CONFIDENTIAL COMPUTING SUMMIT 2026

Hosted by

THE LINUX FOUNDATION + OPAQUE

CONFIDENTIAL COMPUTING SUMMIT 2026

Hosted by THE LINUX FOUNDATION + OPAQUE

Agentic Zero Trust:
At Rest, in Transit, and at Runtime

Who are we?

Nina Polshakova

Software Engineer

- Maintainer of agentgateway, AAIF project and Kubernetes Gateway API implementation
- CNCF Ambassador and former Kubernetes v1.33 Release Lead
- Co-Lead CNCF AI Agentic Standards Whitepaper
- Active member of the CNCF Artificial Intelligence Technical Community Group

Josh Halley

Principal Architect

- Key focus on AI in Cloud Native, Security and Sovereign Critical Infrastructure
- Author of Zero Trust in Resilient Cloud and Network Architectures and Autonomous Networks: Using AI to Advance Operations in SP and Enterprise Domains
- Co-organizer from the CNCF Artificial Intelligence Technical Community Group

Foundations

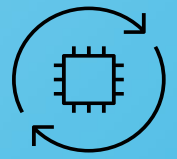
Securing Data in IT Systems Today

Data in Transit

Data traveling from location to location, secured via encryption mechanisms (IPSEC, MACSEC, Etc..)

Data at Rest

Data which resides in storage of in a distributed or centralized file store. Data can be archived with different levels of encryption.



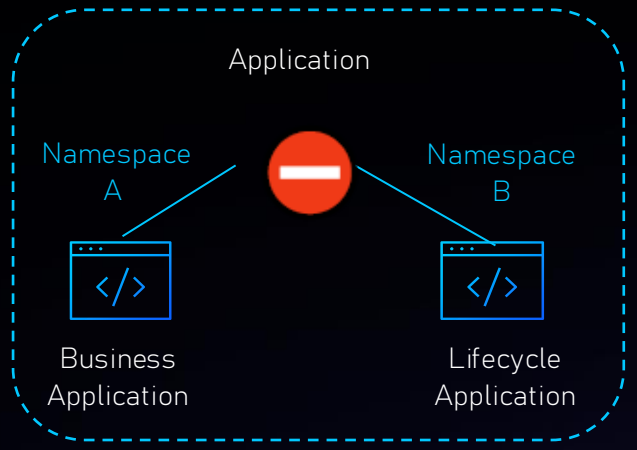
Data at Runtime

Data which is currently in use in the CPU, Memory and GPU of a system. This could include confidential training data (weights).

Zero Trust - Segmentation Recap (Application)

Macro-Application Segmentation (Environment and Namespace)

Application Environment and Tenant

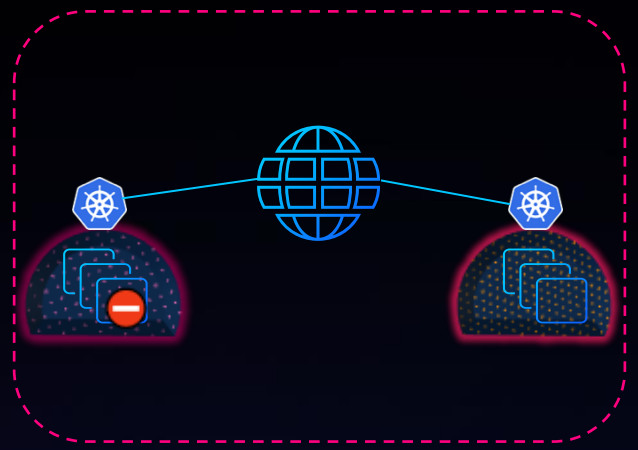


- Logical namespace separation (dev, stage, prod)
- Business and Lifecycle app boundaries
- Prevent x-environment lateral movement

Caveat: Management overhead (policy) across environments

Micro-segmentation (Service-to-Service)

Identity based communication between services



- Method: Service Mesh, API Gateways, mTLS
- Enforcement: Service Identity (SPIFFE/SPIRE)
- Requires explicit permission between services

Caveat: Requires robust PKI and identity management

Nano-Segmentation (Function / Code)

Application policy enforcement (RASP, Policy as Code, eBPF)

```
SEC("lsm/socket_create")
int BPF_PROG(socket_policy, int family, int
type, int protocol, int kern, int ret)
{
    if (ret)
        return ret;

    if (family == AF_INET && type == SOCK_DGRAM)
        return -EPERM;    // blocked ❌

    return 0;             // permitted ✅
}
```

- System Call, Database Query, Data Object
- Limit fallout of a compromise
- Deep integration at application level

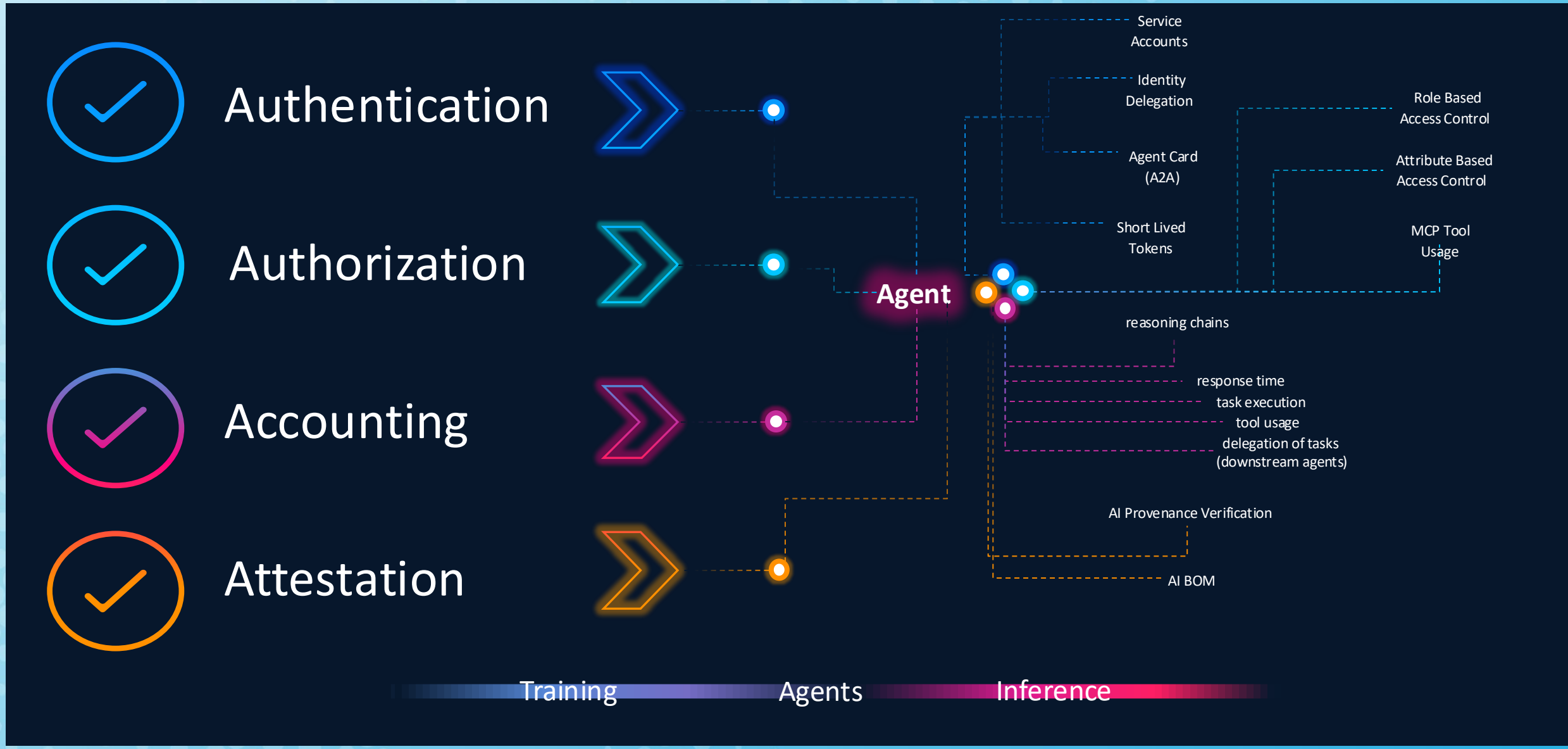
Caveat: Can impact runtime performance

Wide ≥ Environment/NS

Narrow ≥ Service/API

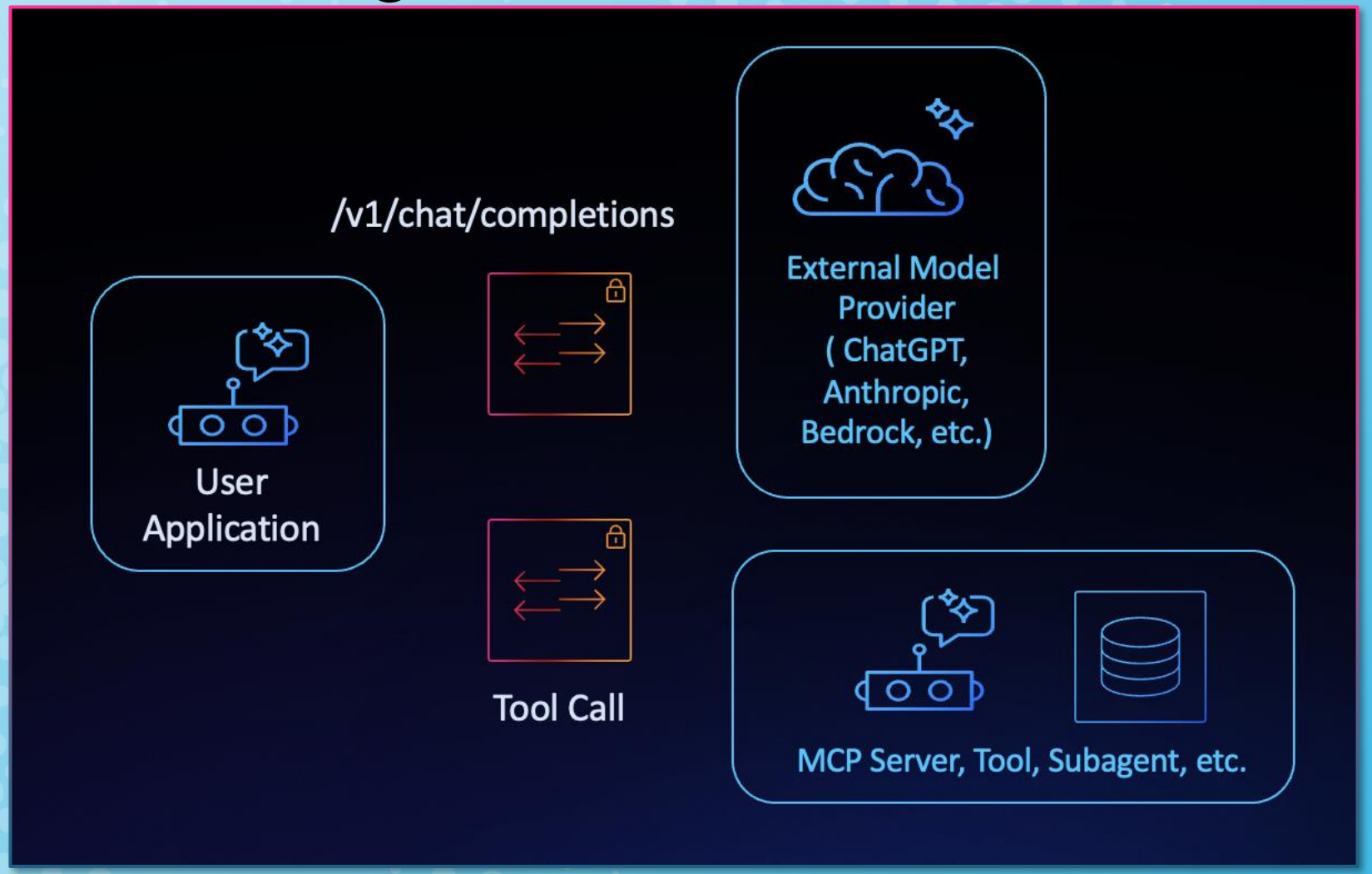
Tiny ≥ Function/Code

The Four A's of Agentic Security

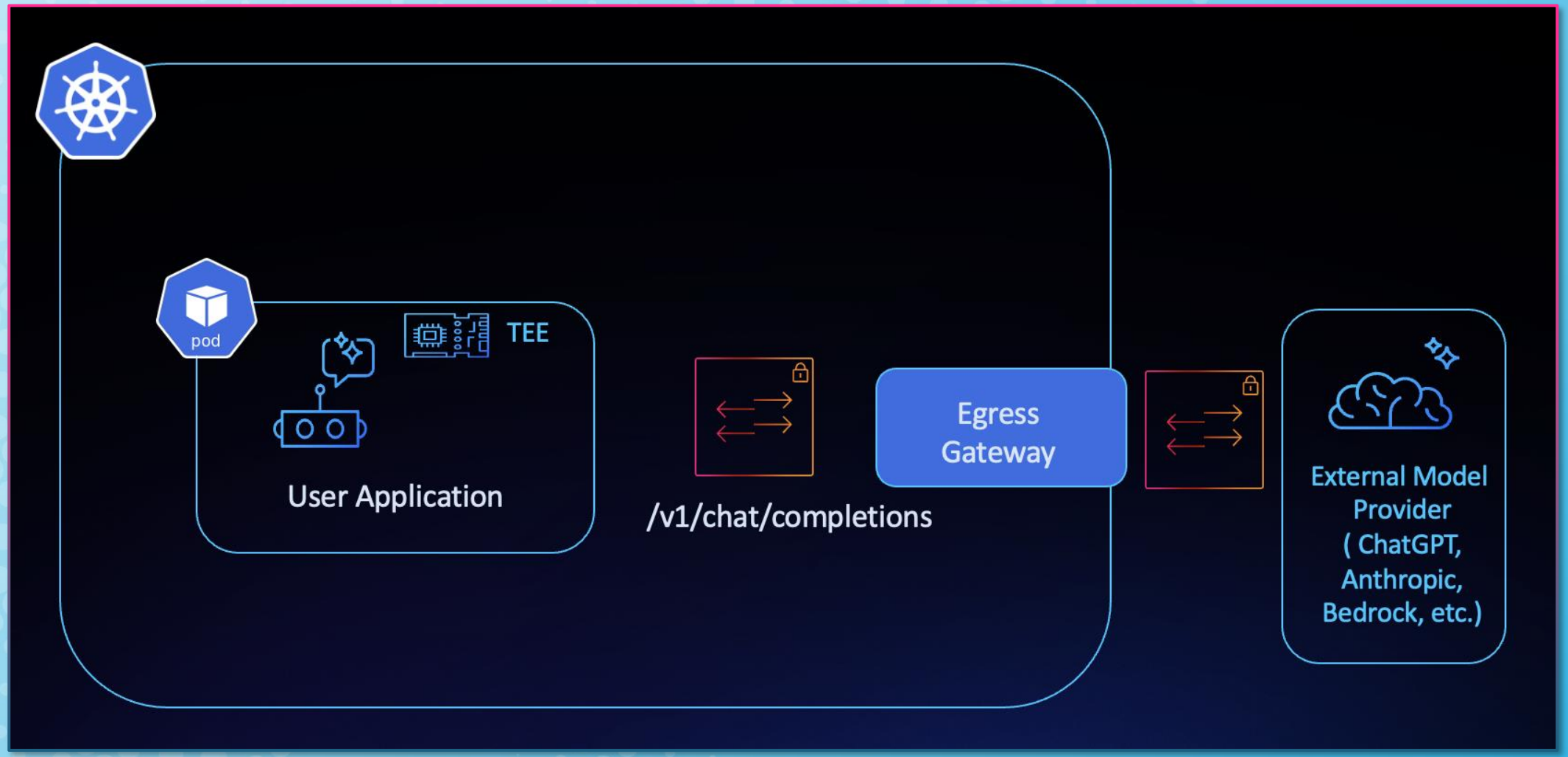


Agentic Pre-requisites

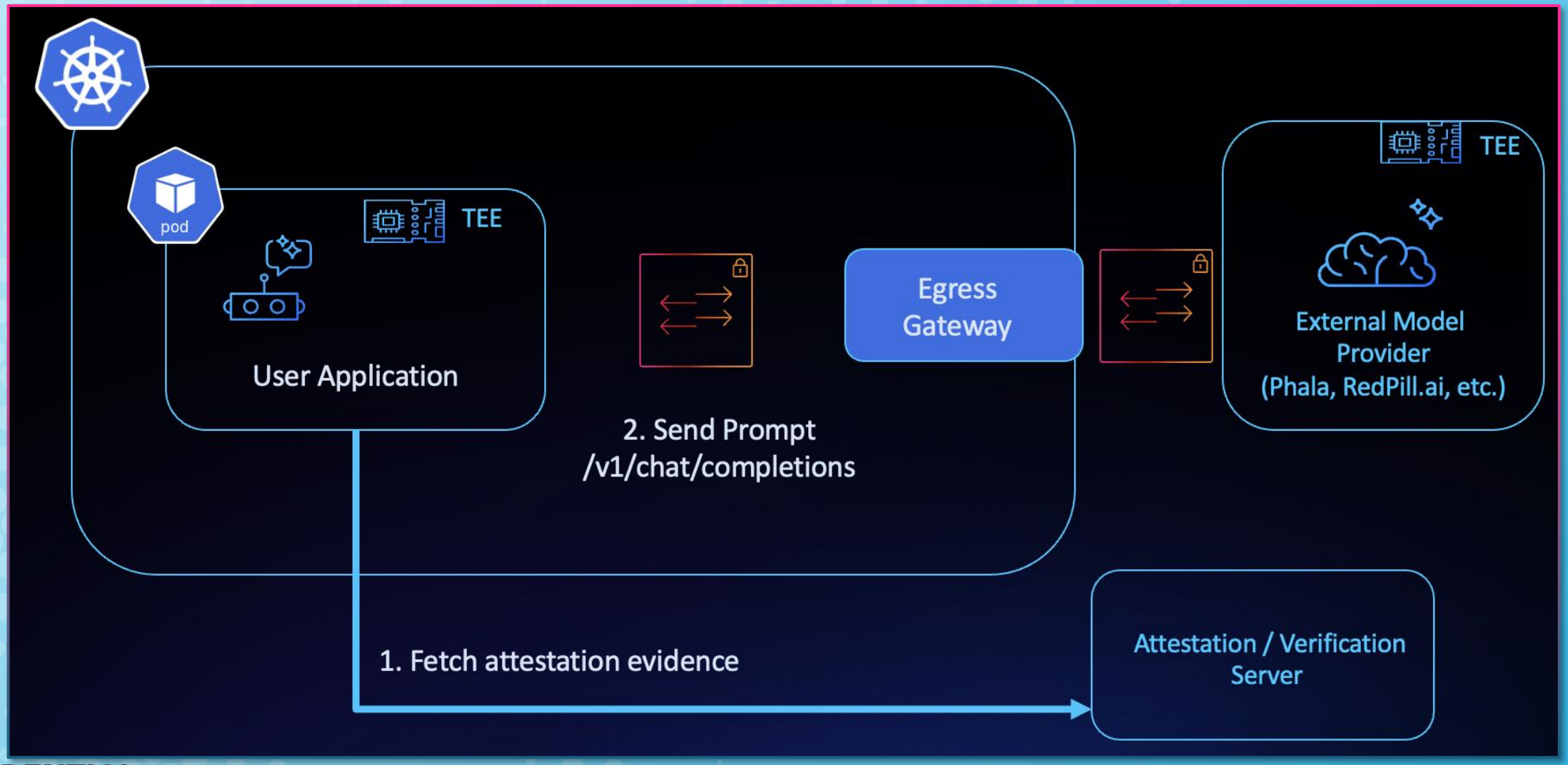
What Does an Agent Need?



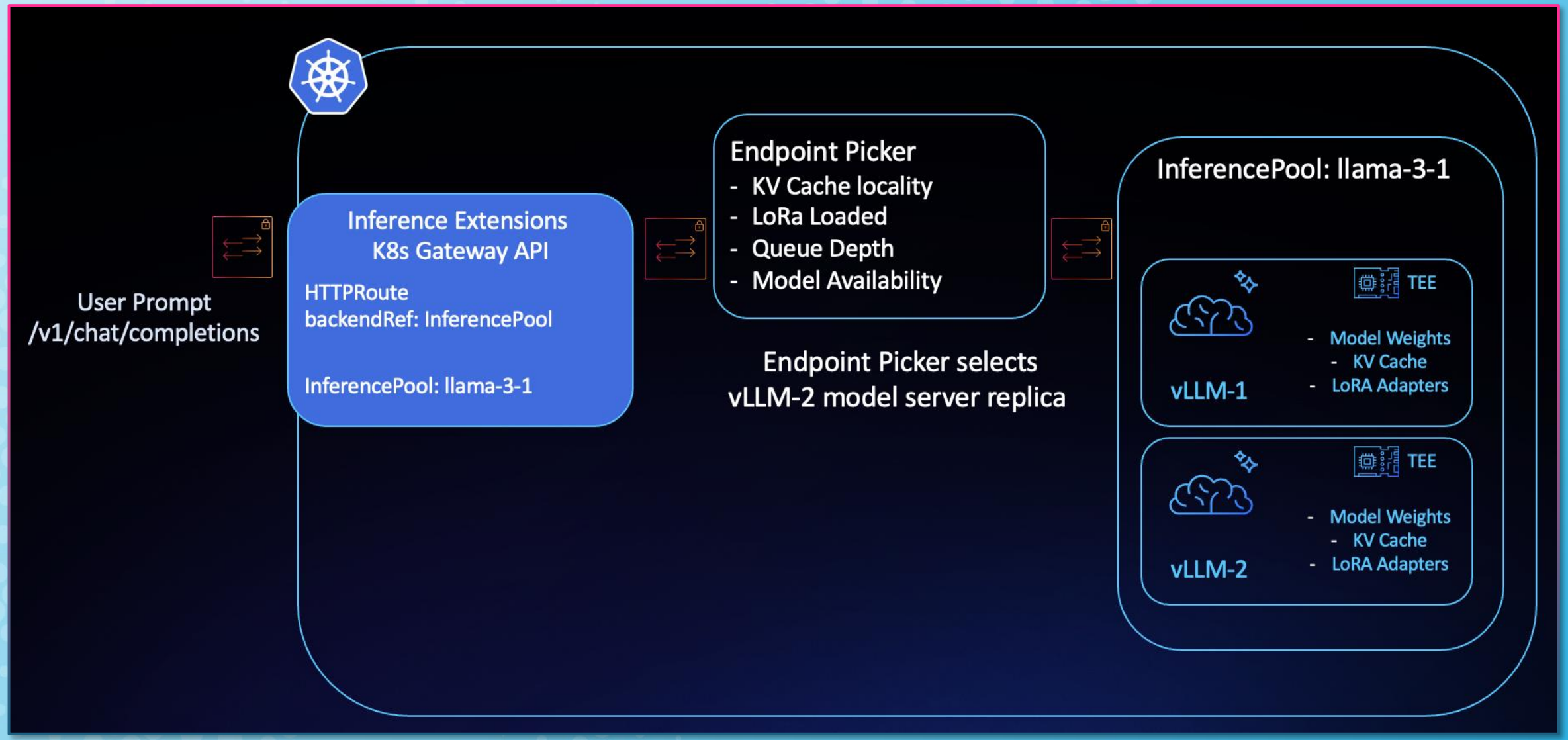
Confidential Computing for Inference?



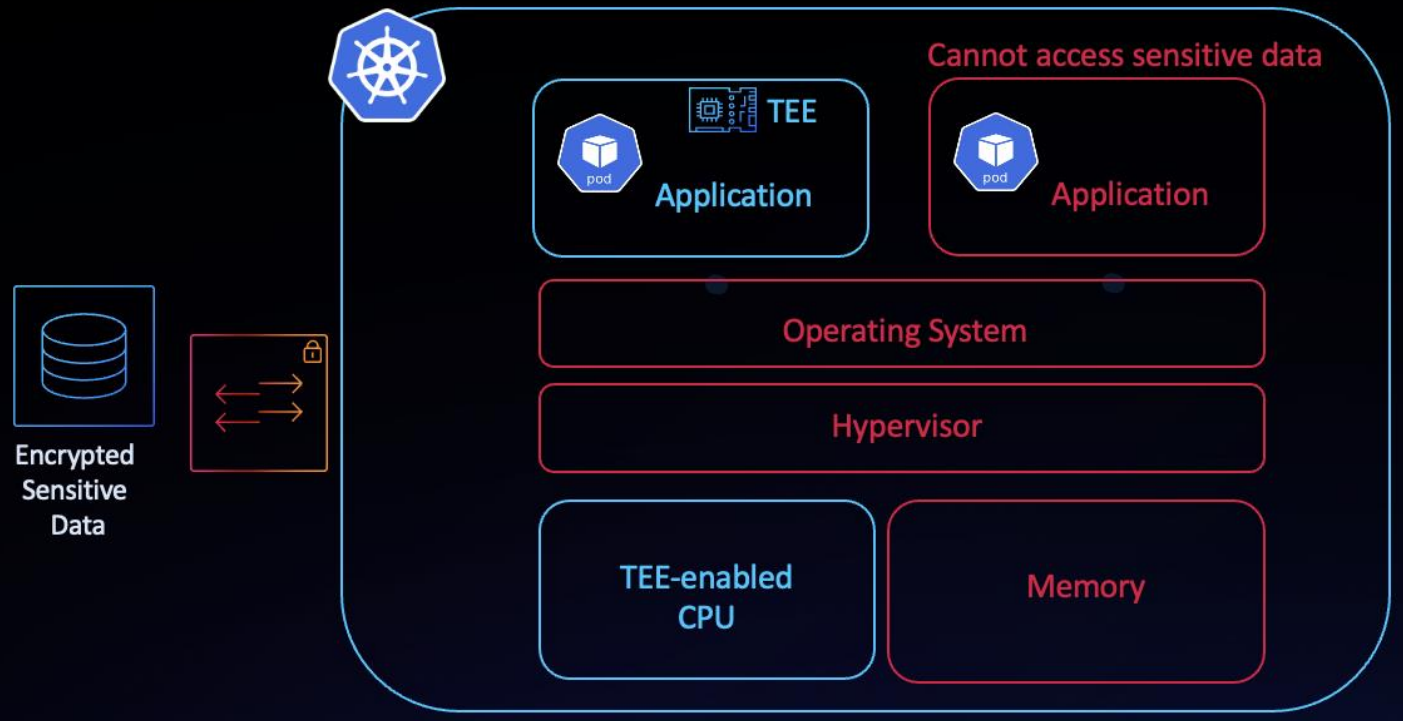
Confidential Inference: Managed Provider



Confidential Inference: Self Hosted?



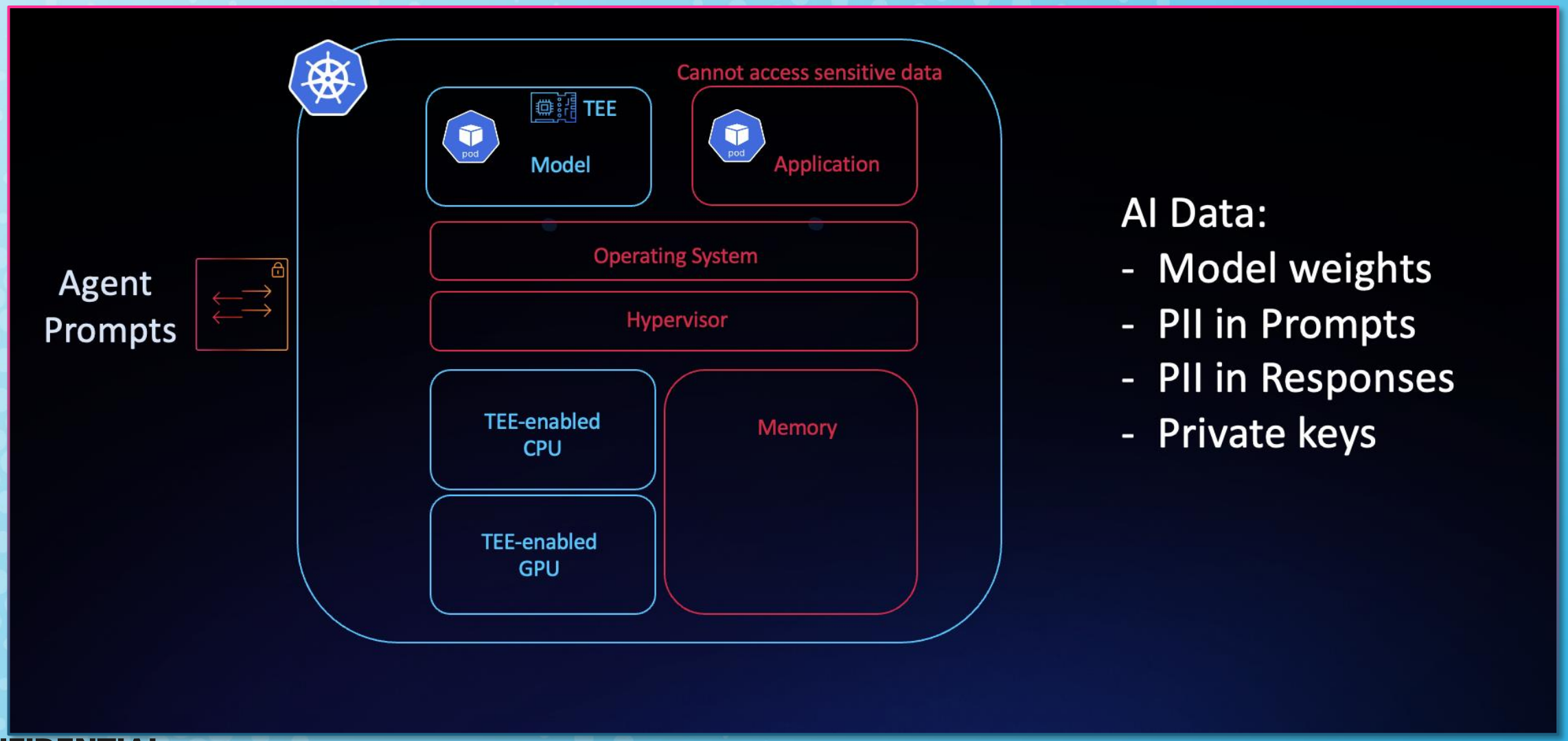
Trusted Execution Environment (Confidential Computing)



Data is securely decrypted and processed in TEE

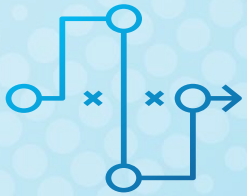
- Isolated compute
- Encrypted memory
- Attestation: hardware generated proof

AI Trusted Execution Environment?

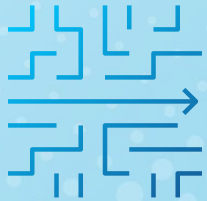


AI Data:
- Model weights
- PII in Prompts
- PII in Responses
- Private keys

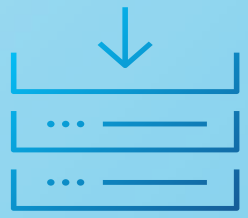
Requirements



Data *in use* is protected — for the CPU *and* the GPU.



The system can *prove* it's genuine (attestation end-to-end).



Data *at rest* (the disk) needs to be secured

Use Case


Scoop Dreams & Rocky Road
PROTECT AI DATA WHILE IT'S BEING SCOOPED
AI AGENTS IN ACTION
RECEIVE REQUEST
UNDERSTAND INTENT
RETRIEVE & USE DATA
REASON & DECIDE
TAKE ACTION
LEARN & IMPROVE
CONFIDENTIAL COMPUTING KEEPS EVERY STEP PRIVATE—EVEN WHILE IT'S HAPPENING.

Agent Architecture – Simple Reflex Pattern

Synthetic Record
(+cardinality)

- Name
- Social Security #
- Primary Account #
- API Key
- CANARY

PII

Agent Extracts
Sensitive Data

Stage 1:
Extraction

Predict(ExtractSensitive)
LM = gpt-oss-20b (GPU0)

Sensitive fields
sent for Agent
based post
classification

sensitive_fields (string)

Stage 2:
Classification

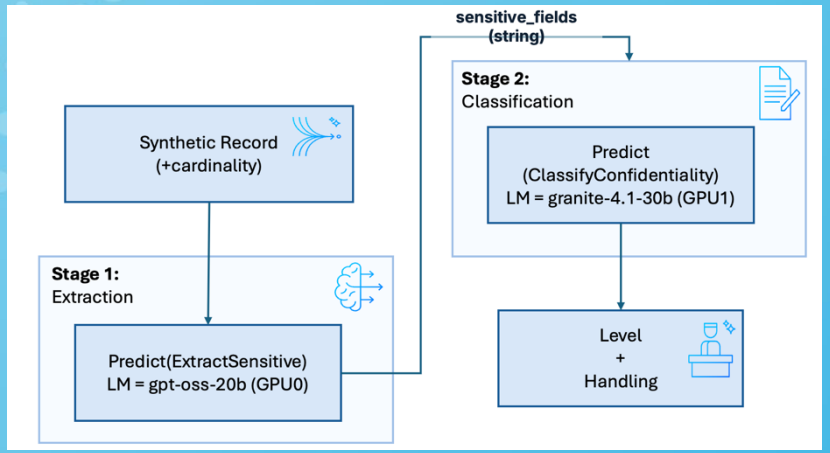
Predict
(ClassifyConfidentiality)
LM = granite-4.1-30b (GPU1)

Level
+
Handling

- RESTRICTED
- Encrypt at Rest

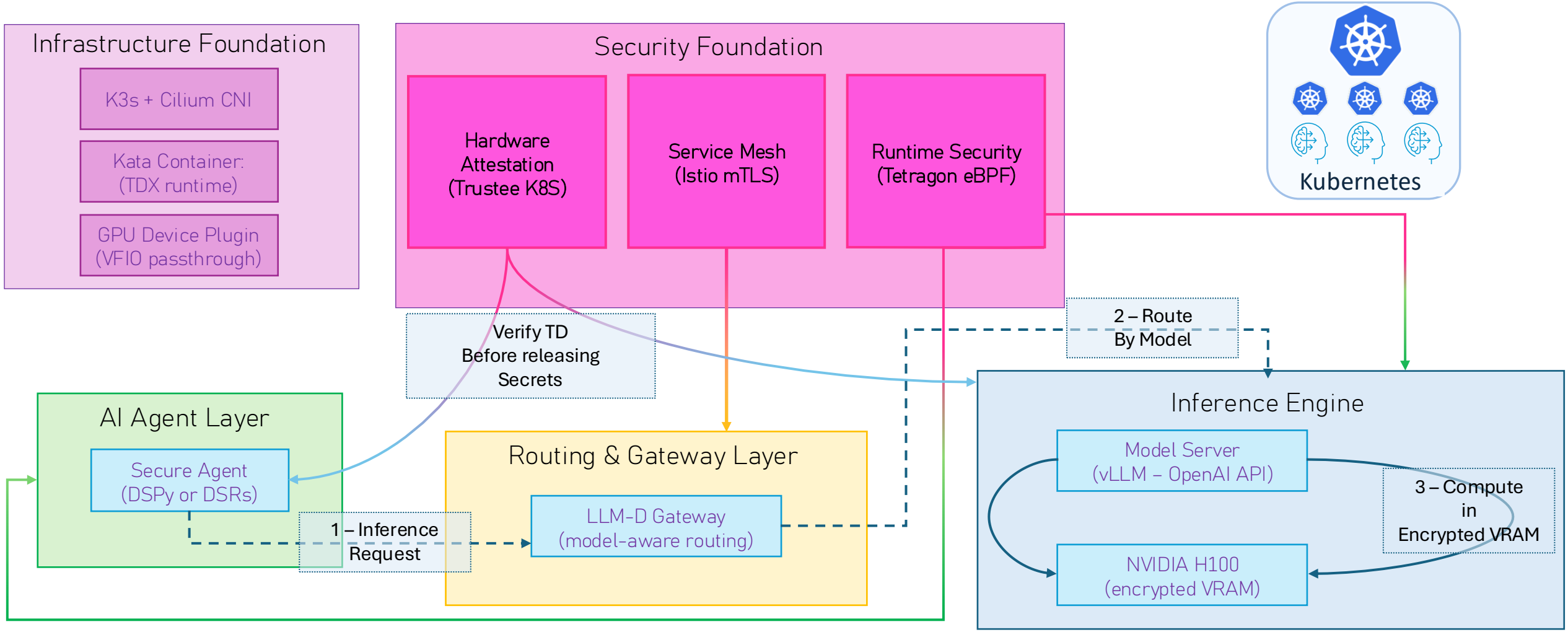
Agent Architecture – Framework Security

Metric	Python/DSPy	Rust/DSRs	Factor
Binary/image size	~150 MB (python:3.12-slim)	5.7 MB binary, 11 MB image	**26x smaller**
Cold start (to first SENT)	30-60 s (pip + import)	< 1 s (exec)	**30-60x faster**
Runtime RSS	~80-120 MB	~12-16 MB	**6-8x less**



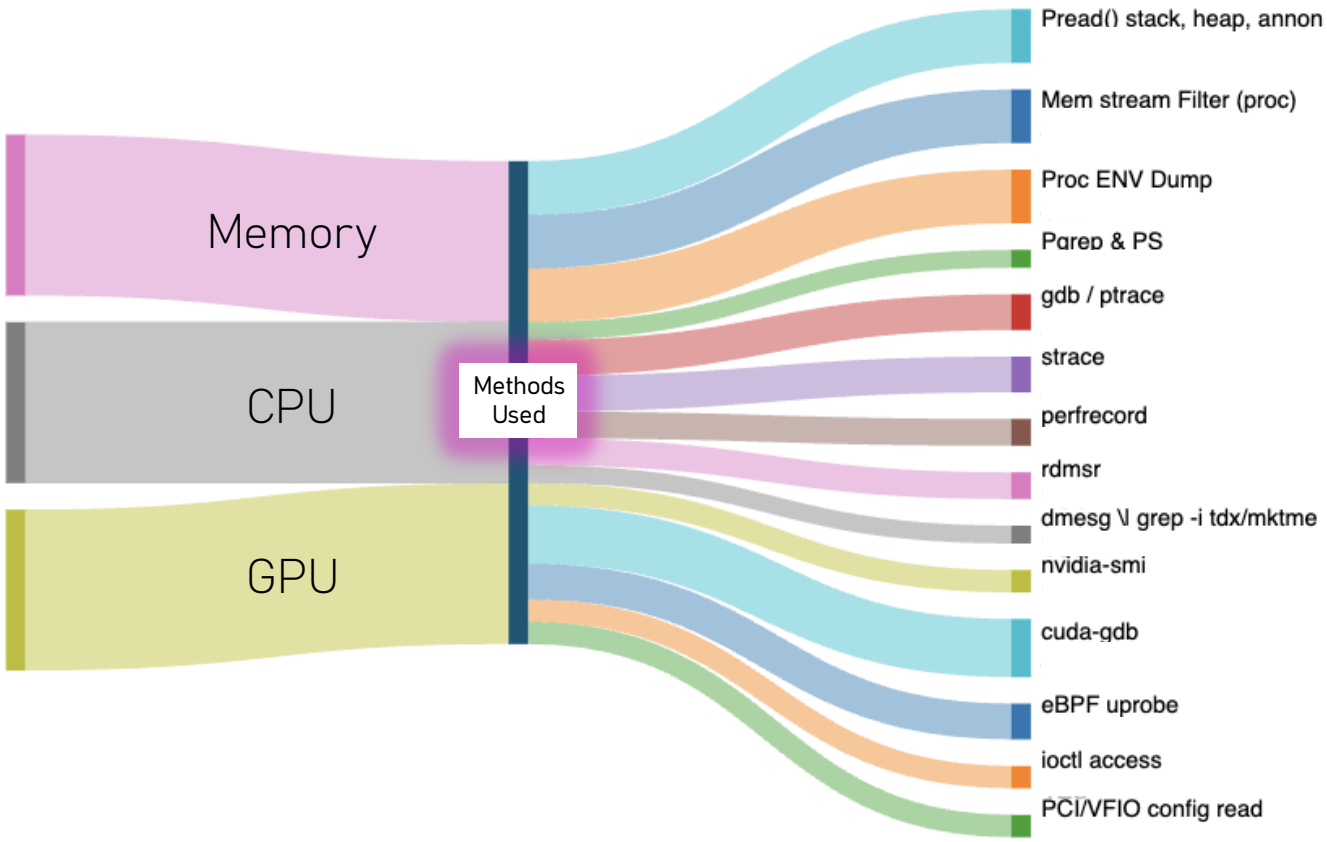
Attack Surface	Rust Framework – Security relevant observations
Runtime memory hygiene	Ownership + drop: secrets freed within the loop tick
Buffer overflows	Rust eliminates a CVE class that matters inside a trust boundary
Supply chain at boot	Rust eliminates runtime supply-chain risk; Python's pip-in-TD is a known gap
Process isolation	Rust is cleaner for async multi-model pipelines
Attestation coverage	binary IS the workload -- the attestation measurement covers the complete code
Dependency audit	~15 Rust crates, all compiled to safe Rust (auditable via cargo audit)

Cluster Overview



Runtime Exploit Levers

Substrate	Method Attempted	Without CC	With CC
Memory	/proc/<pid>/mem scrape (memscan.py); regex grep over mem; read /proc/<pid>/environ + cmdline; read the container rootfs / stdout logs on the host	CAPTURED — process RAM, env, and records are all root-readable	BLOCKED — private RAM is guest_memfd, not host-mapped (pread → EIO)
CPU	enumerate the process (ps/pgrep); attach a debugger (gdb -p/ptrace); trace syscalls (strace -p); sample (perf); boundary recon (rdmsr SEAMRR, dmesg TDX/MK-TME)	CAPTURED — the workload is a host process: debuggable, traceable, sampleable	BLOCKED — no host process; guest vCPU state is TD-protected; DRAM is MK-TME-encrypted
GPU	nvidia-smi (NVML); cuda-gdb attach + VRAM dump; eBPF uprobe on libcuda; open /dev/nvidia*; PCI/VFIO config-space read	CAPTURED — the driver lists the GPU and its VRAM is dumpable	BLOCKED — no host driver / /dev/nvidia* / libcuda / target process; VRAM is CC-encrypted



Examples of Exploitability

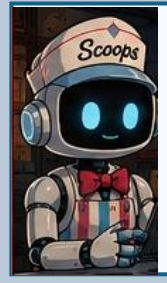
Data in Memory

Host / root-compromised Node Access

Kubernetes Cluster

Worker Node

pod



Agent Application

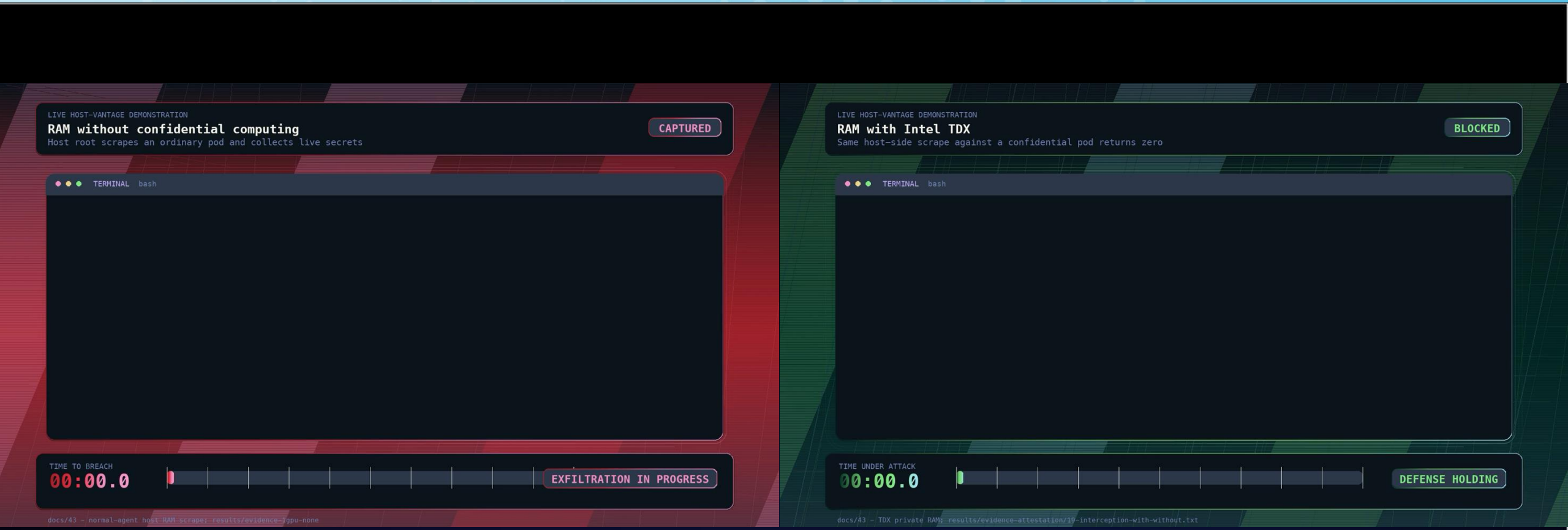
/proc/<pid>/mem

Process RAM

SSN
PAN
API Keys

PII Stolen!
SSN
PAN
API Key

Data in Memory – Before and After



Data in Memory: Secured ☑

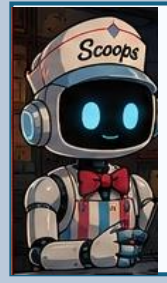
Host / root-compromised Node Access

TDX Confidential VM

Kubernetes Cluster

Worker Node

pod



Agent Application

/proc/<pid>/mem

Private TD RAM

SSN
PAN
API Keys

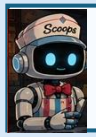
Data in Transit

Host / root-compromised Node Access

Kubernetes Cluster

Worker Node

pod



Agent Application

Process RAM

shared buffers in shared I/O memory

HTTP POST /v1/chat/completions

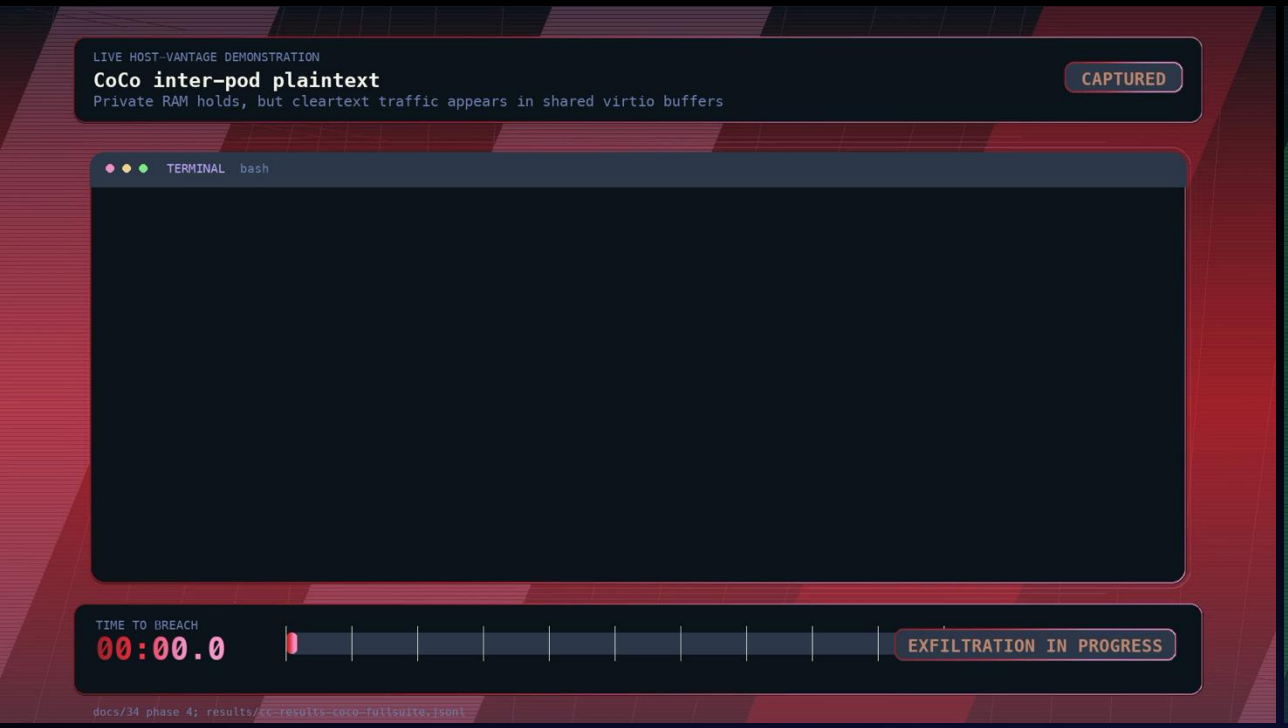
pod

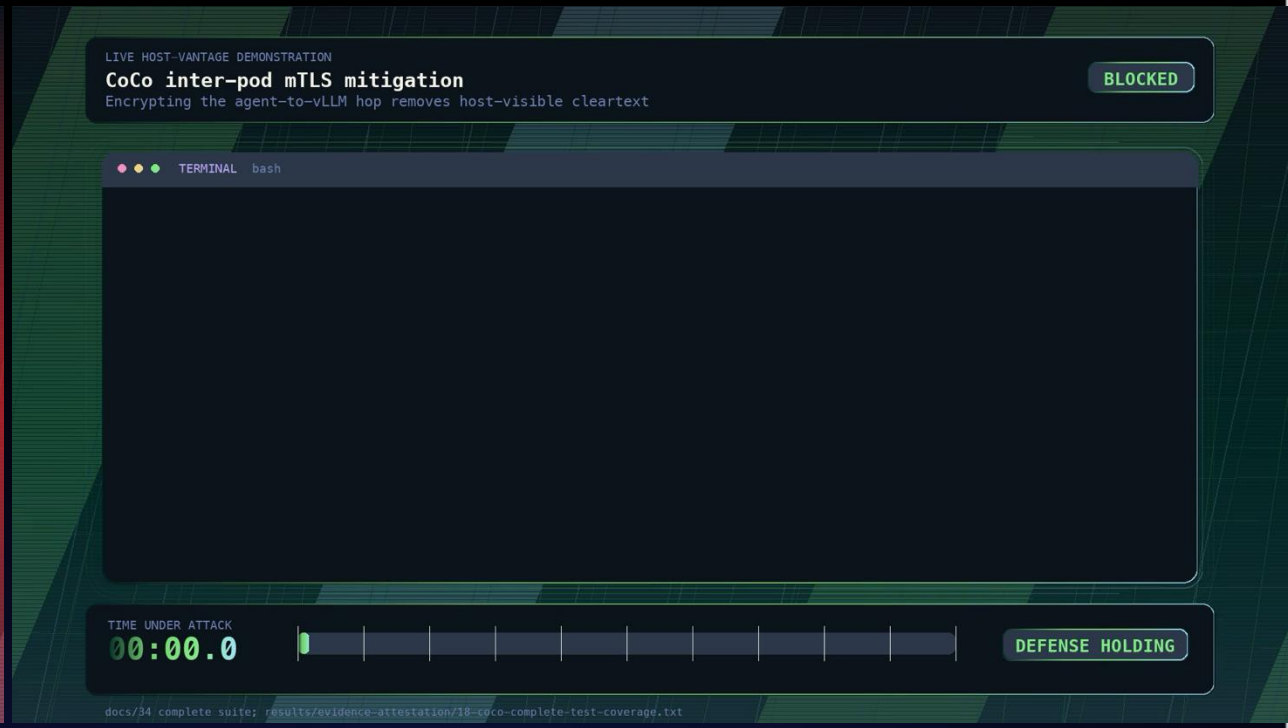
vLLM endpoint

PII Stolen!
SSN
PAN
API Key

Hubble inter-pods traffic extraction

Data in Transit – Before and After





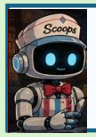
Data in Transit: Secured ☑

Host / root-compromised Node Access

Kubernetes Cluster

Worker Node

CoCo



Agent Application

Private TD RAM

HTTP POST /v1/chat/completions

(encrypted with mTLS)

CoCo

vLLM endpoint

Hubble inter-pods traffic extraction

Data in GPU

Host / root-compromised Node Access

Kubernetes Cluster

Worker Node

Shared Components

- NVIDIA Driver
- NVIDIA Device plugin
- GPU Operator / telemetry
- GPU extraction tools

```
$ nvidia-smi
...
PII recovered
```

Agent Application

vLLM endpoint

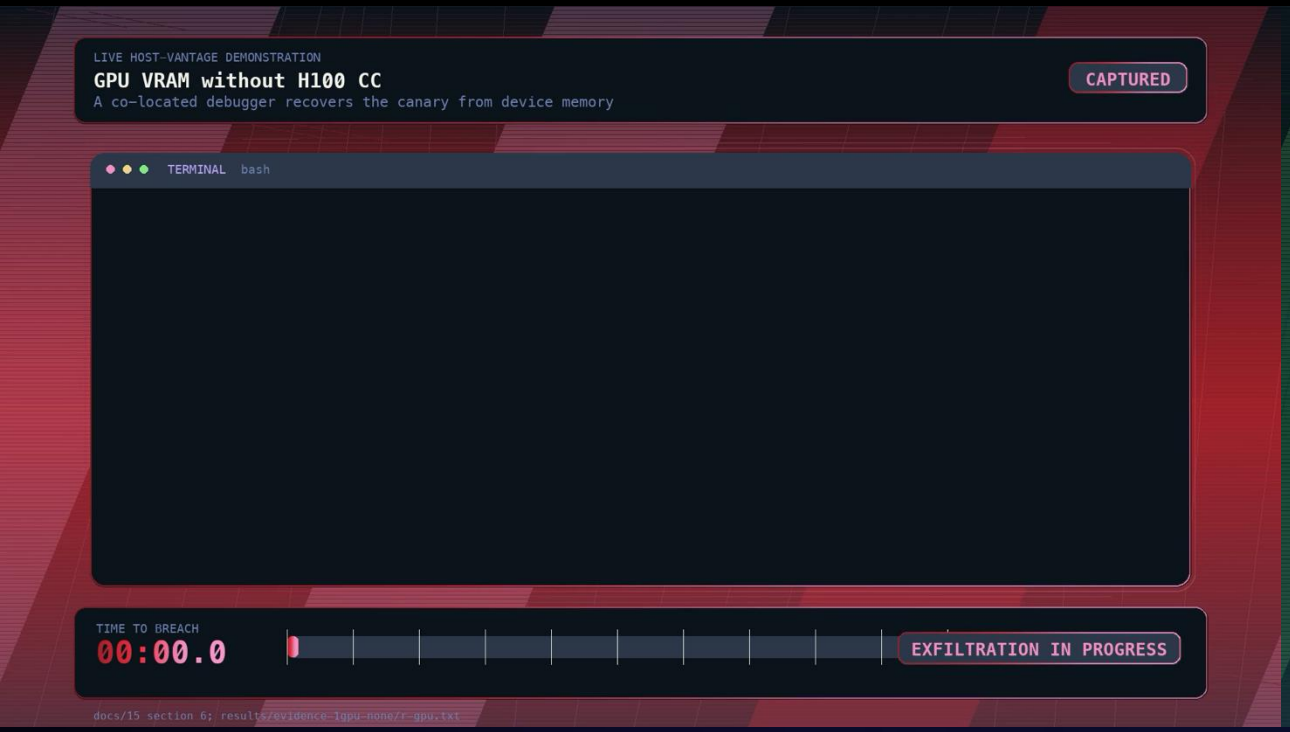
CUDA runtime

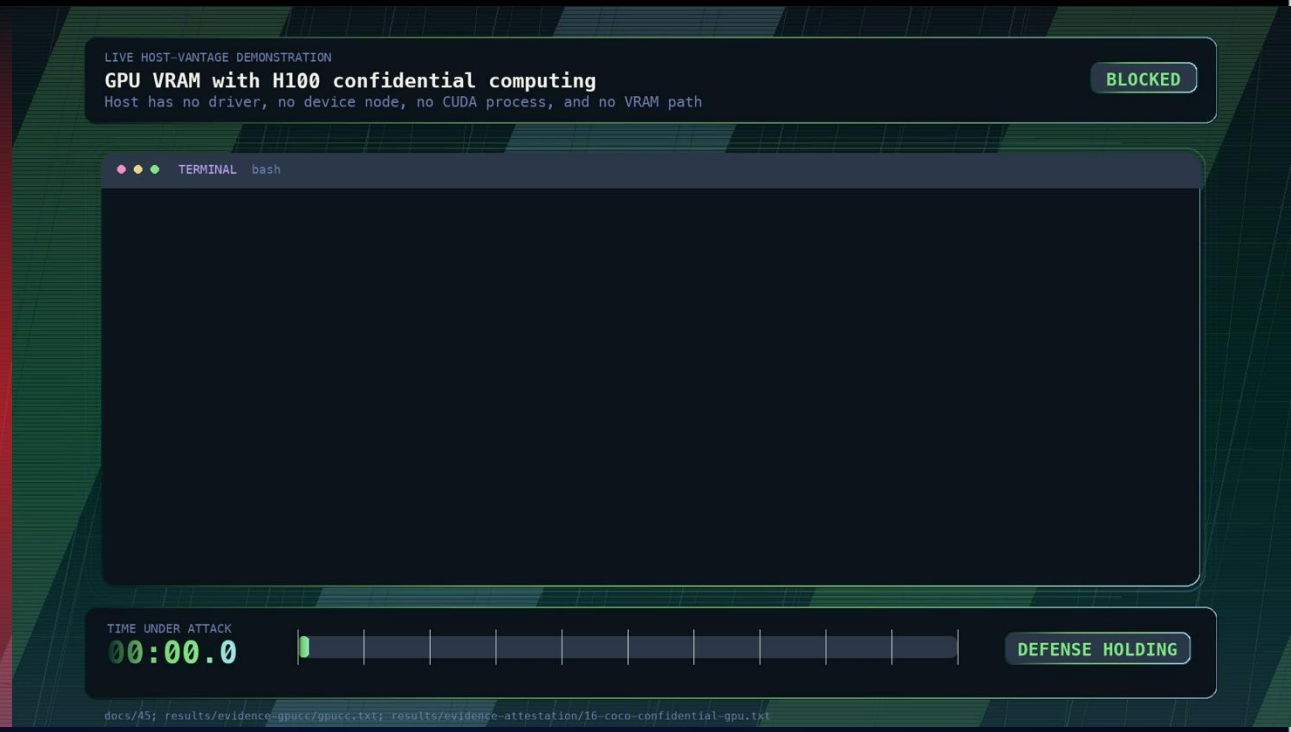
NVIDIA H100 GPU

H100 VRAM (plaintext, inspectable)

Model data, prompts, activations, API data

Data in GPU – Before and After





Data in GPU: Secured ☑

Host / root-compromised Node Access

Kubernetes Cluster

Worker Node

```
$ nvidia-smi
nvidia-smi: command
not found

$ ls /dev/nvidia*
(no /dev/nvidia*
device nodes)

...
0 bytes recovered
```

No host driver
No NVIDIA device nodes
No host CUDA process
No host VRAM

Agent Application

vLLM endpoint

CUDA runtime

NVIDIA H100 GPU (CC mode)

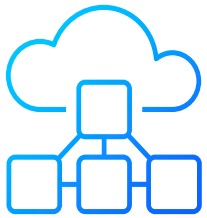
H100 VRAM (Encrypted)

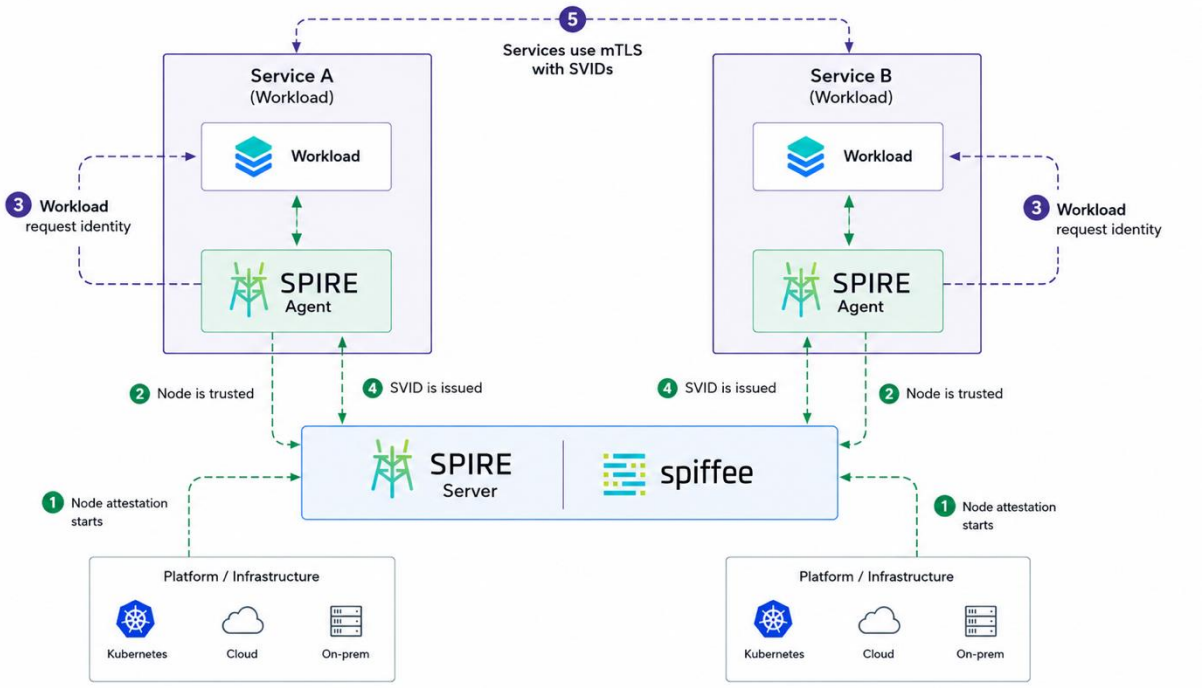
Model data, prompts, activations, API data

Bringing it all together

Attestation of the Workload and Network

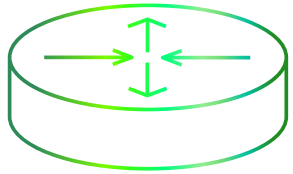
Workloads:





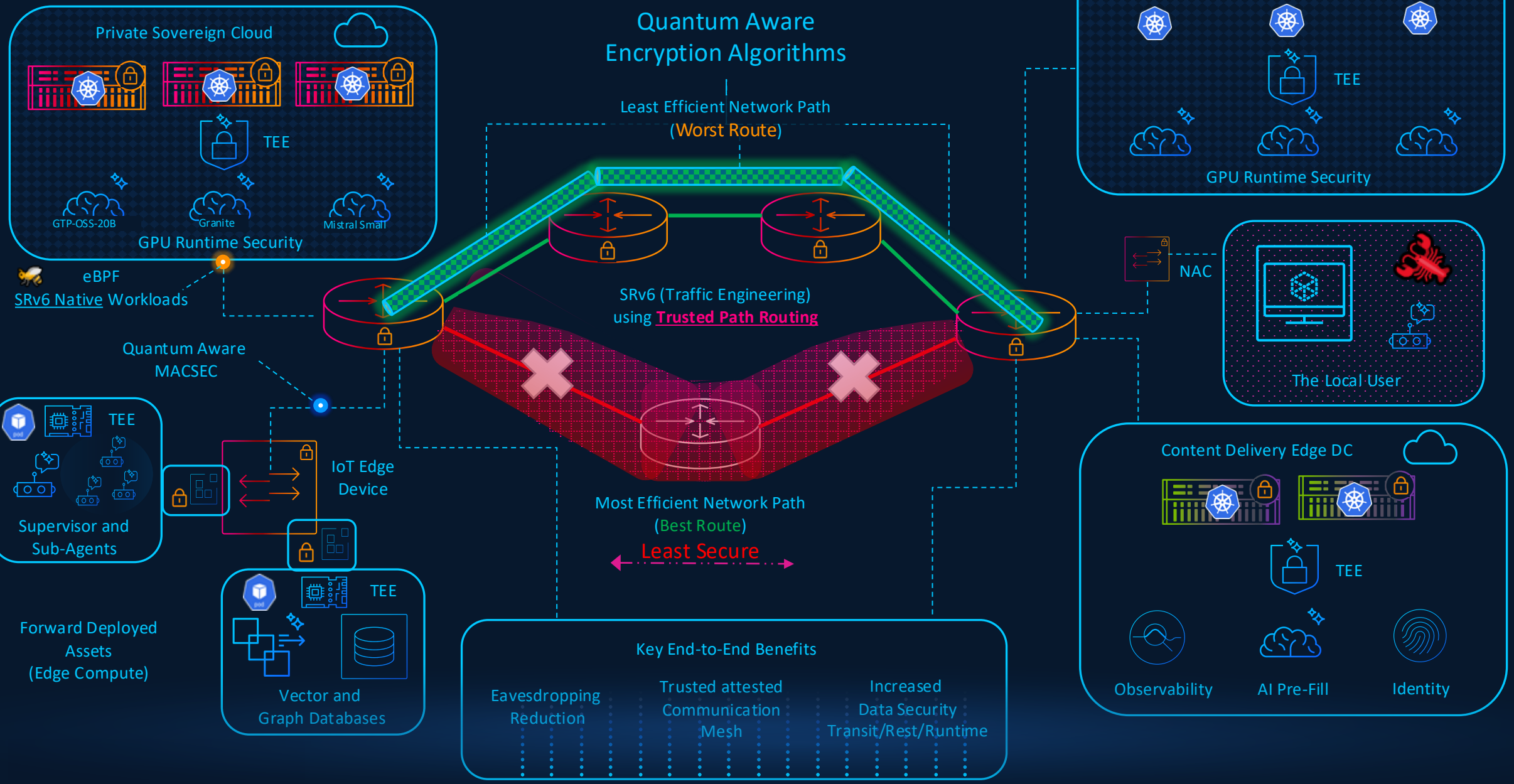
Platform Configuration Registers

Network Device:



PCR Index	Measurement
0	Firmware / BIOS / boot ROM code
1	Firmware / platform configuration data
2	Option ROM / expansion code
3	Option ROM configuration/data
4	Boot loader (MBR / IPL) code
5	Boot loader configuration & GPT
6	Platform state transitions / events
7	Secure Boot policy & keys
8+	OS / kernel measurements (platform-specific)

Achieving Sovereign Security as Scale



Recap

Agentic Security Doesn't Stop Here!

Agent Identity	Governance
Tool Authorization	Explainability
Guardrails	Benchmarking (validation)

What we learnt along the way....

- <u>Zero Trust focuses on removing the weakest link</u>
  - GPU
  - CPU
  - Memory
  - Transit
  - Rest

Thank You!!!