complex

A reference architecture from OCI

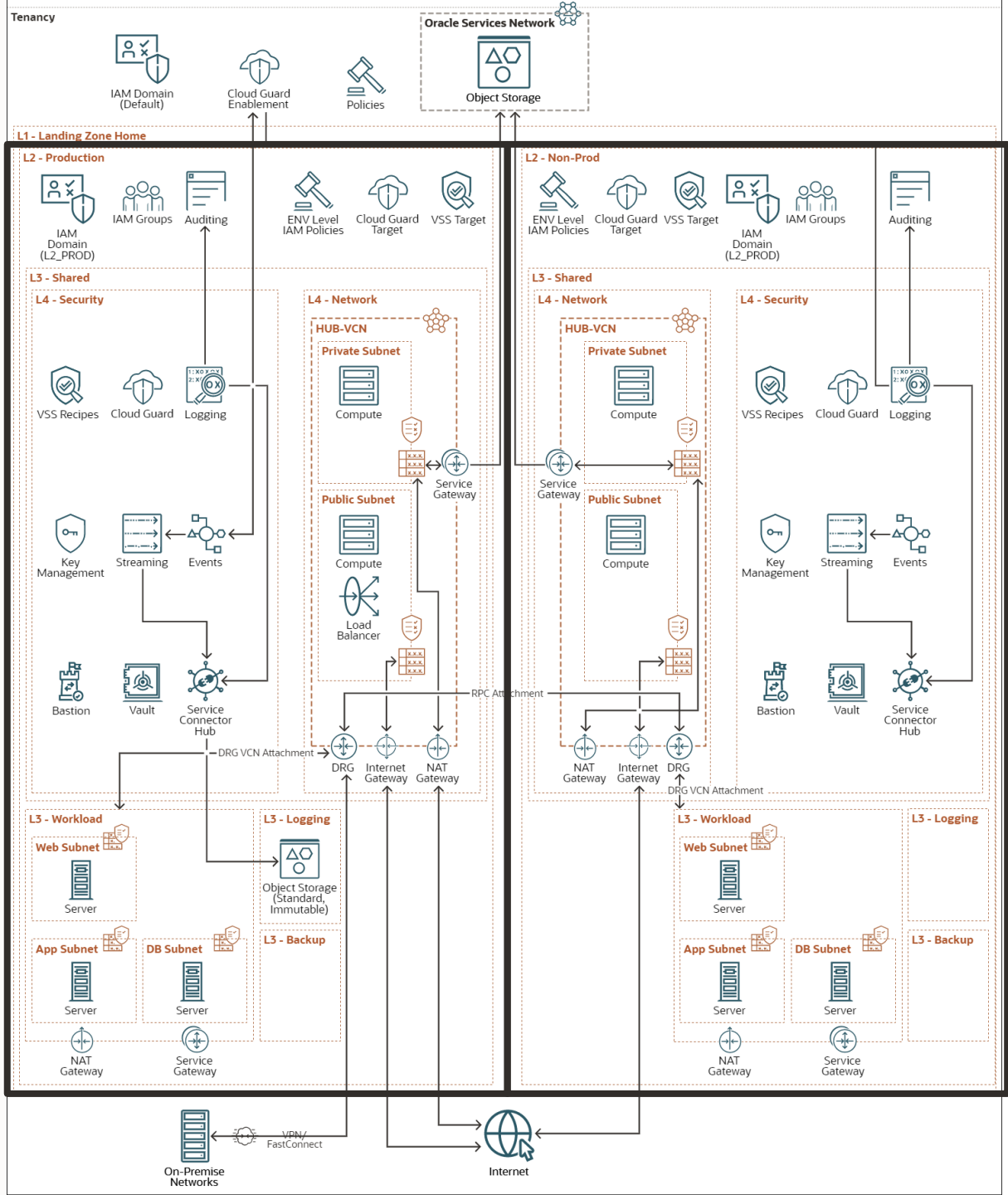


Security *In* the Cloud is Complex

A reference architecture from OCI

Security foundations:

- Development and production workloads segregation.

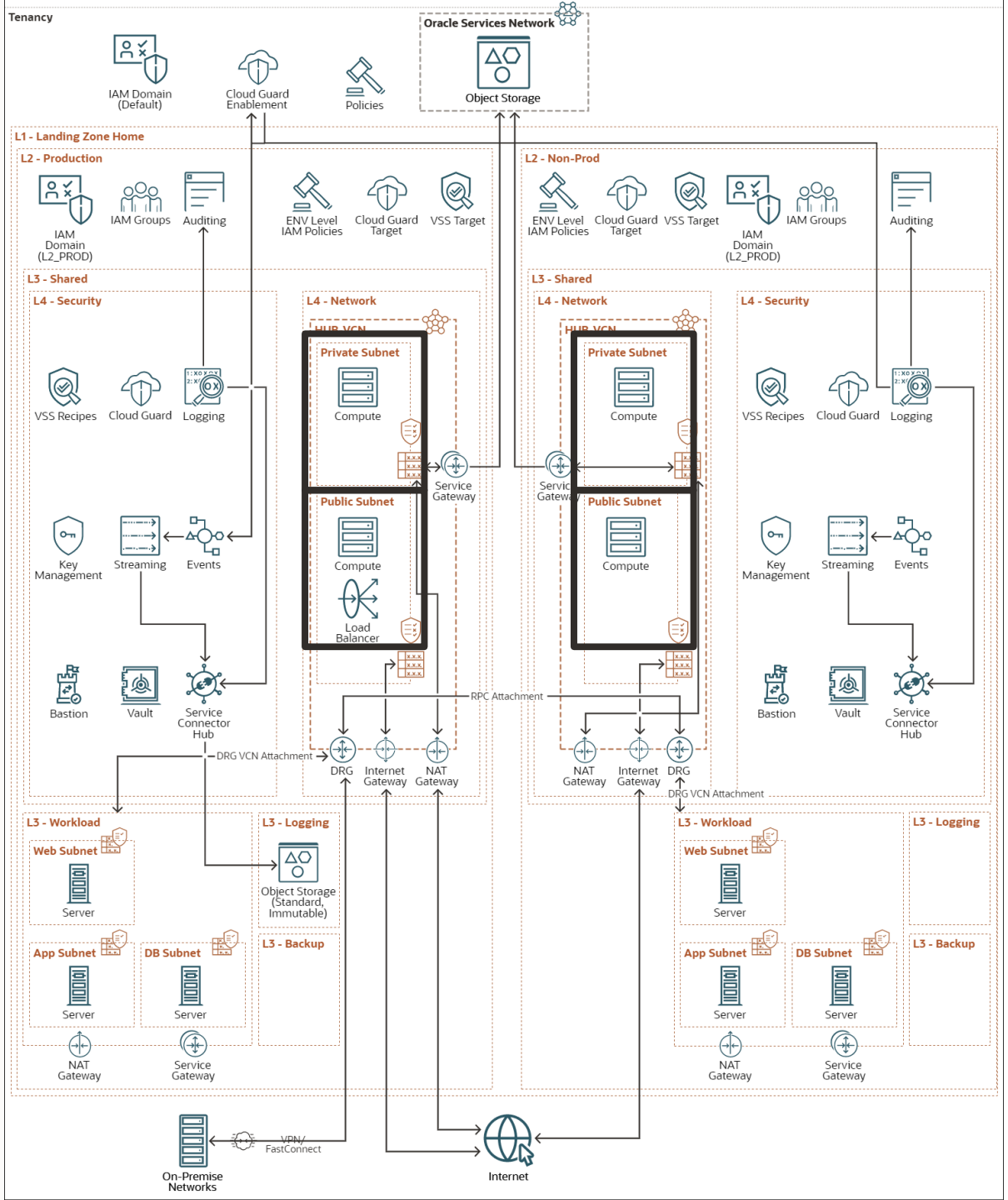


Security *In* the Cloud is Complex

A reference architecture from OCI

Security foundations:

- Development and production workloads segregation.
- Private and public network isolation

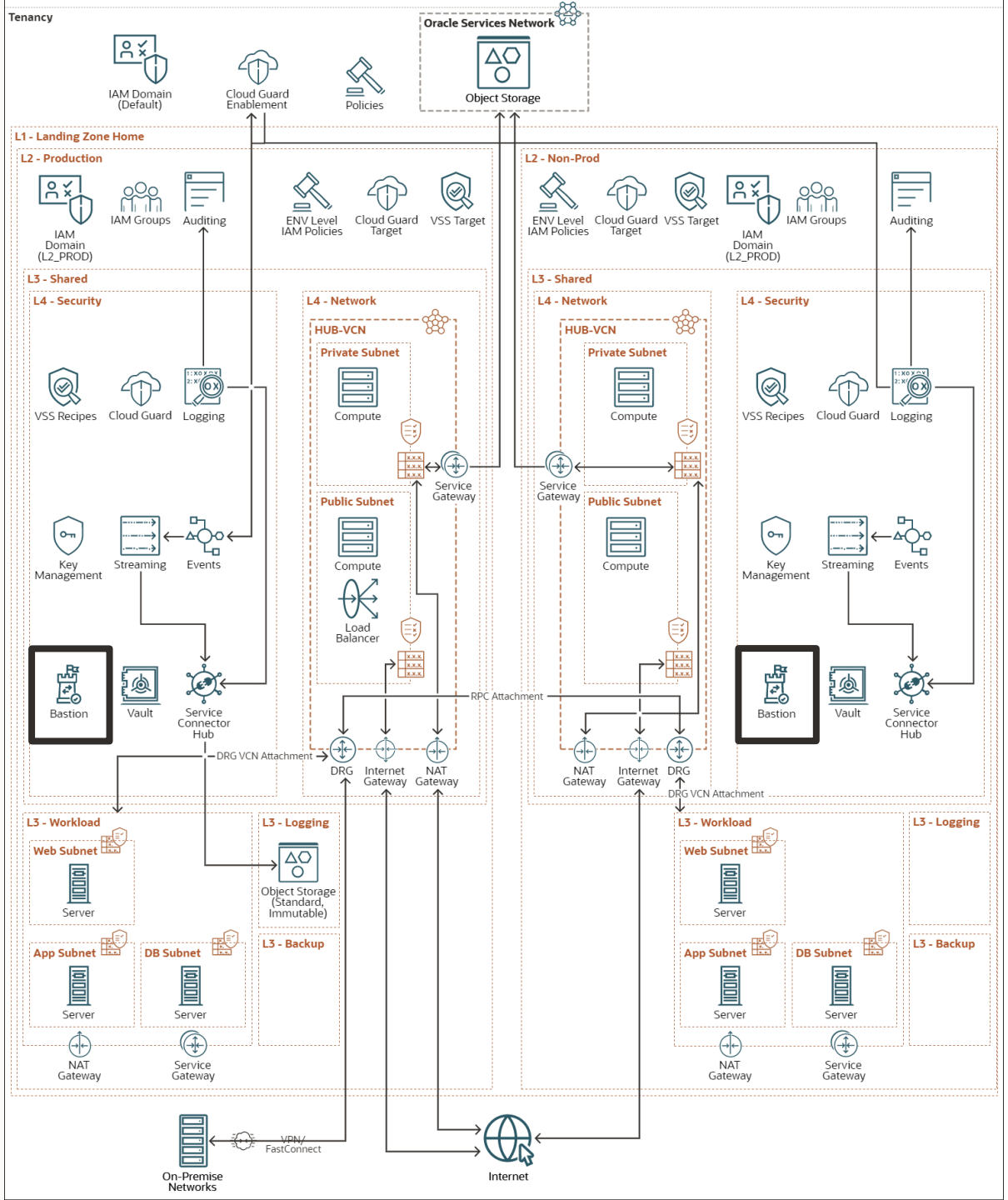


Security *In* the Cloud is Complex

A reference architecture from OCI

Security foundations:

- Development and production workloads segregation.
- Private and public network isolation
  - Secure access to private networks

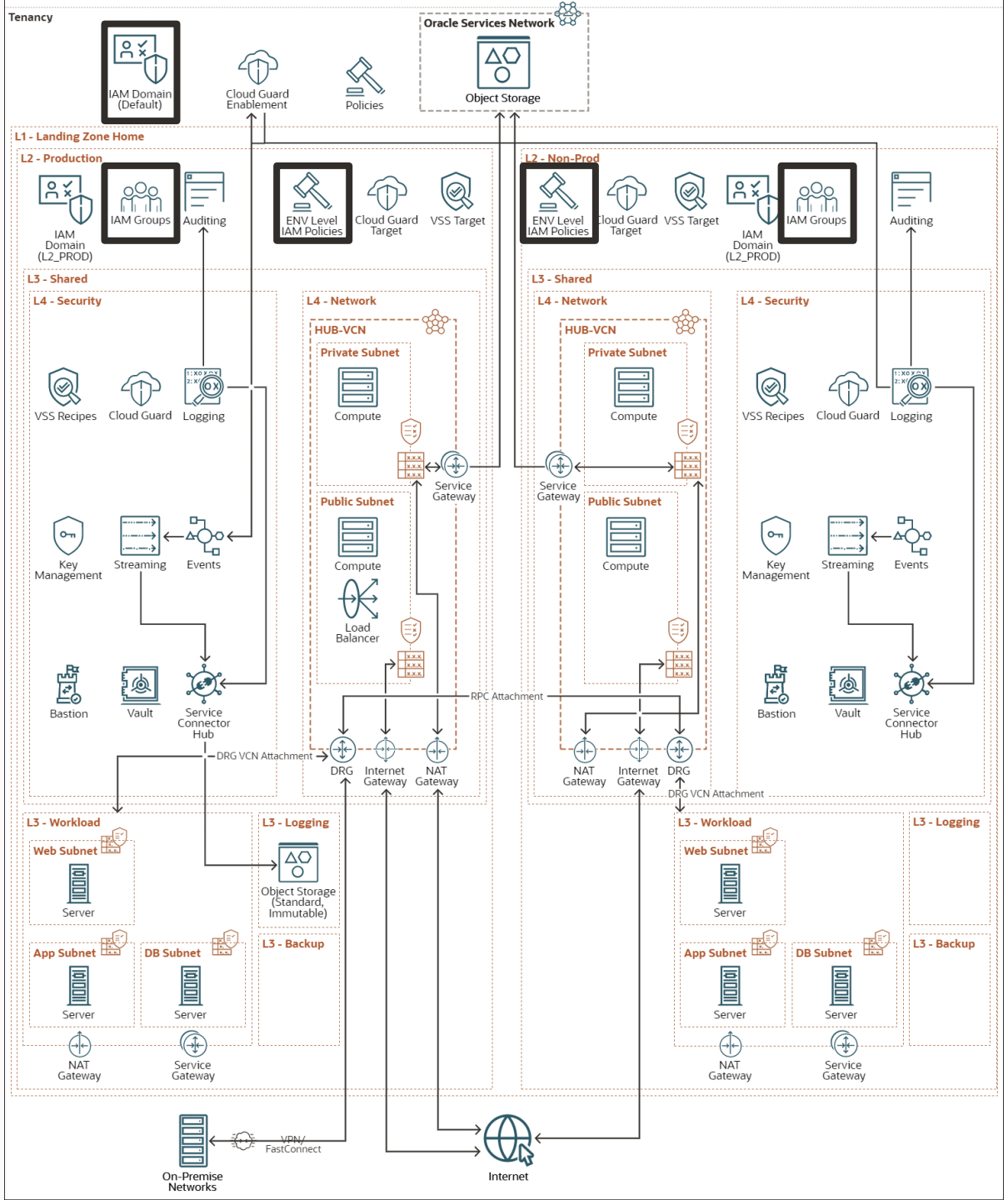


Security *In* the Cloud is Complex

A reference architecture from OCI

Security foundations:

- Development and production workloads segregation.
- Private and public network isolation
  - Secure access to private networks
- Identity and Access Management (IAM)

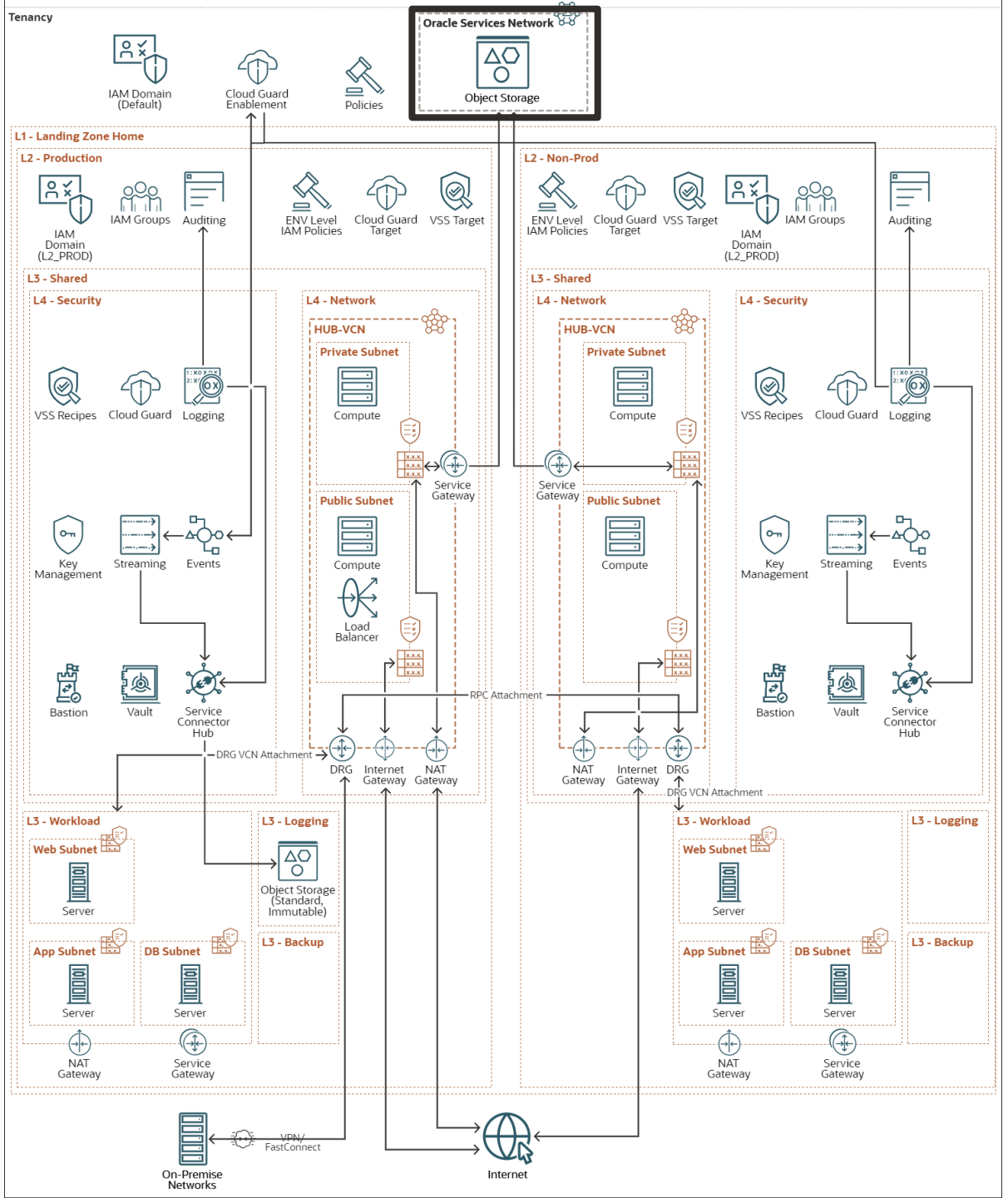


Security *In* the Cloud is Complex

A reference architecture from OCI

Security foundations:

- Development and production workloads segregation.
- Private and public network isolation
  - Secure access to private networks
- Identity and Access Management (IAM)
  - Data access management



Security *In* the Cloud is Complex

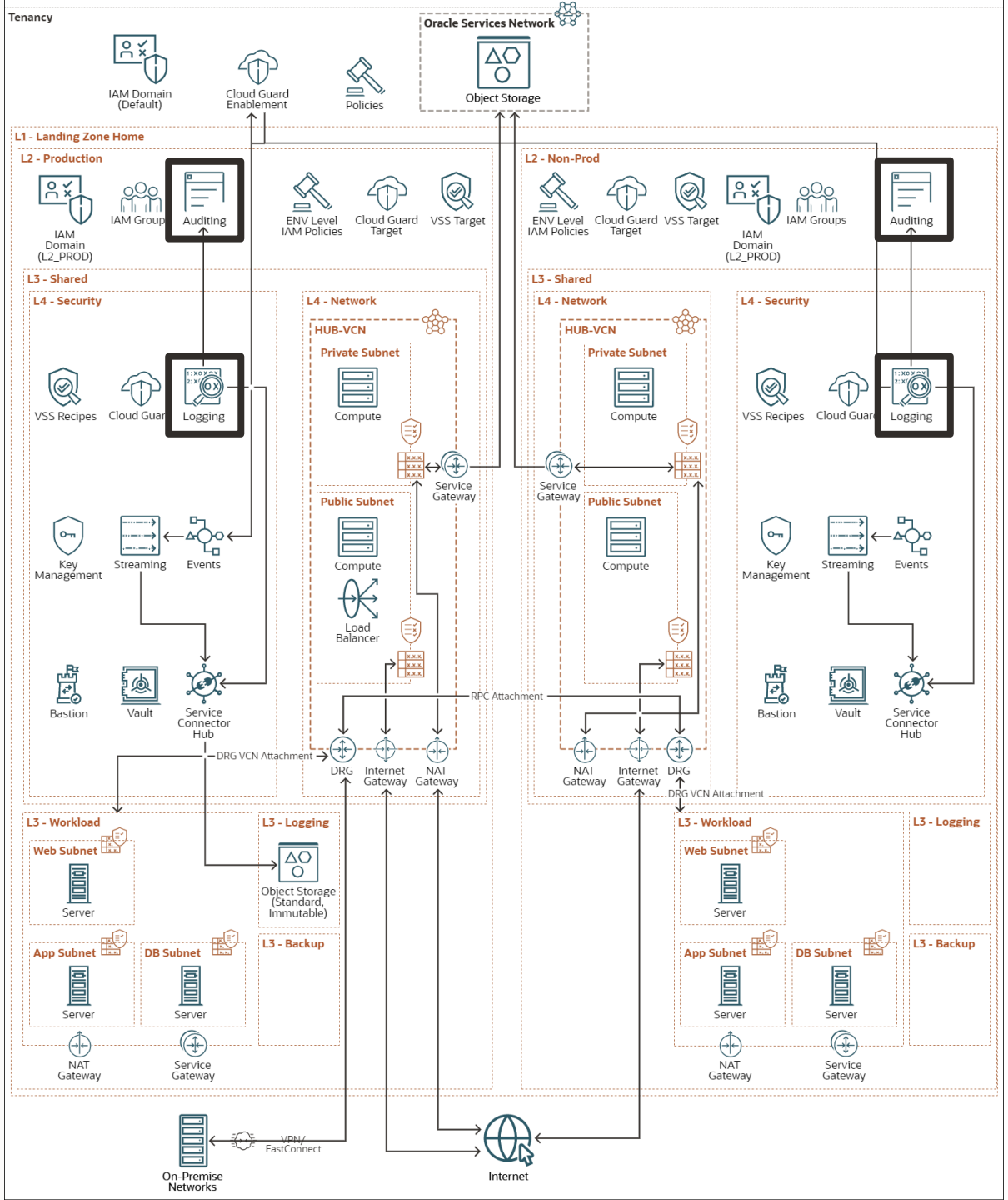
A reference architecture from OCI

Security foundations:

- Development and production workloads segregation.
- Private and public network isolation
  - Secure access to private networks
- Identity and Access Management (IAM)
  - Data access management

Security monitoring:

- Logging and auditing



Security *In* the Cloud is Complex

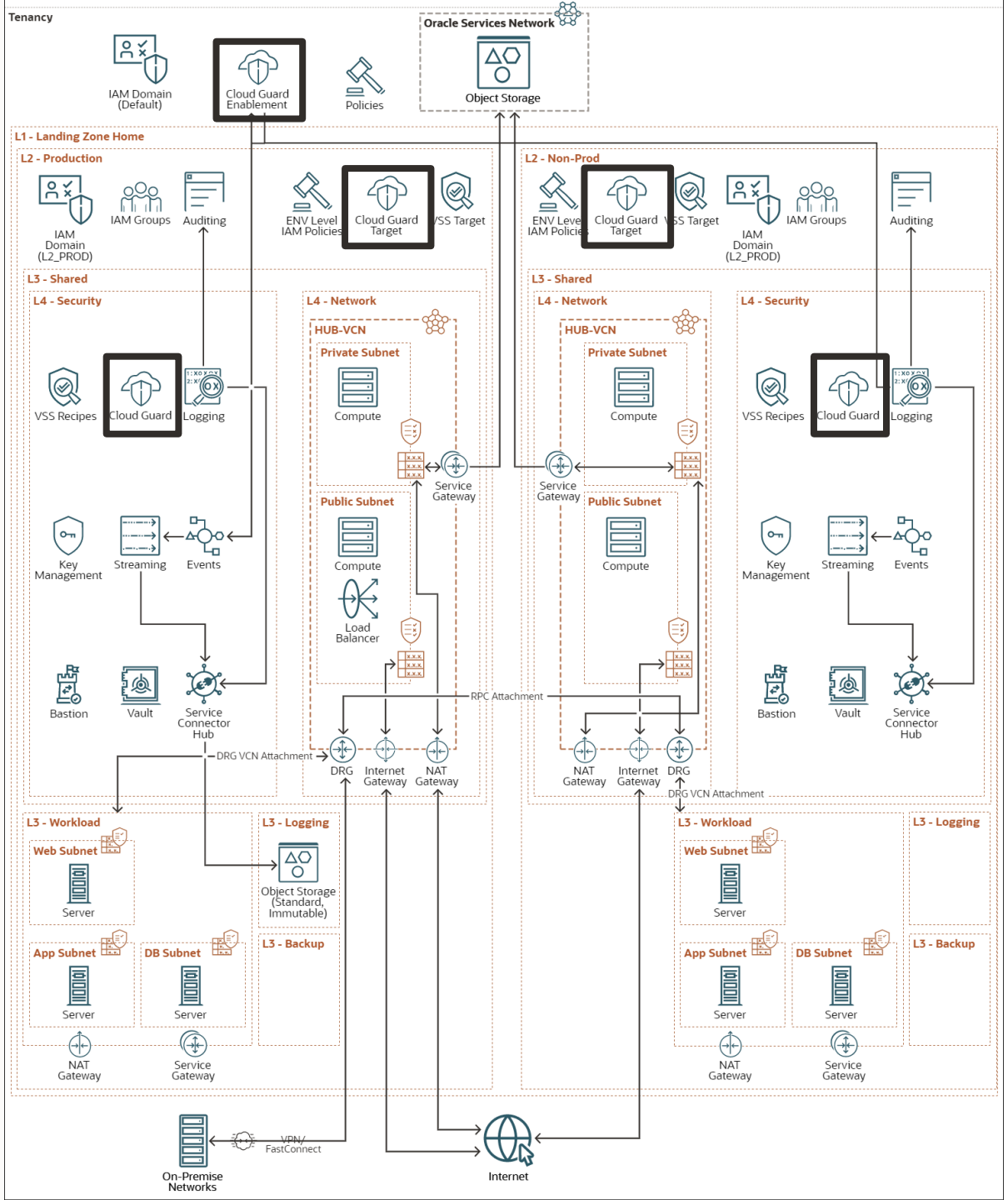
A reference architecture from OCI

Security foundations:

- Development and production workloads segregation.
- Private and public network isolation
  - Secure access to private networks
- Identity and Access Management (IAM)
  - Data access management

Security monitoring:

- Logging and auditing
- Runtime security posture management



Security *In* the Cloud is Complex

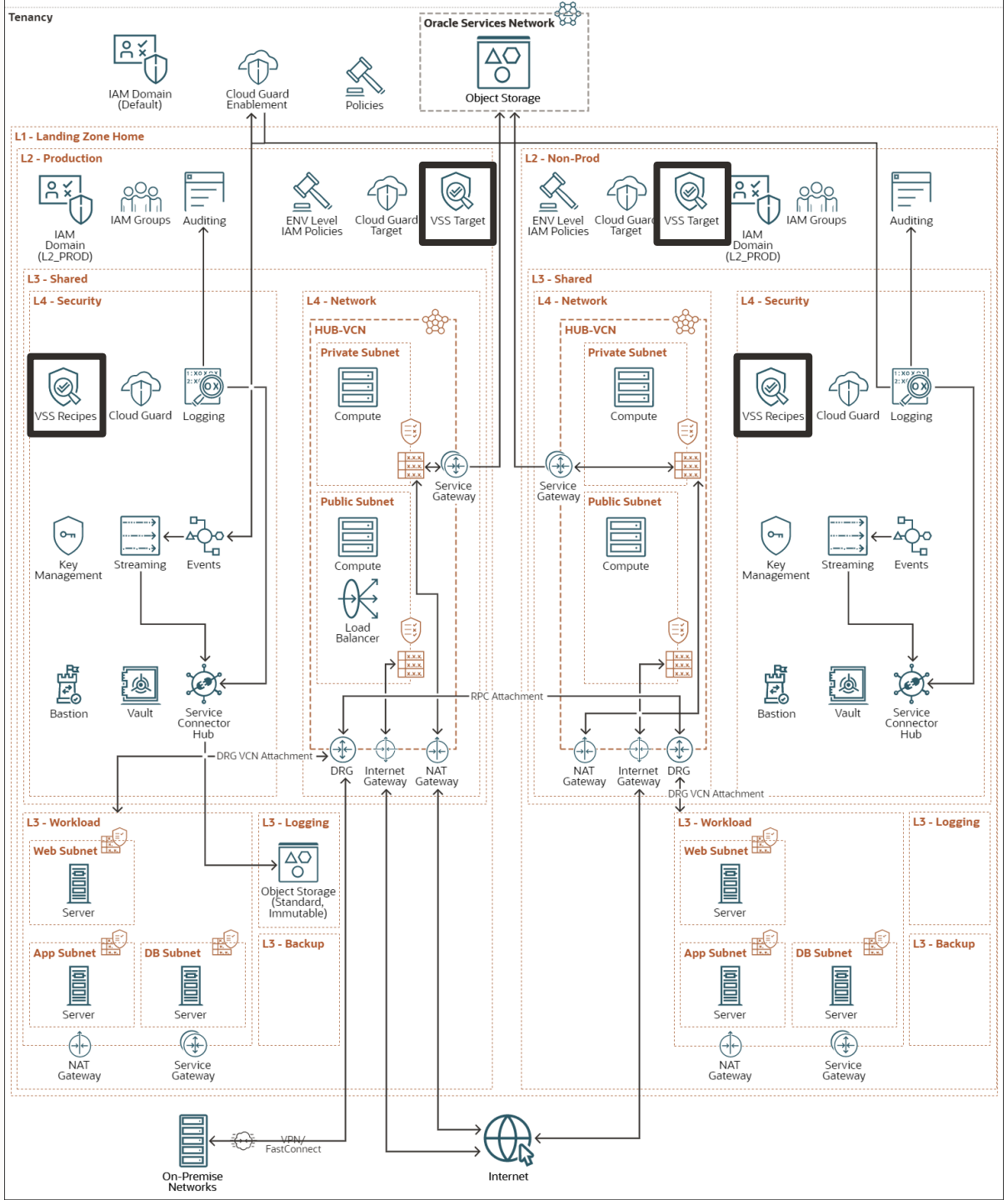
A reference architecture from OCI

Security foundations:

- Development and production workloads segregation.
- Private and public network isolation
  - Secure access to private networks
- Identity and Access Management (IAM)
  - Data access management

Security monitoring:

- Logging and auditing
- Runtime security posture management
- Vulnerability scanning (hosts and images)



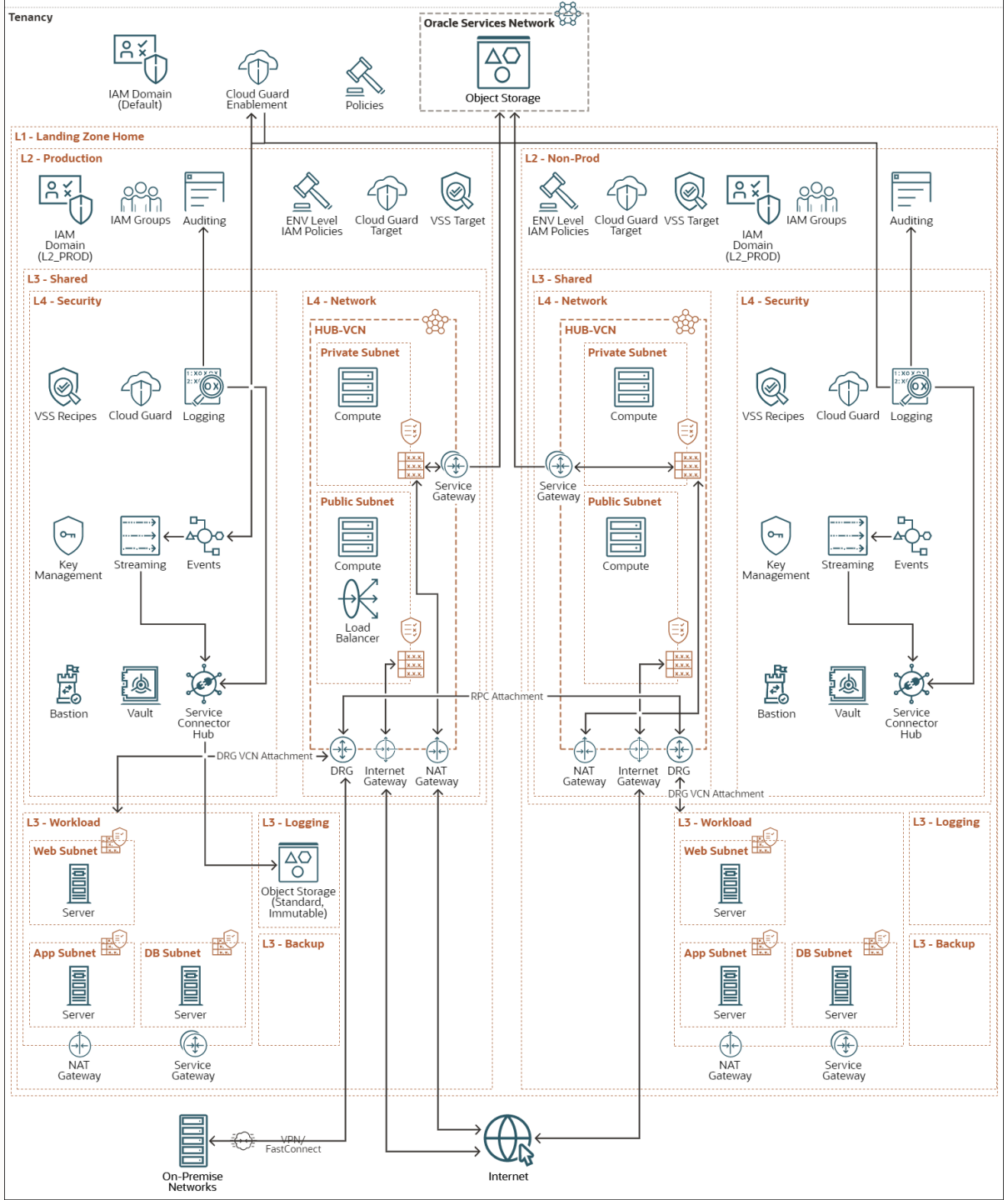
Focus of the Presentation Today

Security foundations:

- Development and production workloads segregation.
- Private and public network isolation
  - Secure access to private networks
- Identity and Access Management (IAM)
  - Data access management

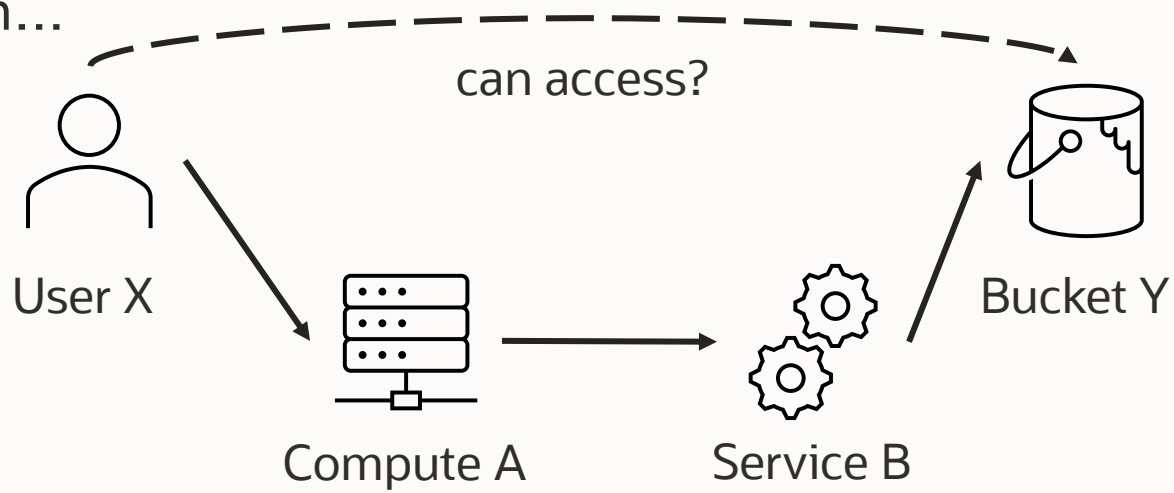
Security monitoring:

- Logging and auditing
- Runtime security posture management
- Vulnerability scanning (hosts and images)



Loosely Coupled Security Is Fragmented Security

And this is a problem…



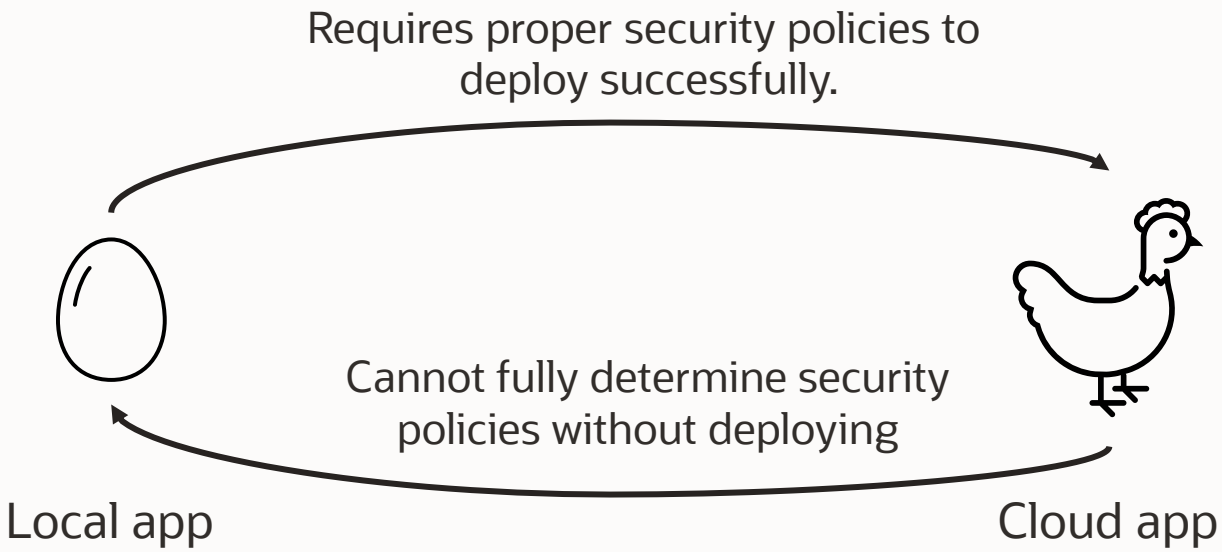
Answering a seemingly simple query like: “Can user X access bucket Y” can become very complex

- Are the resources in the same compartment? If not, are there cross-compartment policies in place?
- Does the network configuration have ingress/egress rules that enable the connection path X → A → B → Y
- Was user X granted access to bucket Y?
- Can service B query bucket Y on behalf of user X?
- Etc.

Checking complex properties of the whole system becomes a costly and piecemeal exercise.

Bootstrapping Cloud-Native Deployment

A chicken and egg problem



- Permission over-provisioning becomes the default solution to bootstrap cloud applications.
- Privilege tightening is often left as future work…

Problems We Have Identified so Far

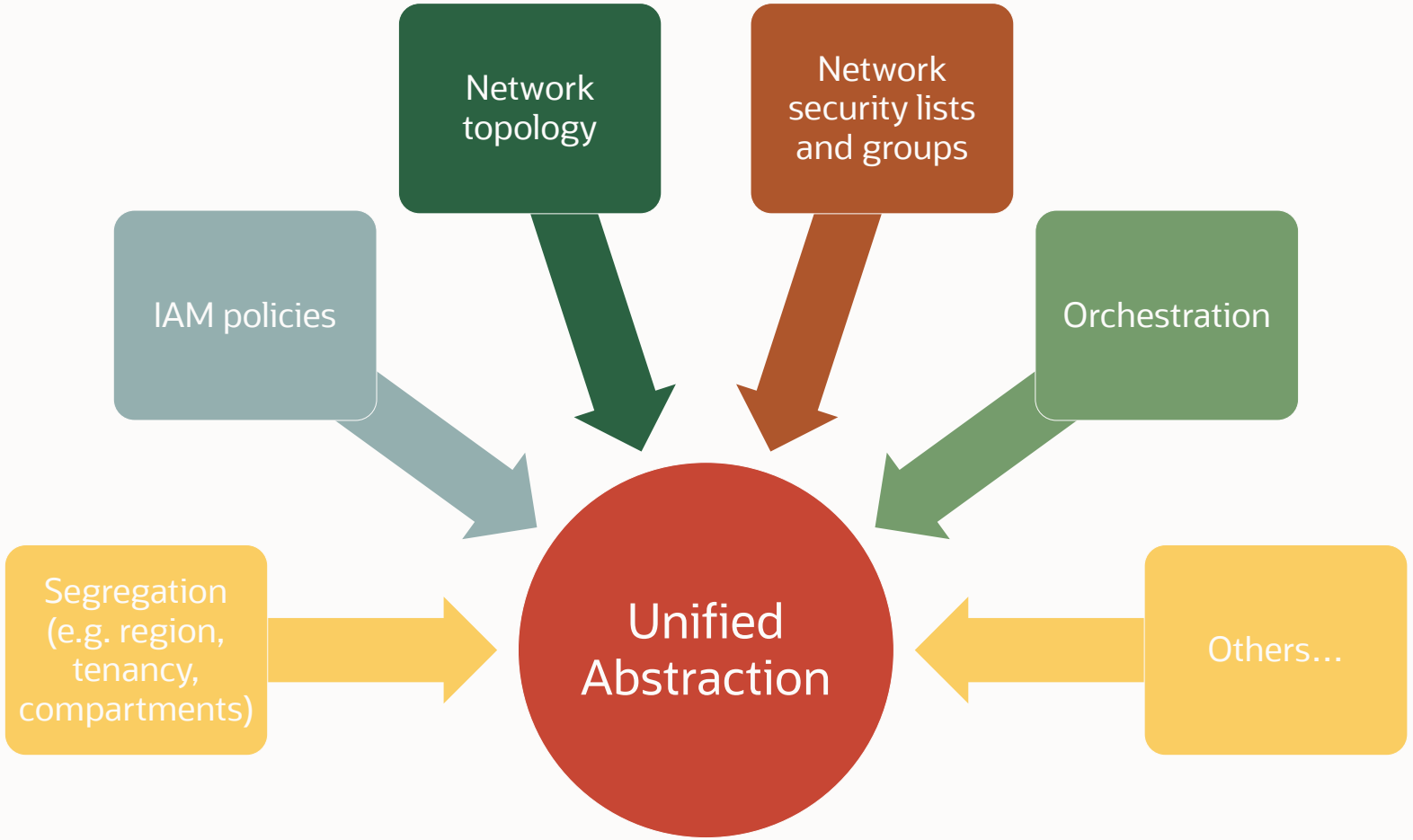
1. Security in the cloud is fragmented.

2. Checking complex security properties that span IAM, network, orchestration, etc. is difficult.

3. Bootstrapping cloud application deployment often require over-provisioning permissions and *hopefully* achieving least-privilege afterwards.

Unified Abstraction Needed

Addressing cloud security fragmentation



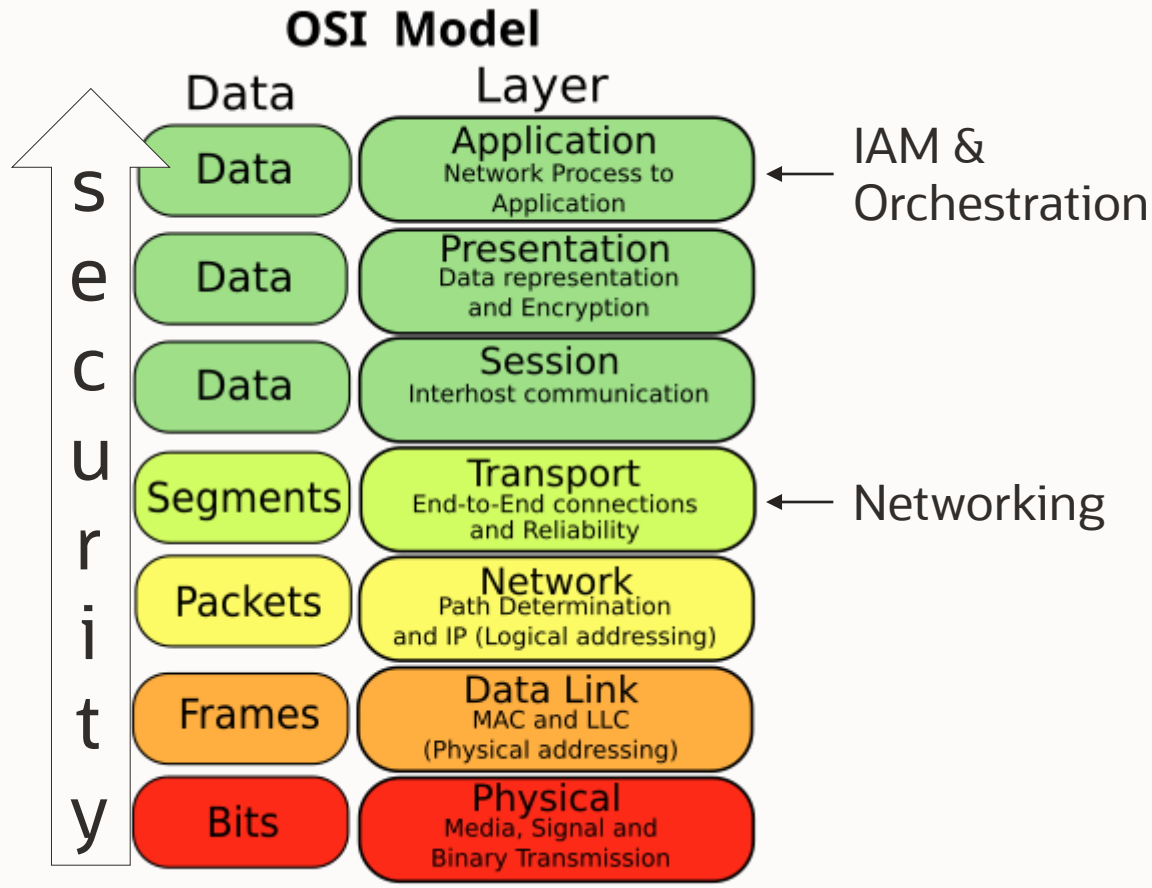
Factors to consider:

- IAM Policies can be:
  - Hierarchical
  - Stateful (e.g. session-based)
  - Positive (e.g. allow…)
  - Negative (e.g. deny …)
- Systems evolve over time
  - Network topology
  - Users
  - Features
- Security requirements evolve
  - Govt. standards
  - 0-days
- Cloud vendor's offering evolve

Querying, Checking and/or Verification Mechanisms Needed

Addressing the need to check complex security properties

- Several approaches to check security properties of cloud-native applications have been published.
  - However, they are typically limited to one dimension (i.e. networking or IAM).
- Security cuts across the stack.
  - Reasoning about network and IAM separately serves researchers, not security.
- We need new theories and tools to check and verify the security of cloud-native systems.



Policy Inference Mechanisms Needed

Example of a typical cloud-native application

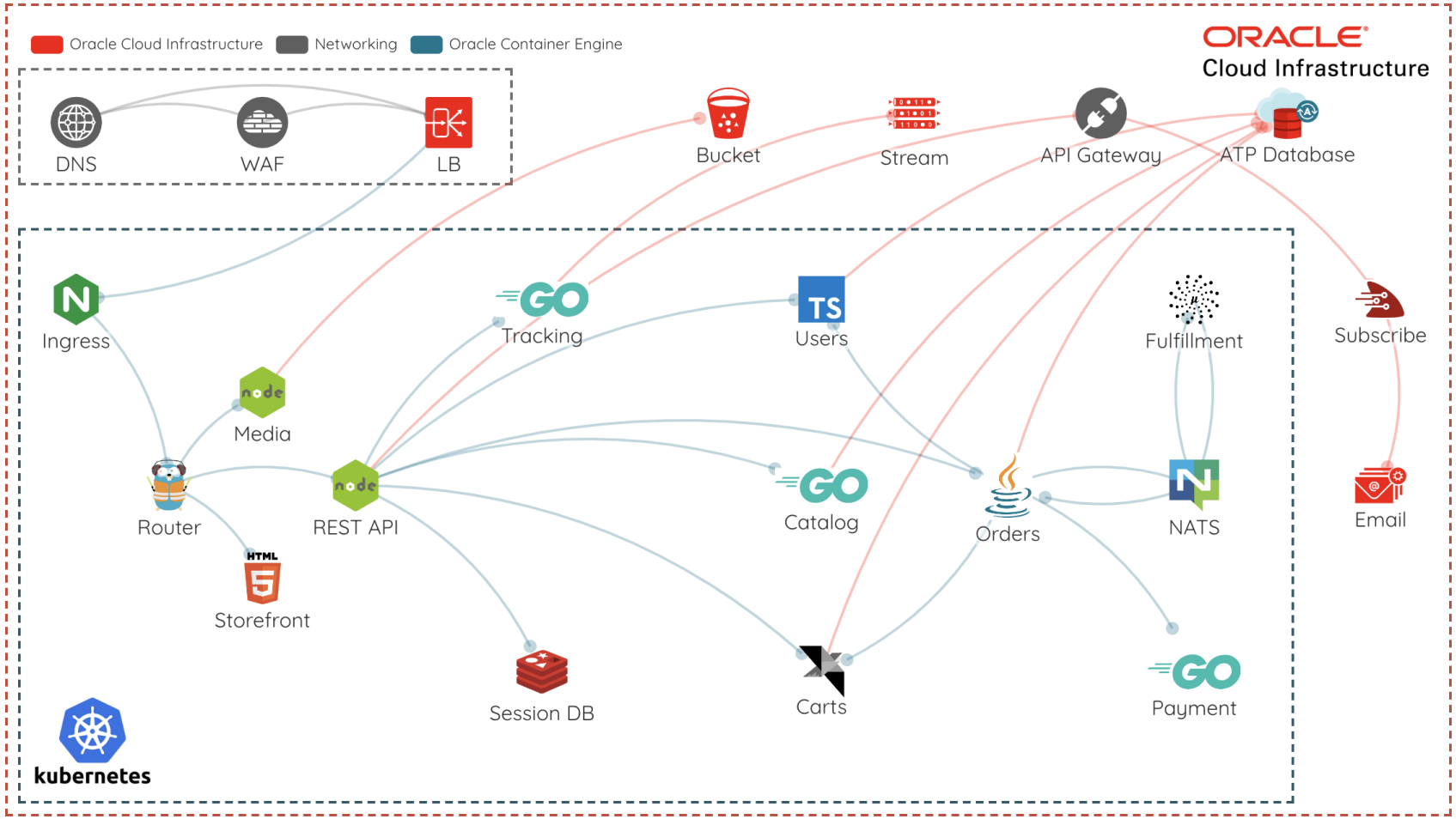
In the figure:

- Red links are governed by networking and IAM policies.
- Blue links are governed by networking and Kubernetes policies.
- Grey links are governed by networking and IAM policies.

Policies can only be tested after the application is deployed.

Debugging policy failures can be tedious.

Over-provisioning is an easy fix.

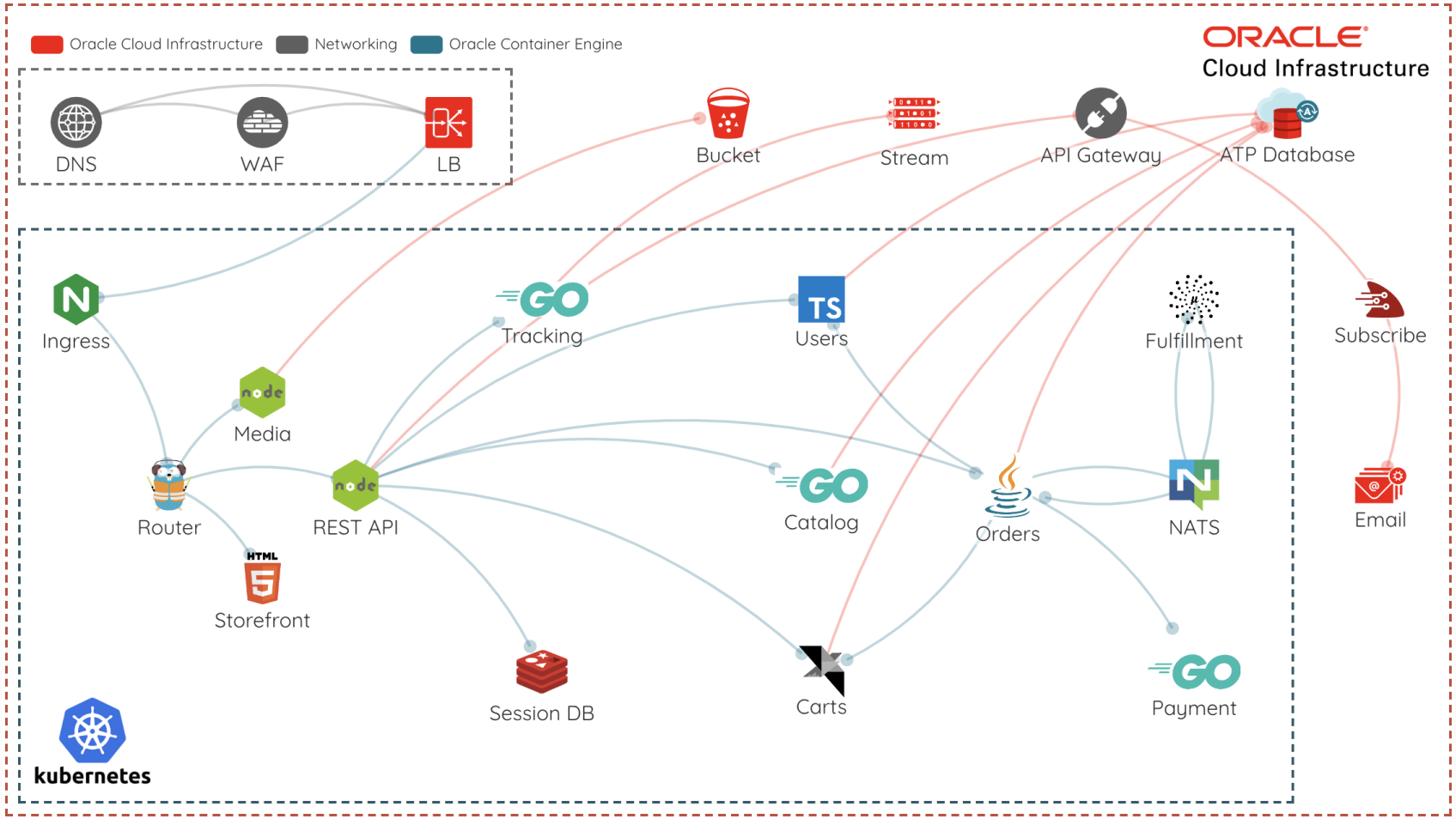


Policy Inference Mechanisms Needed

Addressing bootstrapping and least privilege issues

Potential solutions to infer least privilege policies:

- Emulate the cloud environment locally (e.g. LocalStack) design a policy by hand and test.
- Build a model of the network and application architecture and infer an ideally provably least privilege policy.
- Deploy in an over-provisioned development environment and infer a least privilege policy based on dynamic observations.



Conclusion

Challenges we are facing now:

- Lack of a unified abstraction
- Difficulty checking cross-domain security properties
- Inferring least privilege policies for cloud-native applications.

Once this is addressed, constraining information flows will likely become a priority.