Attestation for Hosted Workloads

Liam Farrelly, Product Engineer @ Evervault

Agenda

—Context

—Trust Model

—Walkthrough: Using Attestation to Authorize Enclaves

—Q&A

Context

About Evervault

—We build infrastructure products to solve complex security and compliance challenges.

—All of our products use cryptography to solve these problems.

—Became design partners for AWS Nitro Enclaves in 2020

—Used it to protect user private keys with our Encryption Engine

Experience building on Enclaves

—Nitro Enclaves allowed us to build a secure, attestable service as a small team.

—Involved some heavy lifting to get to production:

- Tooling
- Observability
- Scaling

—Overall a success

Packaging up what we learned

- —Enclaves builds on our experience from running AWS Nitro Enclaves in production
- —Aims to reduce the initial engineering effort:
  - Easier build process
  - Features to support lift & shift of existing containers
  - Clients with attestation built in
  - Managed deployments

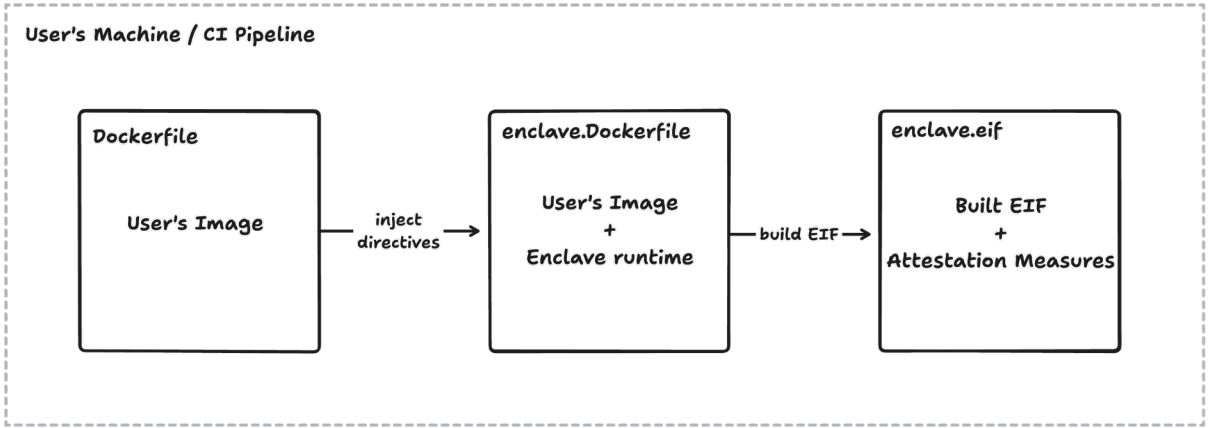
Easier Build Process

—Our CLI lets users build their Enclave as though it is a standard Docker Image

—Best effort reproducible builds

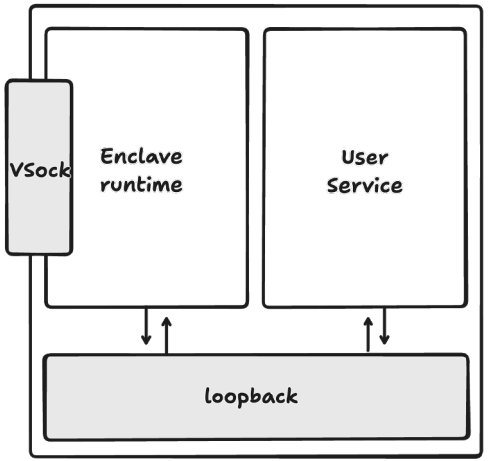
—Support for pinning signing certificates

—Support for selectively including features in the runtime



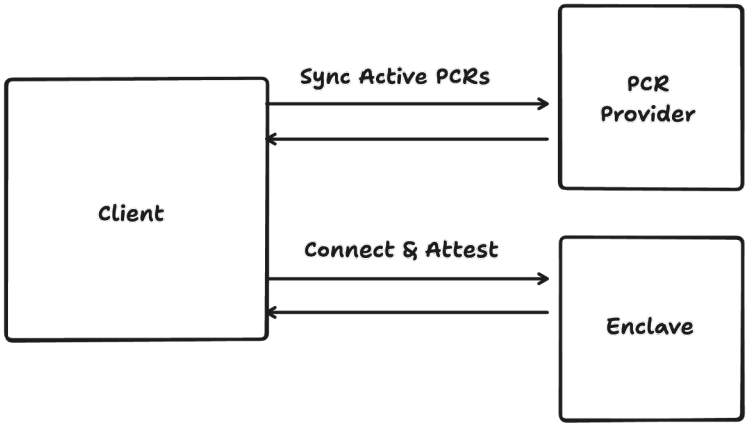
Supporting existing containers

- —Provide configuration in keeping with traditional, on-demand compute platforms
- —In-Enclave runtime abstracts away the Enclave environment
  - Manages the Enclave<>Host bridge
  - Handles TLS certificate provisioning & termination
  - Exposes attestation documents
  - Selective support for network egress



Clients with attestation built in

- —Maintain a Rust library to validate Attestation documents
- —Support attestation from 5 Client SDKs
- —Adding support for syncing Attestation Measures from Client SDKs
    - Avoids complex orchestration issues when Attestation is user facing



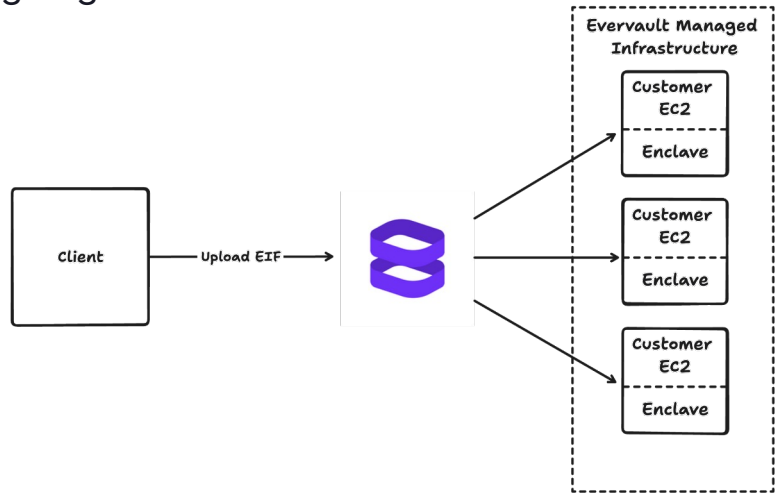
Managed deployments of TEEs

—Users build their own image, and provide to us for deployment

—We manage the instance provisioning, configuration, and routing

—We handle scaling, and apply security patches

—We provide notifications for expiring Enclave signing certificates



Trust Model

The Trust Model for managed deployments

—Managed deployments introduce an interesting trust model…

—From our customers’ perspective, they want to only trust their Enclave

—From our CloudSec perspective, we only want to trust our Control Plane

The Trust Model: Customer Perspective

—Customers build & sign their Enclave image on their own machine

—Built Enclave image includes the Evervault Runtime

—Our CLI [1] and Runtime [2] are open source

—The customer uploads their built & signed image to Evervault

—We handle Enclave orchestration, *but not builds*

[1] https://github.com/evervault/enclave-cli

[2] https://github.com/evervault/enclaves

The Trust Model: Evervault Perspective

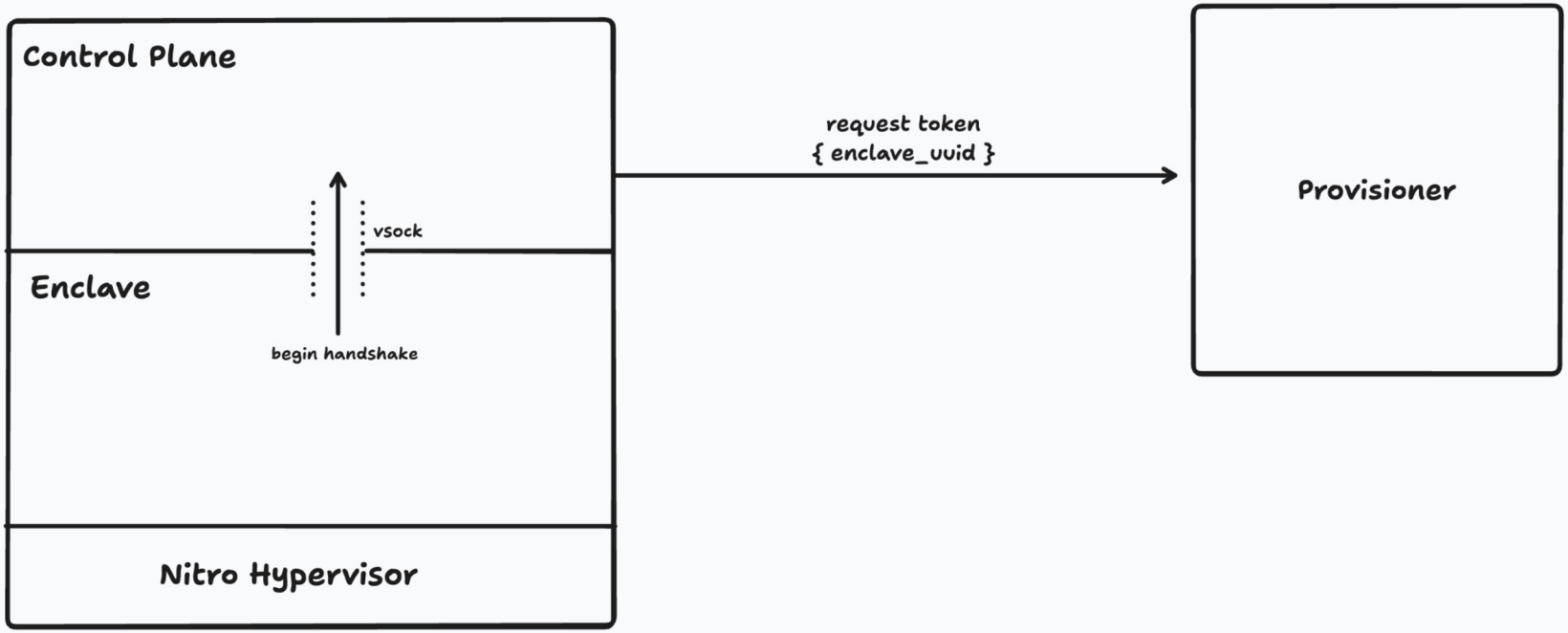
—We wrap the customer image in a Control Plane before deploying

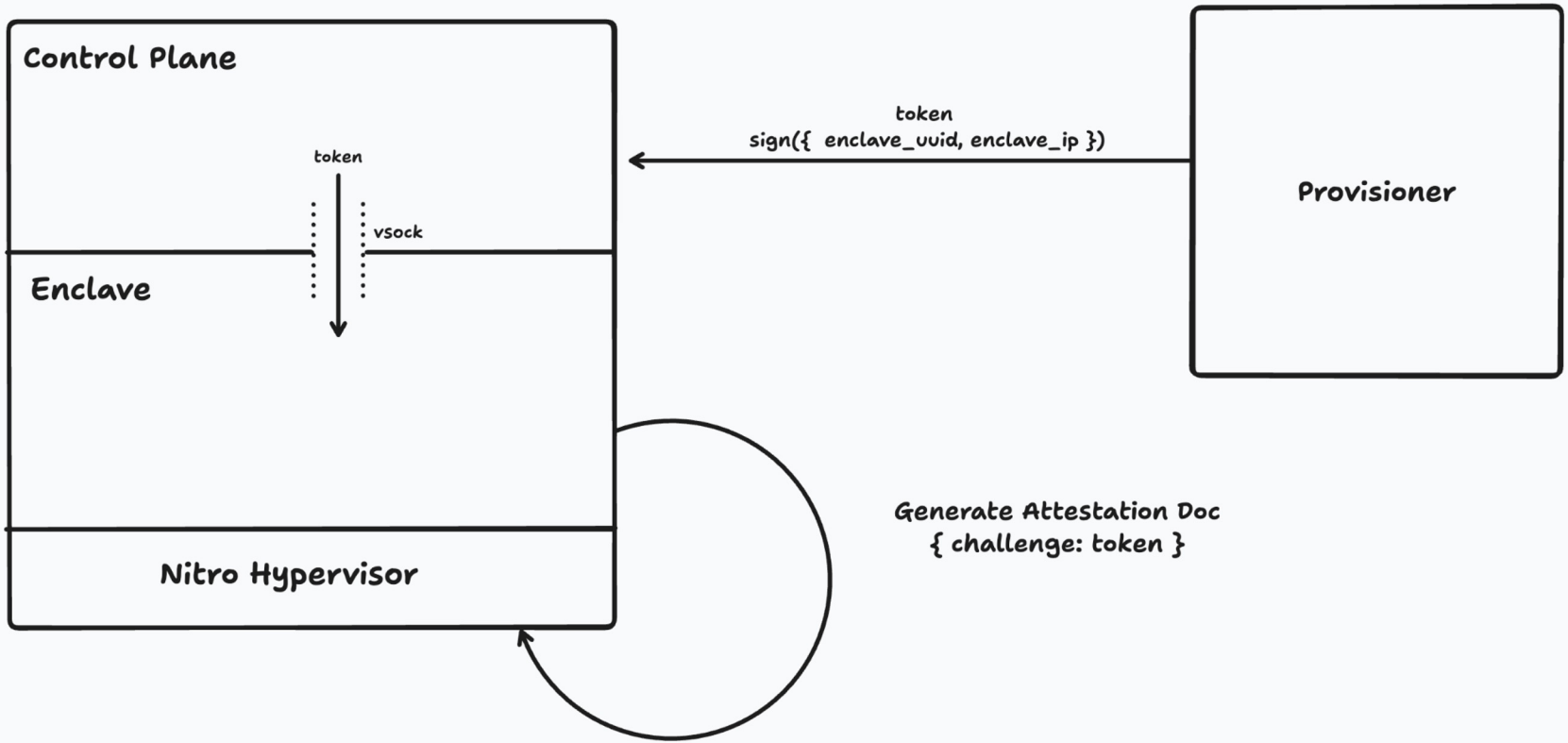
—Our Control Plane is launched with the Enclave’s ID, and internal credentials

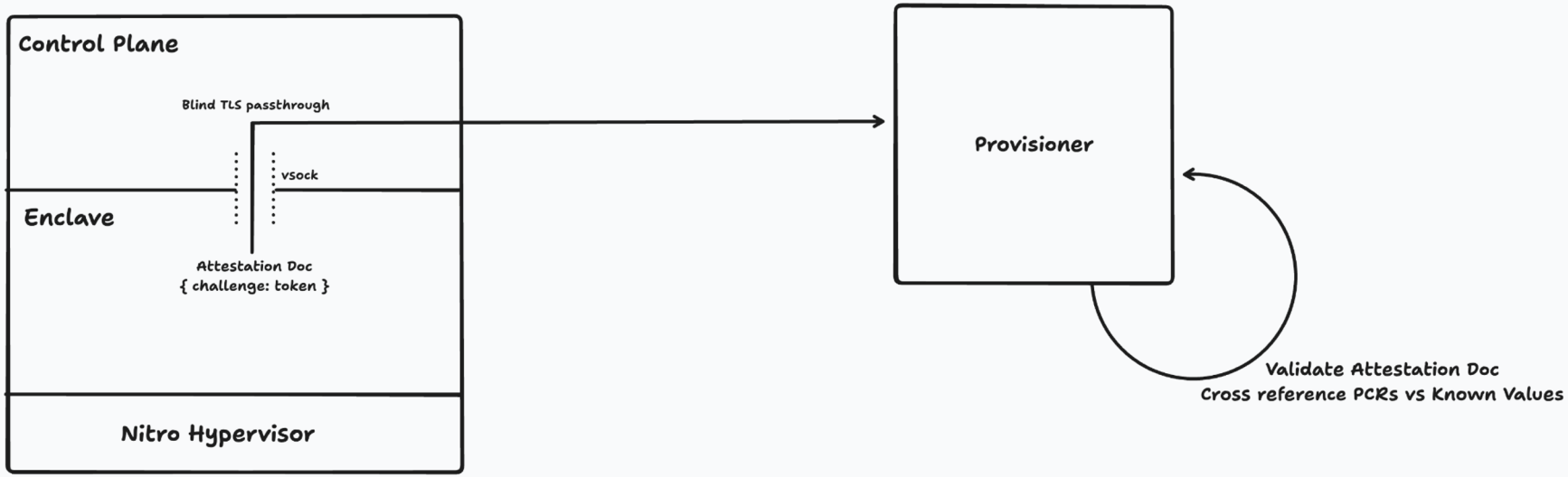
—Cannot know if the Enclave is being deployed with a valid Runtime

—Need to treat the in Enclave processes as untrusted entities

Using Attestation to Authorize Enclaves


Control Plane
vsock
Enclave
begin handshake
Nitro Hypervisor
request token
{ enclave_uuid }
Provisioner


Control Plane
token
vsock
Enclave
Nitro Hypervisor
token
sign({ enclave_uuid, enclave_ip })
Provisioner
Generate Attestation Doc
{ challenge: token }


Control Plane
Blind TLS passthrough
vsock
Enclave
Attestation Doc
{ challenge: token }
Nitro Hypervisor
Provisioner
Validate Attestation Doc
Cross reference PCRs vs Known Values

Assertions from the Attestation Document

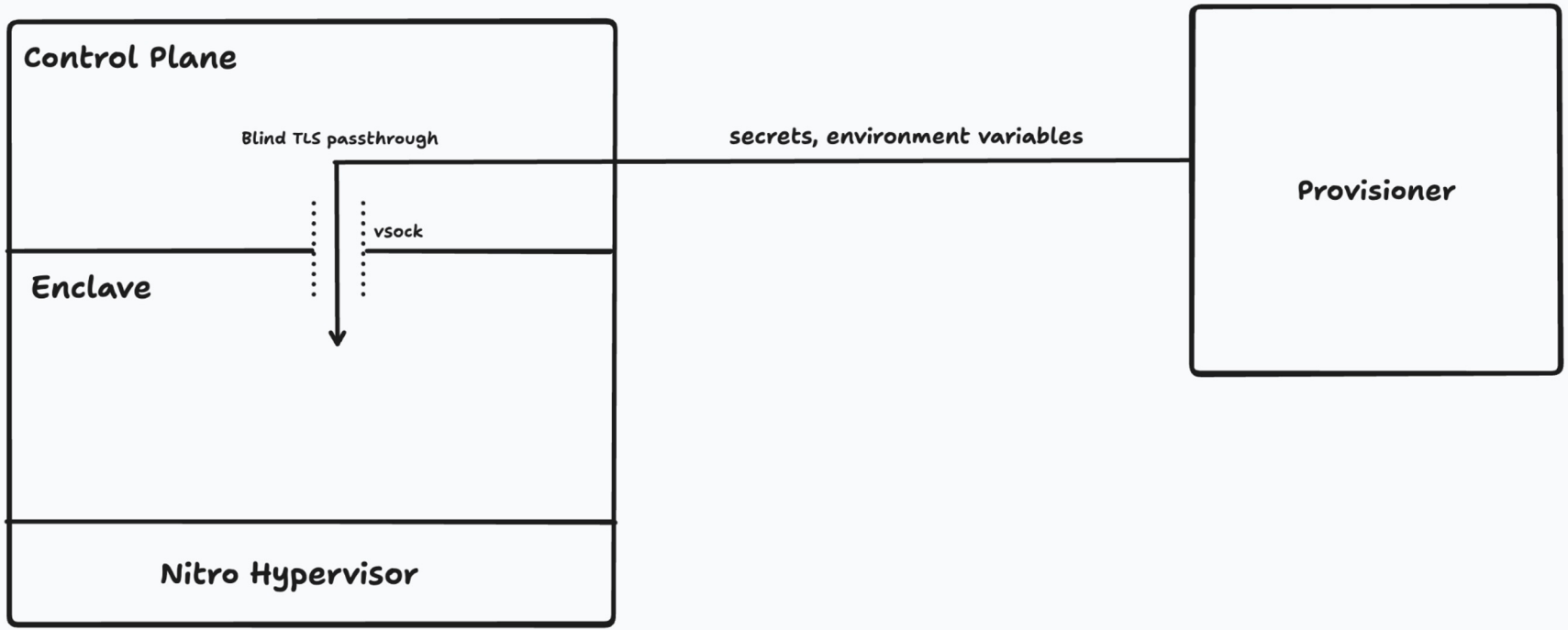
—Using the Attestation Document, the Provisioner can verify the integrity of the Enclave.

—The Attestation Document itself allows us to verify the connection is from an Enclave.

—The challenge allows us to reliably identify the Enclave.

—We can then cross reference the PCRs in the Attestation Document and the IP of the request against the known Enclave information.

—Once all of these tests pass, we can trust the Enclave.


Control Plane
Blind TLS passthrough
secrets, environment variables
Provisioner
vsock
Enclave
Nitro Hypervisor

Takeaways

—This model of authorization improves upon standard authorization, typically based on instance identity.

—By centering our Enclave authorization around attestation, we only issue secrets to instances based on the integrity of the deployed image.

Thank you