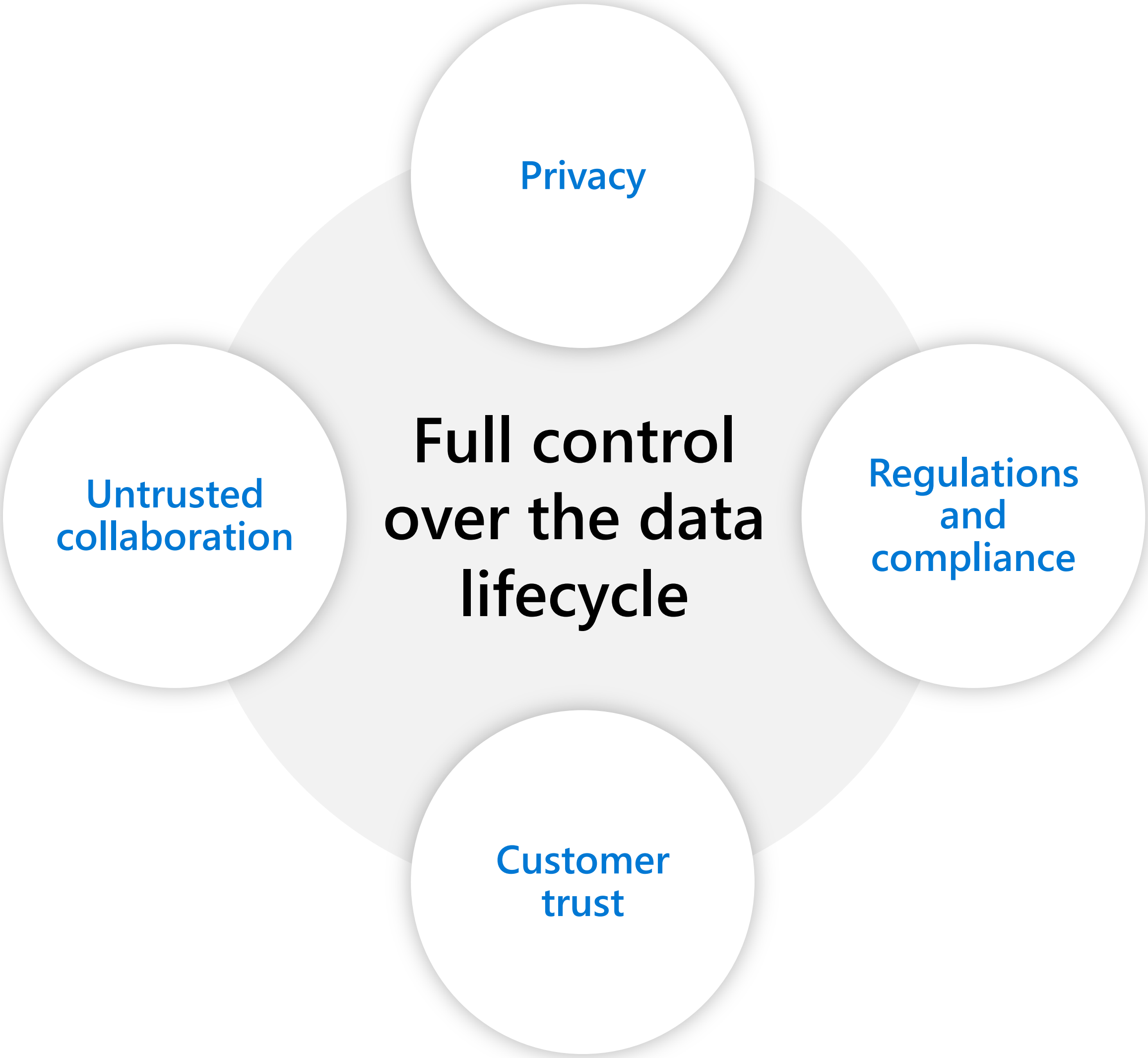
Confidential Computing in the Cloud

1. Overview
2. Hardware acceleration for ML
3. Software updates & transparency

Cédric Fournet
Confidential Computing - Microsoft Research

Cloud customers are increasingly looking for ways to trust as little as possible



Data protection

EXISTING ENCRYPTION

Data at rest

Encrypt inactive data when stored in blob storage, database, etc.

Data in transit

Encrypt data that is flowing between untrusted public or private networks

CONFIDENTIAL COMPUTING

Data in use

Protect/encrypt data that is in use, while in RAM, and during computation

CONFIDENTIAL COMPUTING

Data in use

Protect/encrypt data that is in use, while in RAM, and during computation

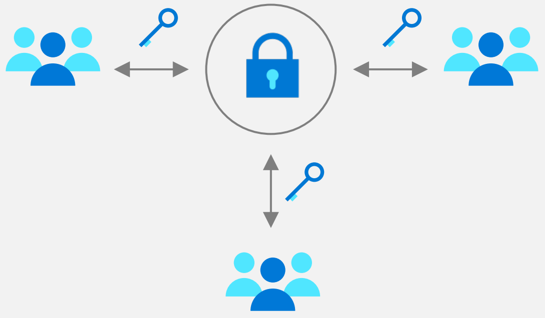
Defense in depth from others

- Malicious admins
- Hackers
- Access without consent

Protect customer data from myself & platform

- Guest OS
  Host OS kernel
- VM admin
  Host admin
- Hypervisor
  Physical hardware access

Share data with multi-party securely



Public cloud can be “Private cloud”

Government

- Digital identity
- Critical infrastructure
- Anti-corruption
- Cyber crime prevention
- Judicial proceedings and case management
- Deployed and disconnected operations
- Safeguarding / vulnerable population protection (including child exploitation, human trafficking, etc.)

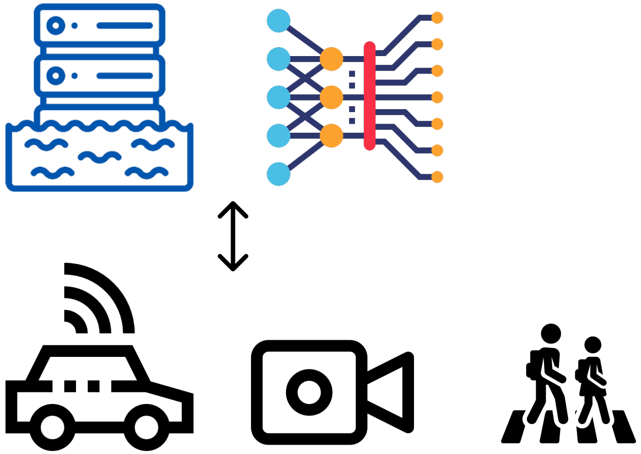
Financial Services

- Anti-money laundering
- Digital currencies
- Secure Payment Processing including Credit Card and Bank Transactions
- Fraud prevention
- Credit risk assessment and qualification from combined bank records
- Capital Markets e.g.: Securing Quantitative Hedge Funds code and models
- Proprietary analytics / algorithms

Healthcare

- Disease diagnostic
- Insurance fraud prevention
- Drug development
- Contact tracing
- Records and evidence management
- Insurance fraud, waste, and abuse prevention

Bosch Research – Autonomous Driving (AD)



Developing models for AD requires collecting, storing, and processing of **PII*-poisoned sensor data** from cars

GDPR requires lawful basis for processing AD data, which could be **consent**, which is challenging,

or **legitimate interest**, which requires, among others, that **there is no less intrusive way to achieve the same result**

Confidential Computing allows for **least privacy intrusive storage and processing**, minimizing privacy and legal risk, and for leap-frogging competition

* PII = Personally Identifiable Information

Bosch Research – AD Training Pipeline

BOSCH

Acquisition → Transmission → De-identification → Storage → Semantic Labeling → Training

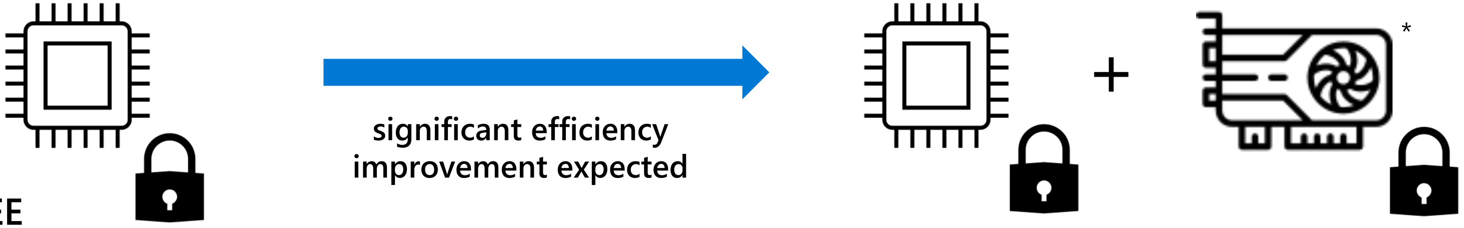
Secure transfer of data into de-identification TEE

Detection and separation of image regions with PII in a CPU TEE

Separate storage of PII and non-PII data

Semantic labeling of non-PII footage

Training with original and labelled footage in CPU TEE



CPU TEE

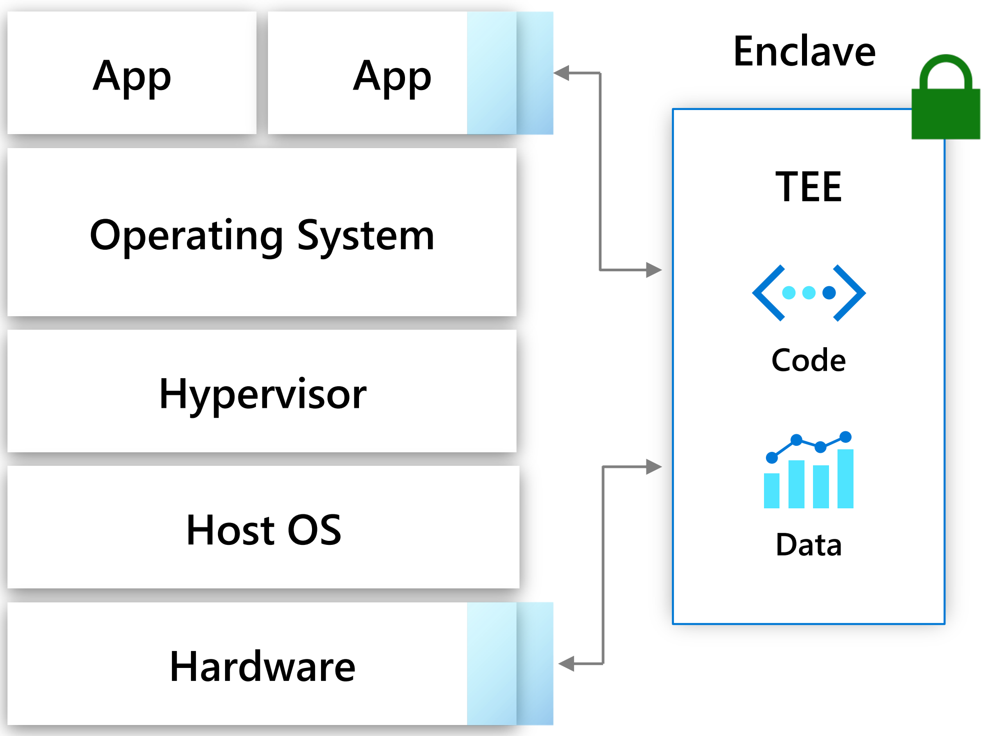
* Graphics Card icon by Icons8

Trusted Execution Environments (TEE)

Minimize attack surface to CPU

Isolates the code and data of a given confidential workload from any other code running in a system with encrypted memory

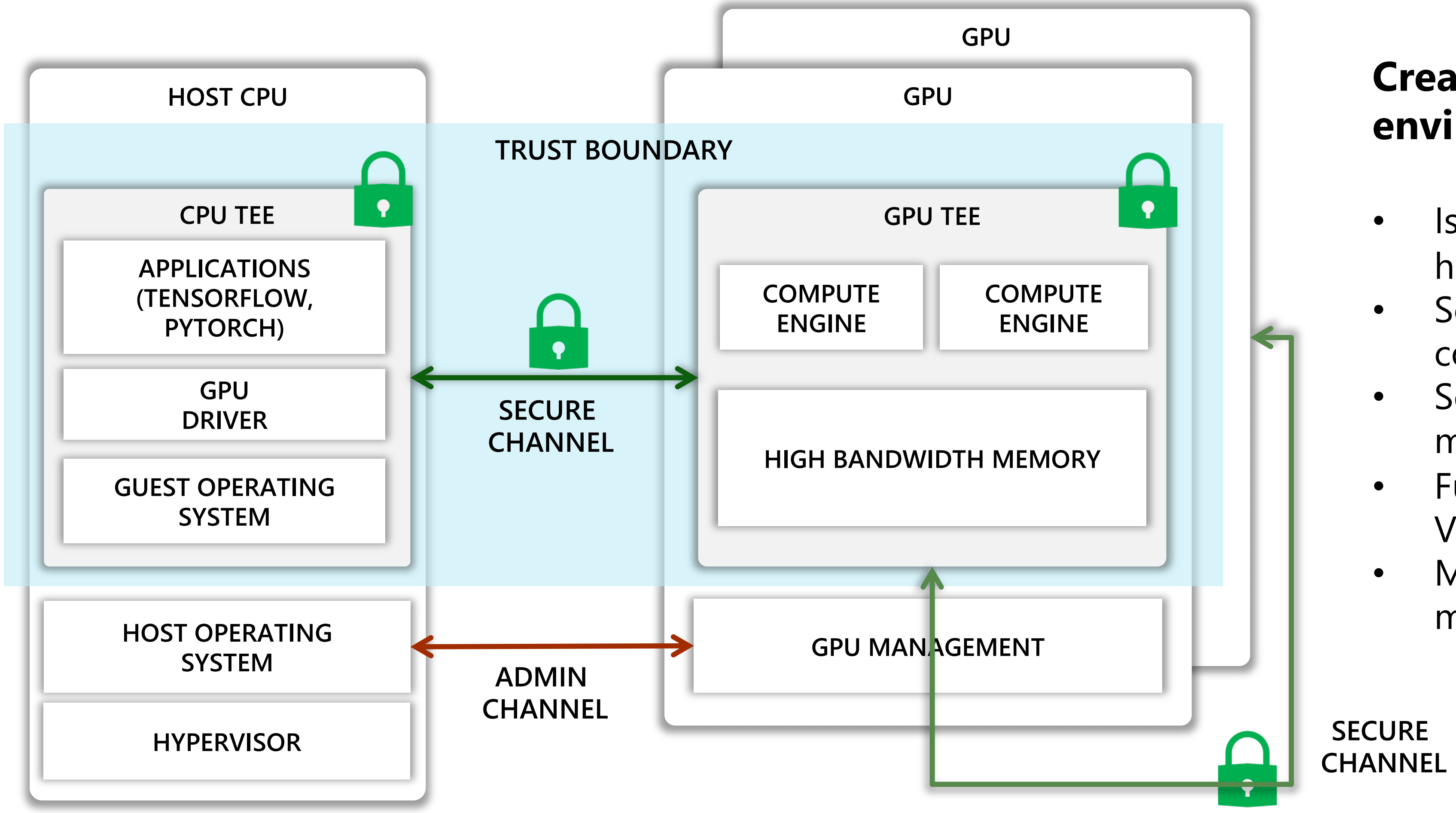
ENCLAVE AS AN EXAMPLE OF TEE



TEE Foundations

- Hardware root-of-trust
- Remote attestation
- Trusted launch
- Memory isolation and encryption
- Secure key management

Confidential AI Hardware



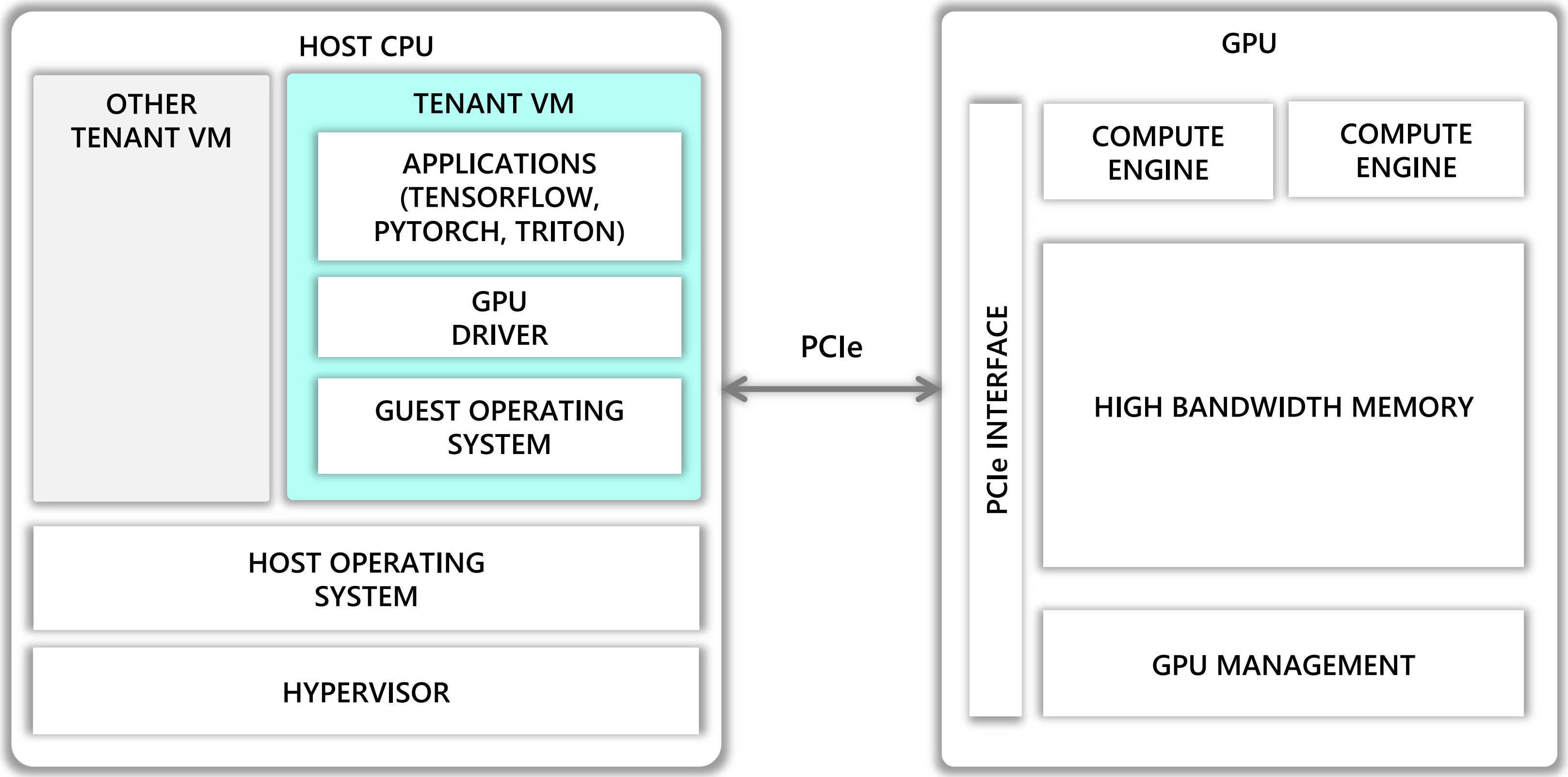
Create a unified trusted execution environment for GPU offload

- Isolate code and data from untrusted host and GPU management
- Securely and transparently transfer code and data between CPU and GPU
- Secure communication between multiple GPUs
- Full attestation of GPU state, including VBIOS and microcode
- Minimal impact on programming model or performance

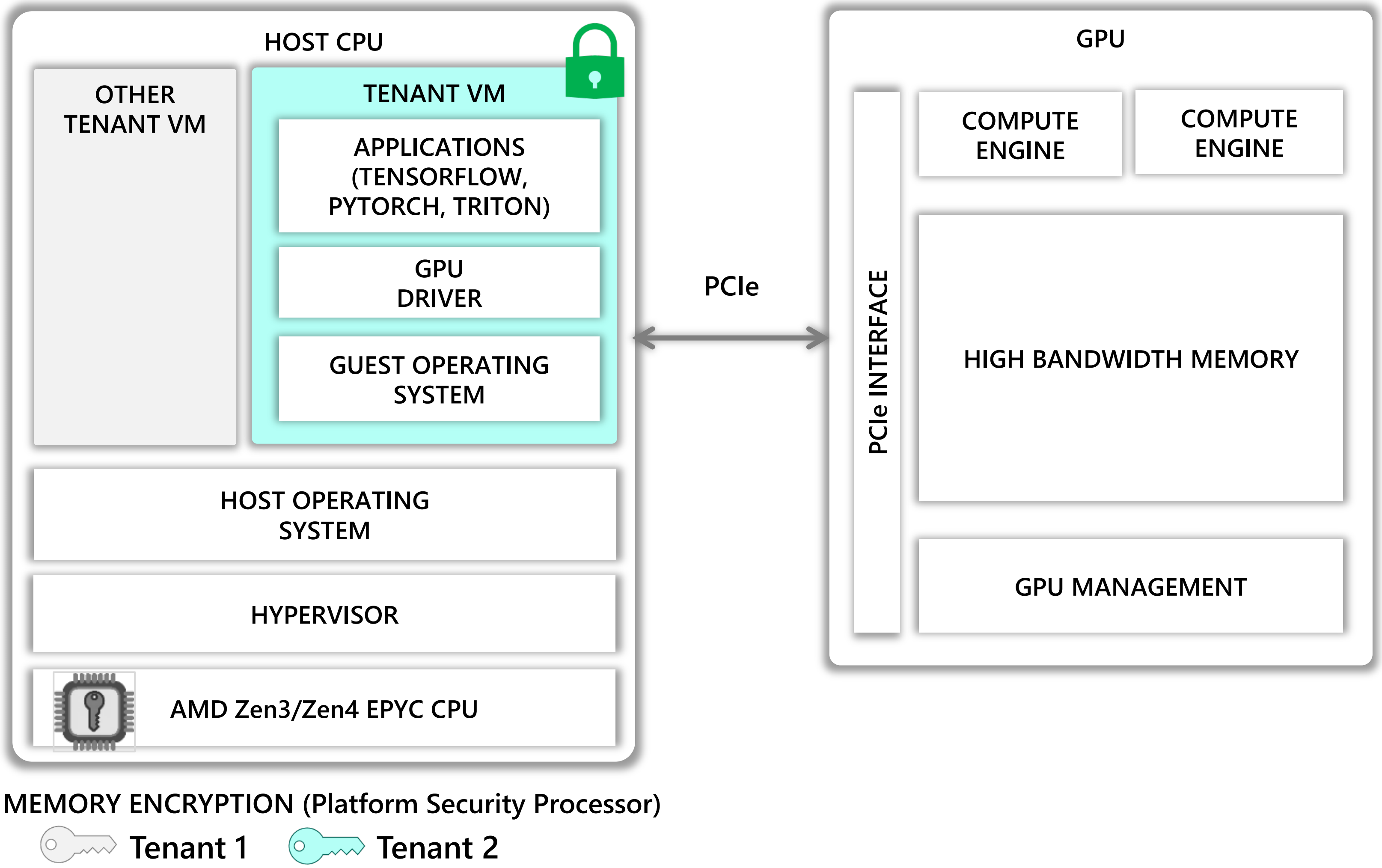
Oblivious Multi-party Machine Learning using Trusted Processors, S&P 2016
Graviton: Trusted Execution Environments on GPUs, OSDI 2018
Confidential Machine Learning within Graphcore IPUs (under submission)

Implementing Confidential Computing in NVIDIA GPUs

1. Isolate tenant VM



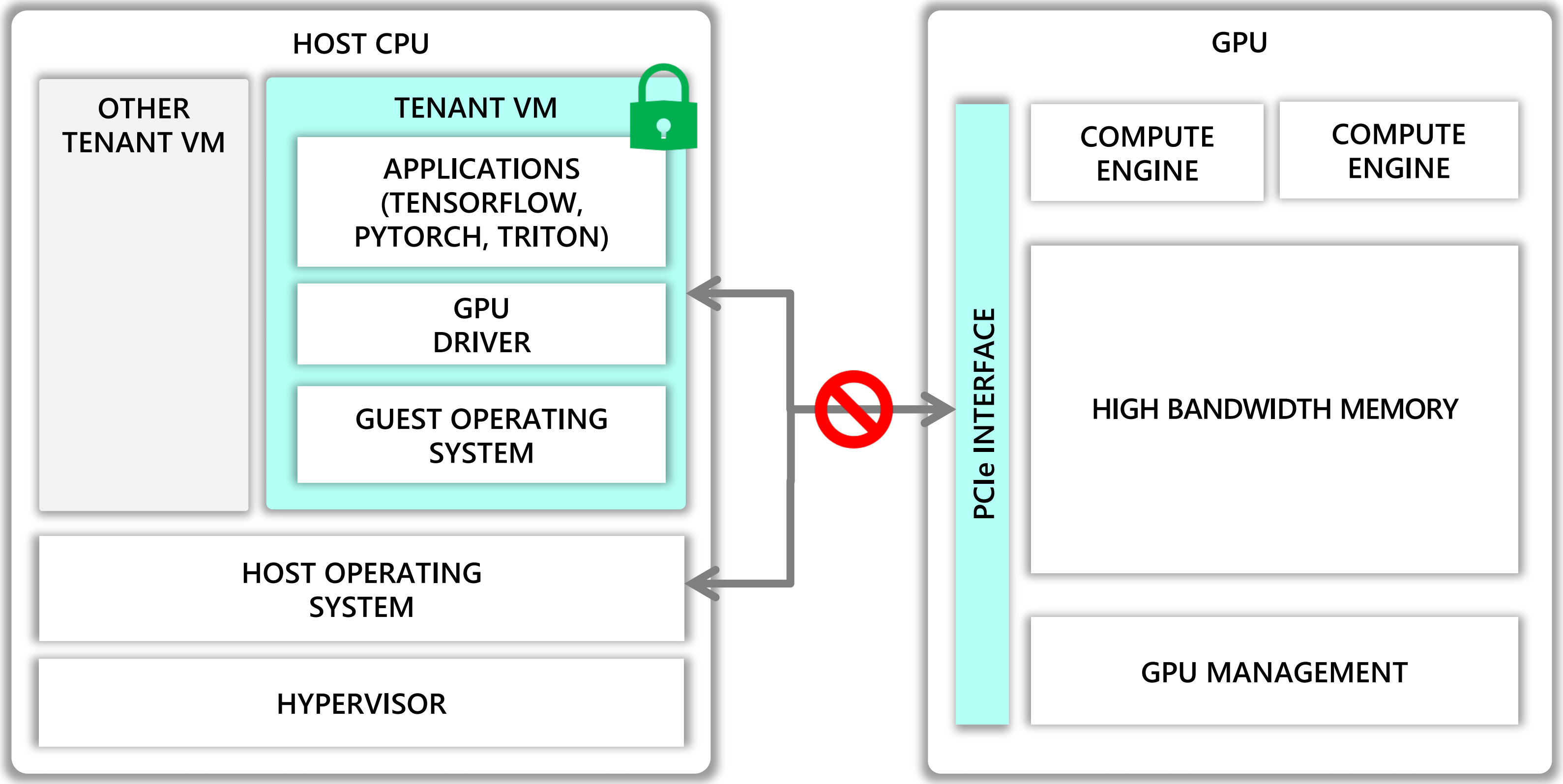
Implementing Confidential Computing in NVIDIA GPUs



1. Isolate tenant VM

- SEV-SNP encrypts VM memory using VM specific encryption keys

Isolate GPU Memory

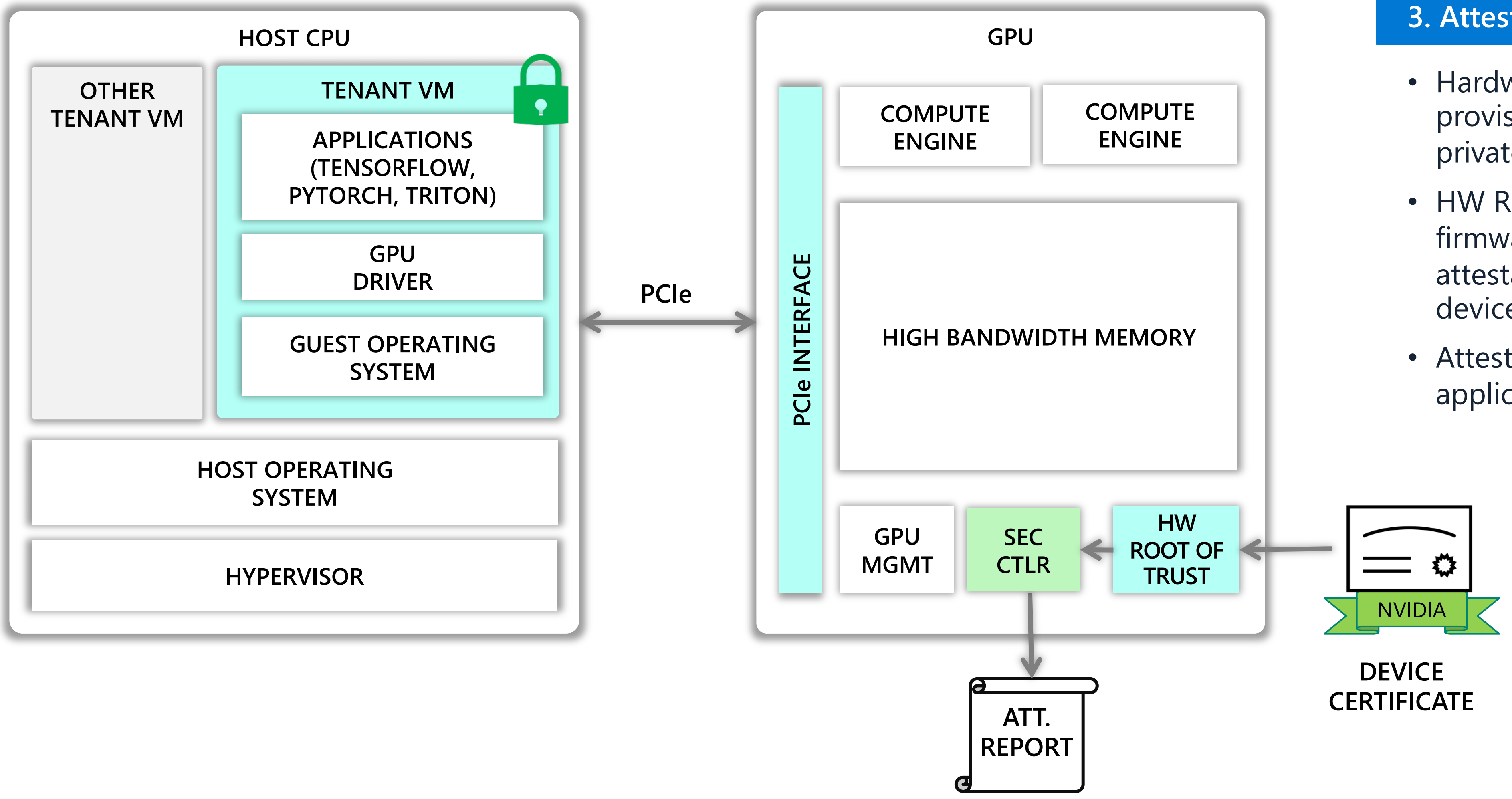


1. Isolate tenant VM

2. Isolate GPU Memory

- Block MMIO access to GPU memory and security sensitive configuration from the host and other GPUs
- Block outbound access from GPU engines unless encrypted

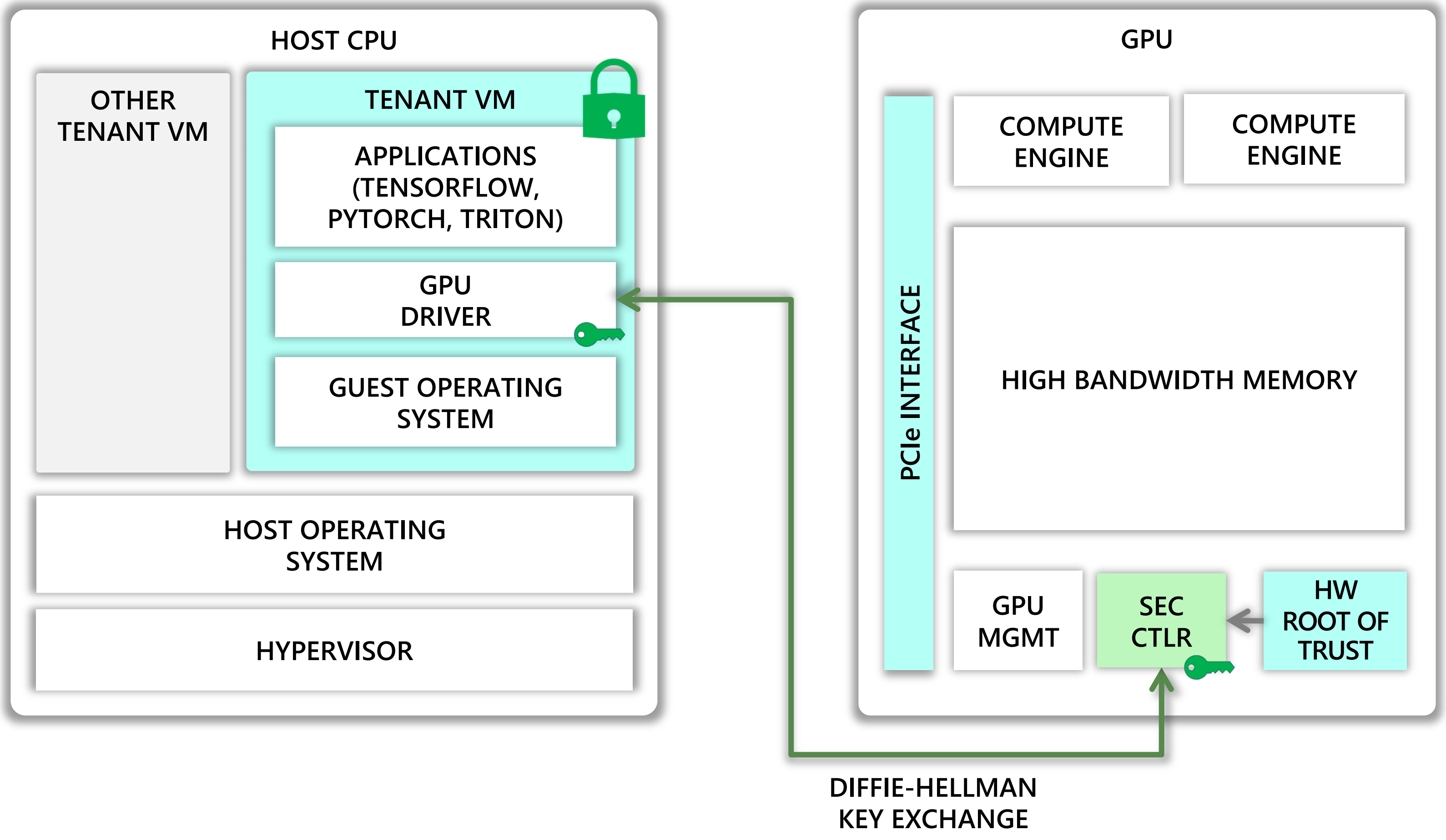
GPU Attestation



1. Isolate tenant VM
2. Isolate GPU Memory
3. Attest GPU state

- Hardware root-of-trust provisioned with device-specific private key endorsed by NVIDIA
- HW RoT measures device state, firmware state and generates the attestation report signed with device-specific key
- Attestation available to guest applications and the GPU driver

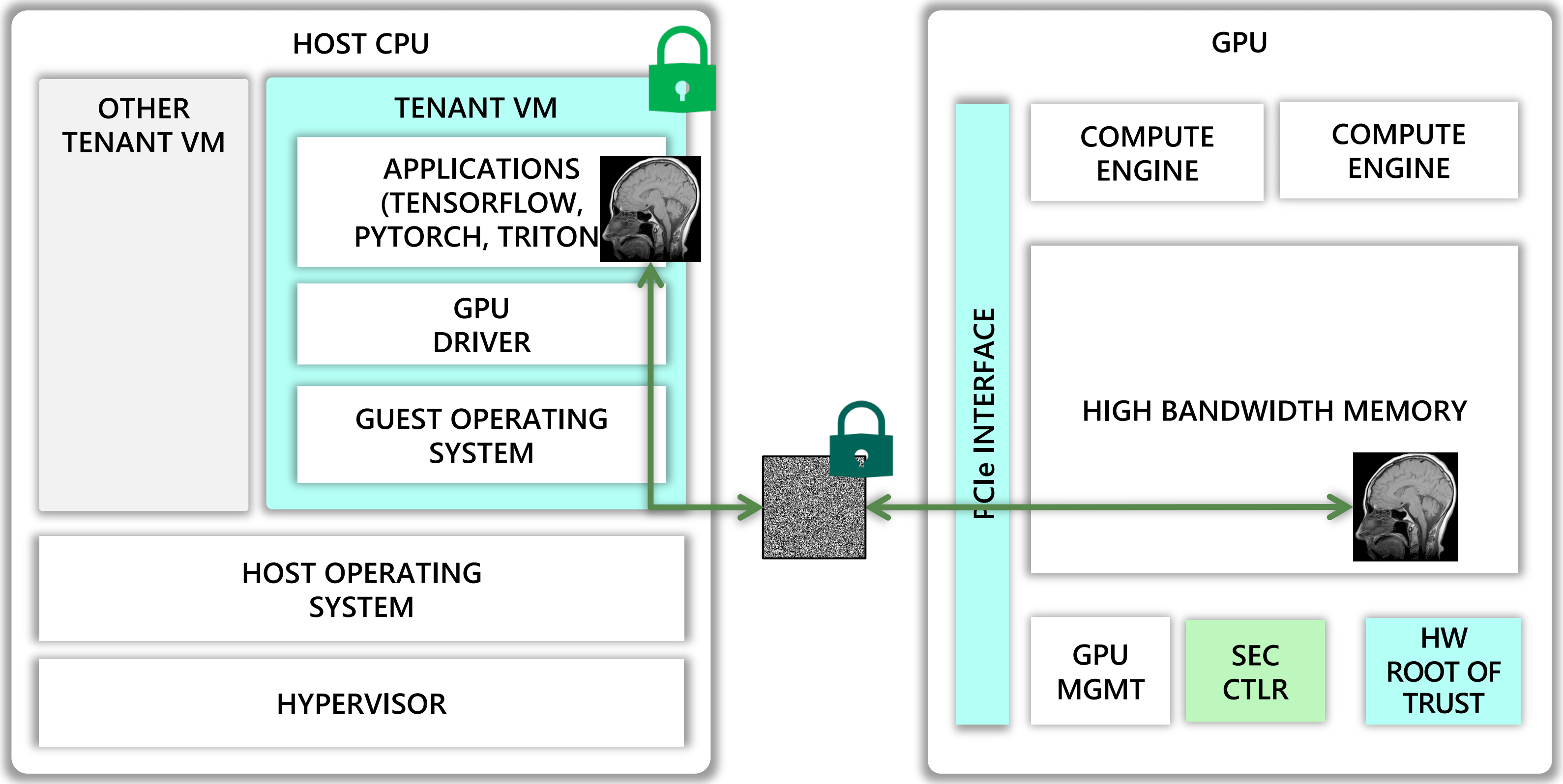
Secure Key Exchange



1. Isolate tenant VM
2. Isolate GPU Memory
3. Attest GPU state
4. Secure Key Exchange

- GPU driver verifies attestation
- SPDM backed Diffie-Hellman key exchange between hardware root-of-trust and GPU driver

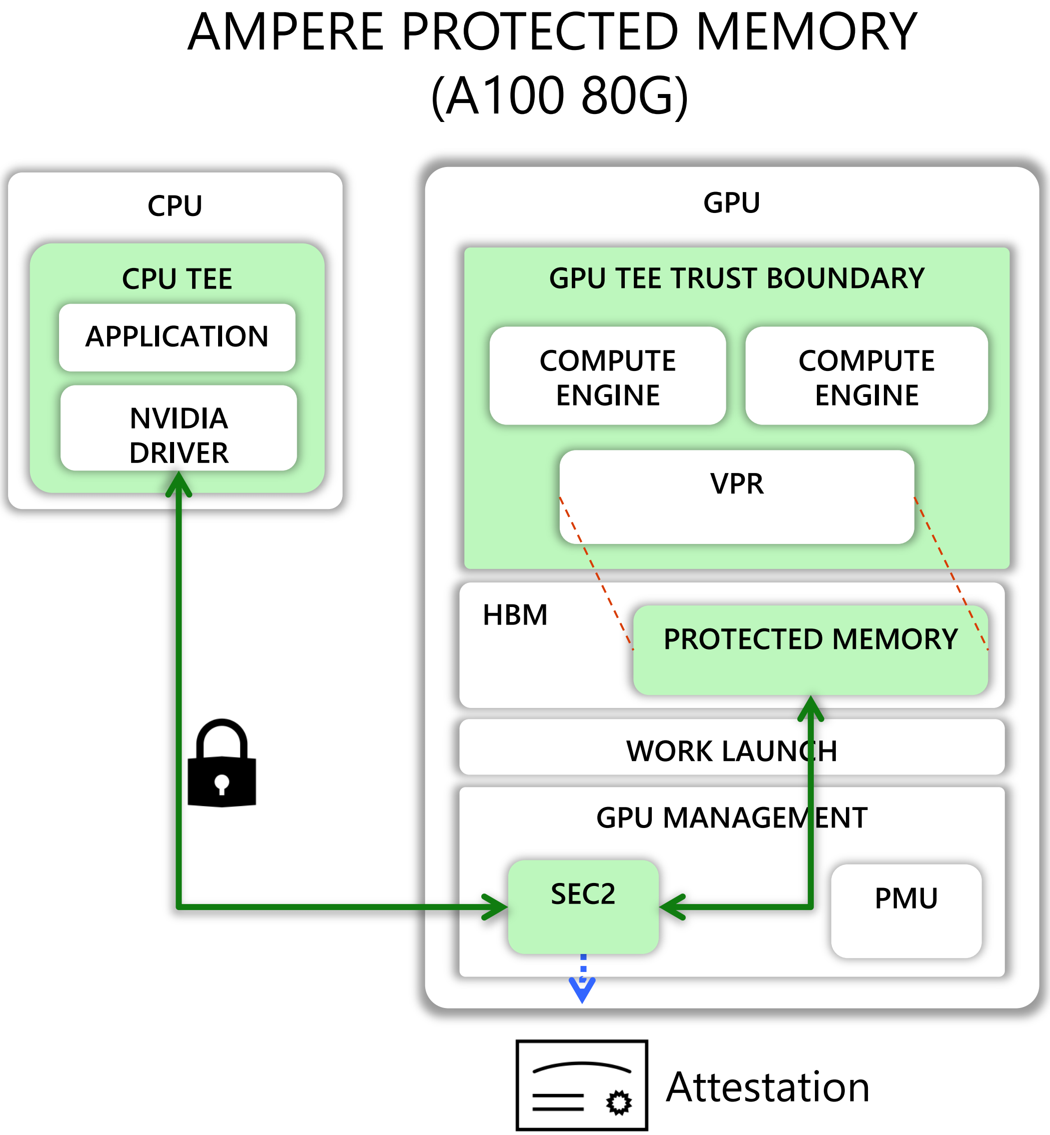
Encrypted Data Transfer

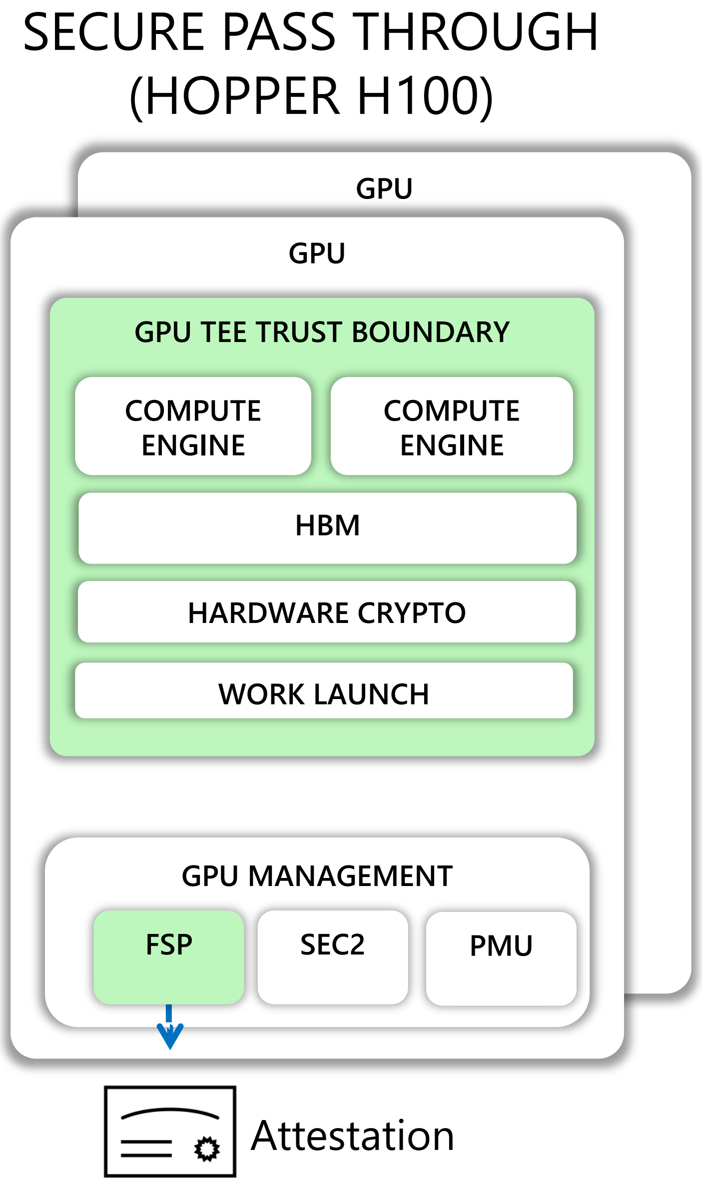


1. Isolate tenant VM
2. Isolate GPU Memory
3. Attest GPU state
4. Create Secure Channel
5. Data transfer encryption

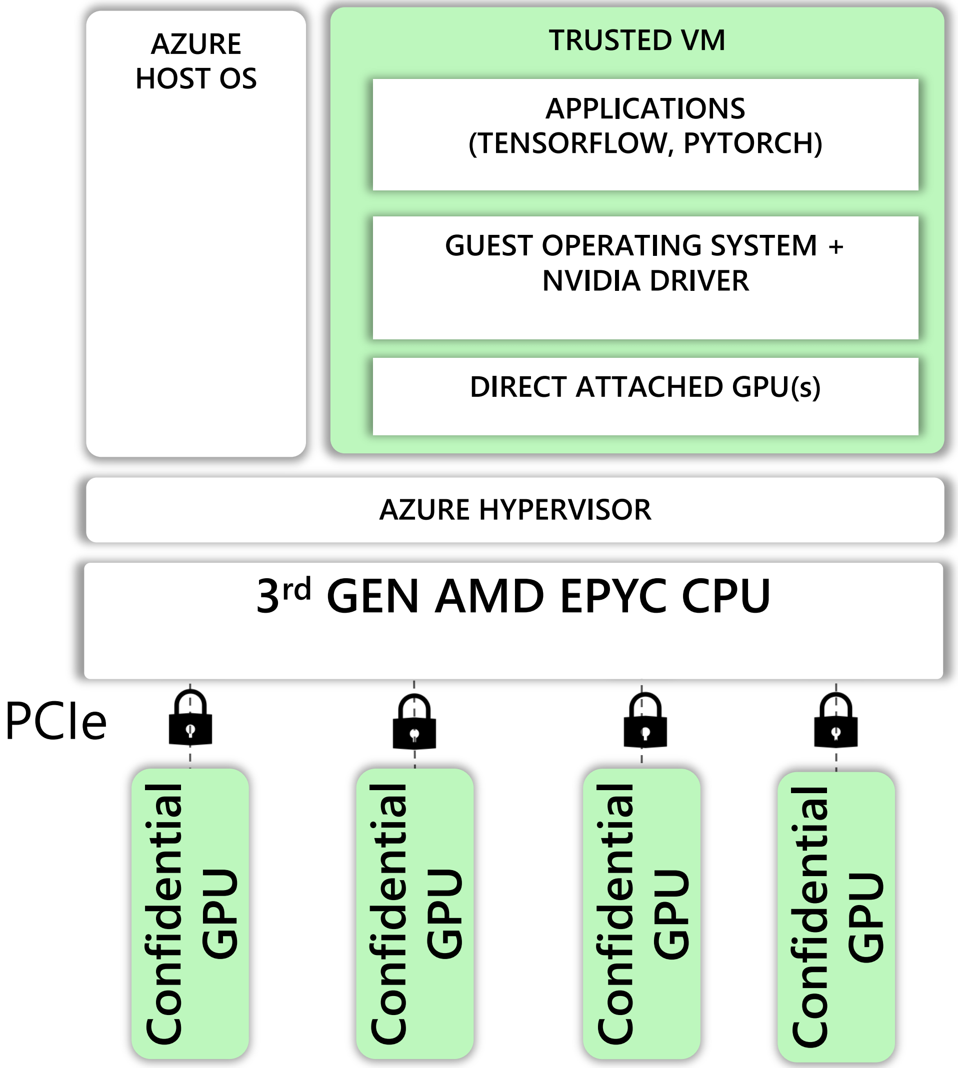
- Memory transfers from VM to GPU are encrypted by the driver and fetched then decrypted by the GPU into HBM memory
- Memory transfers from GPU to VM are encrypted by the GPU and then decrypted by the driver into the VM

NVIDIA's Confidential GPU Roadmap (GTC)





Azure Confidential GPU VM
(NCCv4 series private preview)



Trusted Launch
- Secure Boot guards against rootkits and boot kits
- vTPM based attestation of entire boot chain (UEFI, OS, drivers)

3rd Gen AMD EPYC CPU
- Up to 96 cores and 880 GB memory

4x NVIDIA Ampere A100 GPU
- 80G HBM2 memory
- PCIe Gen4
- Protected memory technology

The Confidential Computing Update Problem

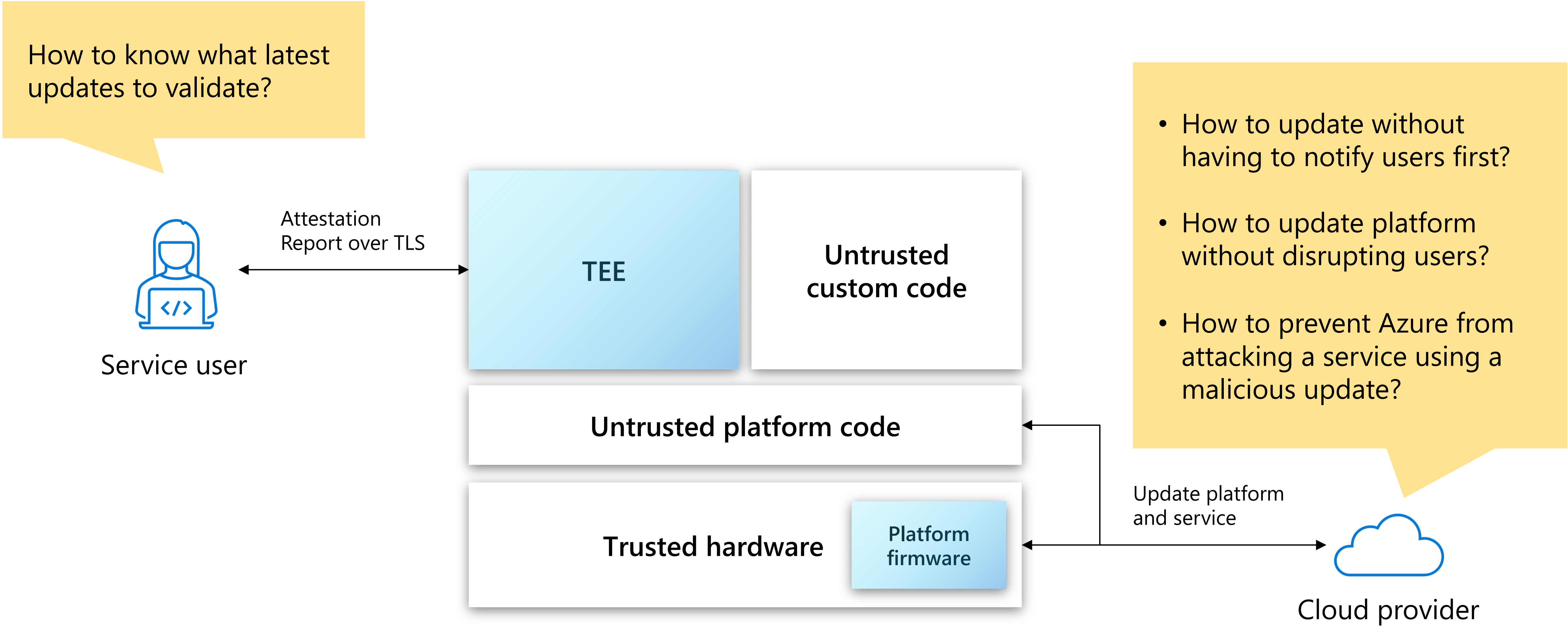
Suppose we have ubiquitous hardware support
for Trusted Execution Environments in the cloud.

How can we update software without breaking remote attestation?

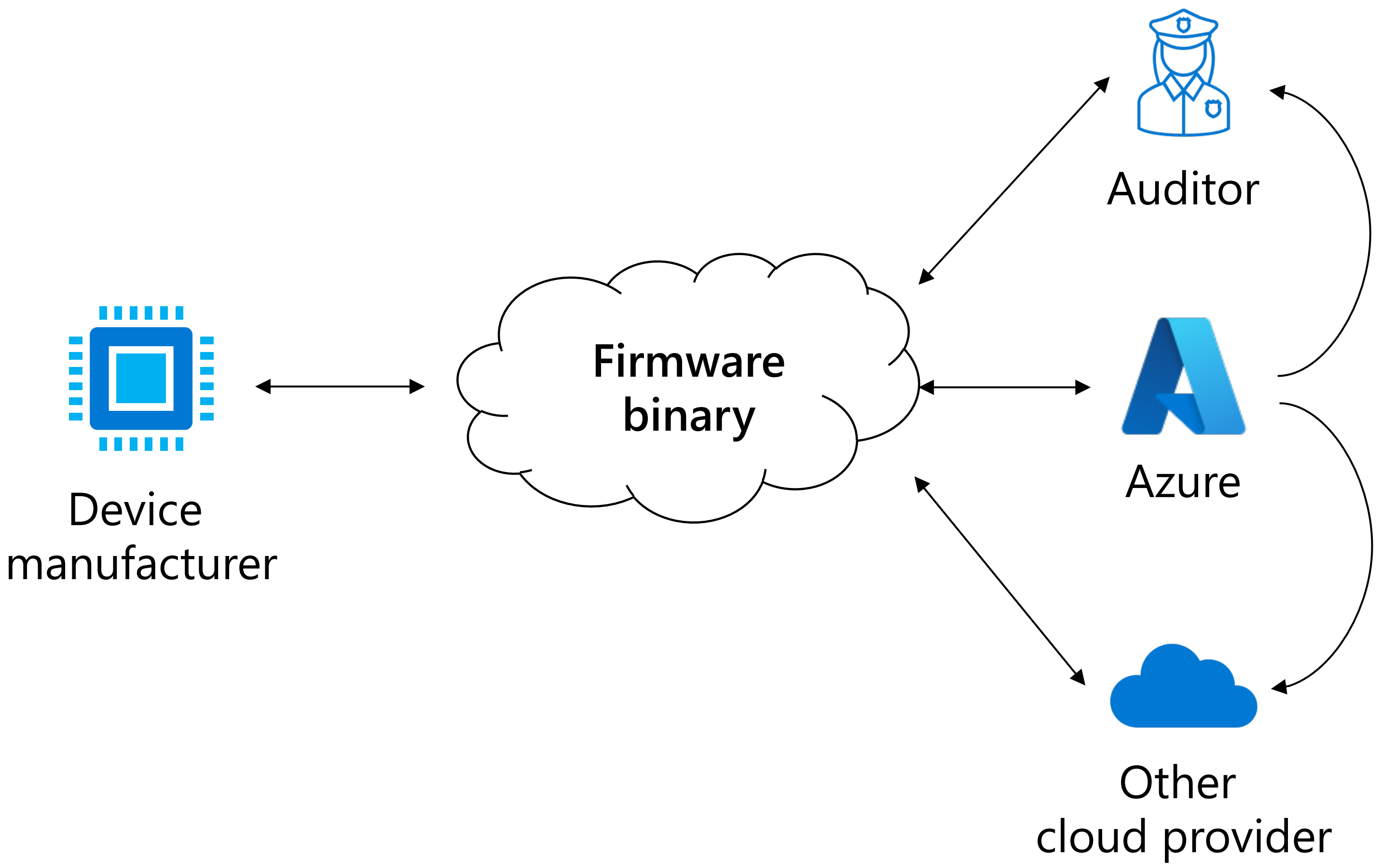
- Firmware, Drivers, Utility VMs, Containers, Framework/Runtime, Crypto libraries, Application code, Data sources/keys, Configuration, Elasticity,…
- Critical vulnerability patches too

Even if all their details are captured in attestation reports,
we can't expect relying parties to review/authorize every update

Challenge: Updatable confidential services



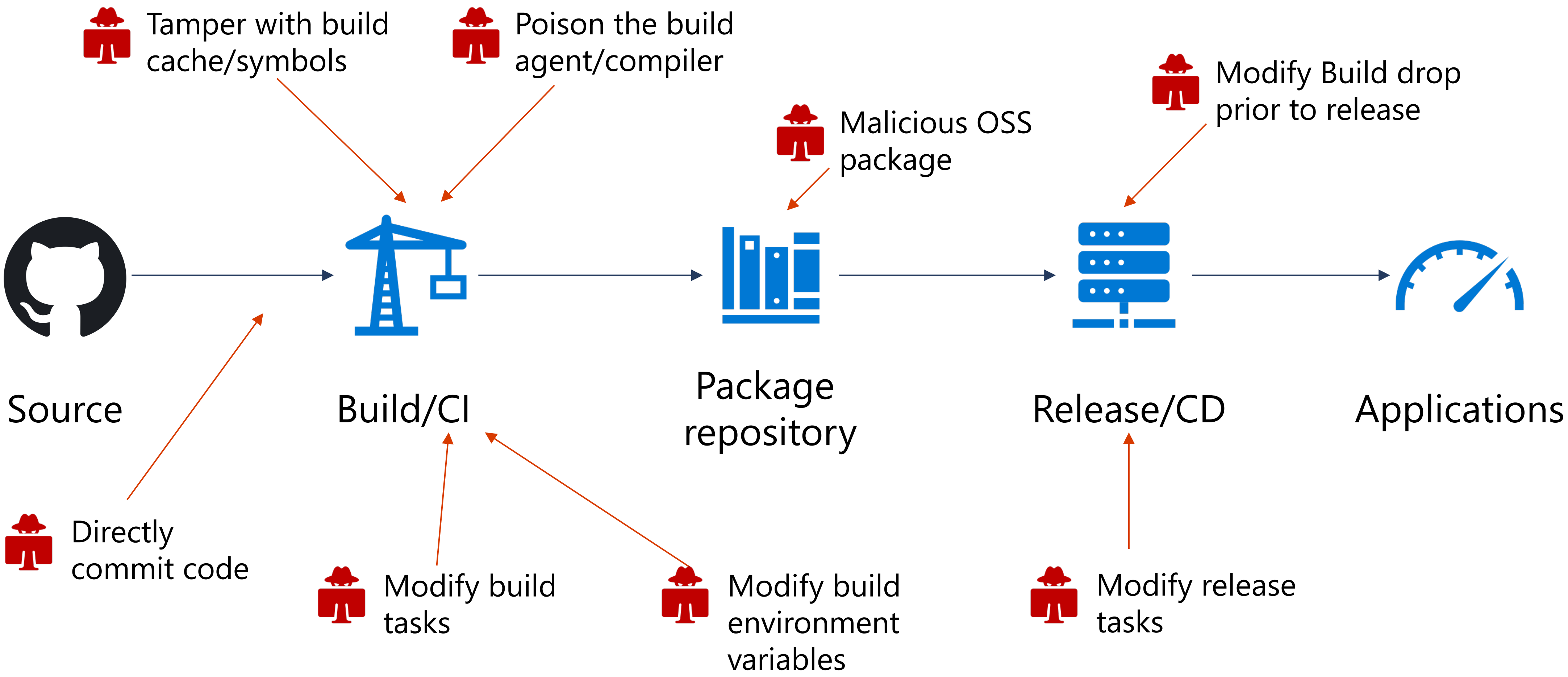
Challenge: Firmware auditability



Can we prove the provenance and audit of firmware deployed on Azure?

Can we share the cost of auditing firmware across cloud service providers?

Challenge: Software supply chain attacks



Transparency: Core Intuitions & Prior Work

We cannot stop supply chain actors from making false claims, but we can make them accountable by requiring all claims be registered in verifiable **transparency ledgers**.

This ensures that malicious actors that make contradictory claims to different entities (customers, auditors, regulators) can be detected and held accountable.

All relying parties must first verify the proof of ledger registration to ensure the claims they use will be auditable—this verification is cheap and can be done offline.

Examples of Transparency Systems

Certificate Transparency [RRC 6962] Adam Langley, Emilia Kasper, Ben Laurie (Google)

CONIKS: bringing key transparency to end users , M. S. Melara, A. Blankstein, J. Bonneau, E. W. Felten, and M. J. Freedman (USENIX Security'15).

Keeping authorities "honest or bust" based on large-scale decentralized witness cosigning (IEEE S&P '16)

CHAINIAC: Proactive Software-Update Transparency via Collectively Signed Skipchains and Verified Builds (Usenix'17, EPFL)

Contour: A practical system for binary transparency logging on bitcoin the latest authorized binary version. M. Al-Bassam, S. Meiklejohn (Data Privacy Management, Cryptocurrencies and Blockchain Technology, 2018).

Types of evidence

Git commit
File hash
…

Source

Packages.json
Package URL
…

Assembly

Makefile
Dockerfile
…

Build

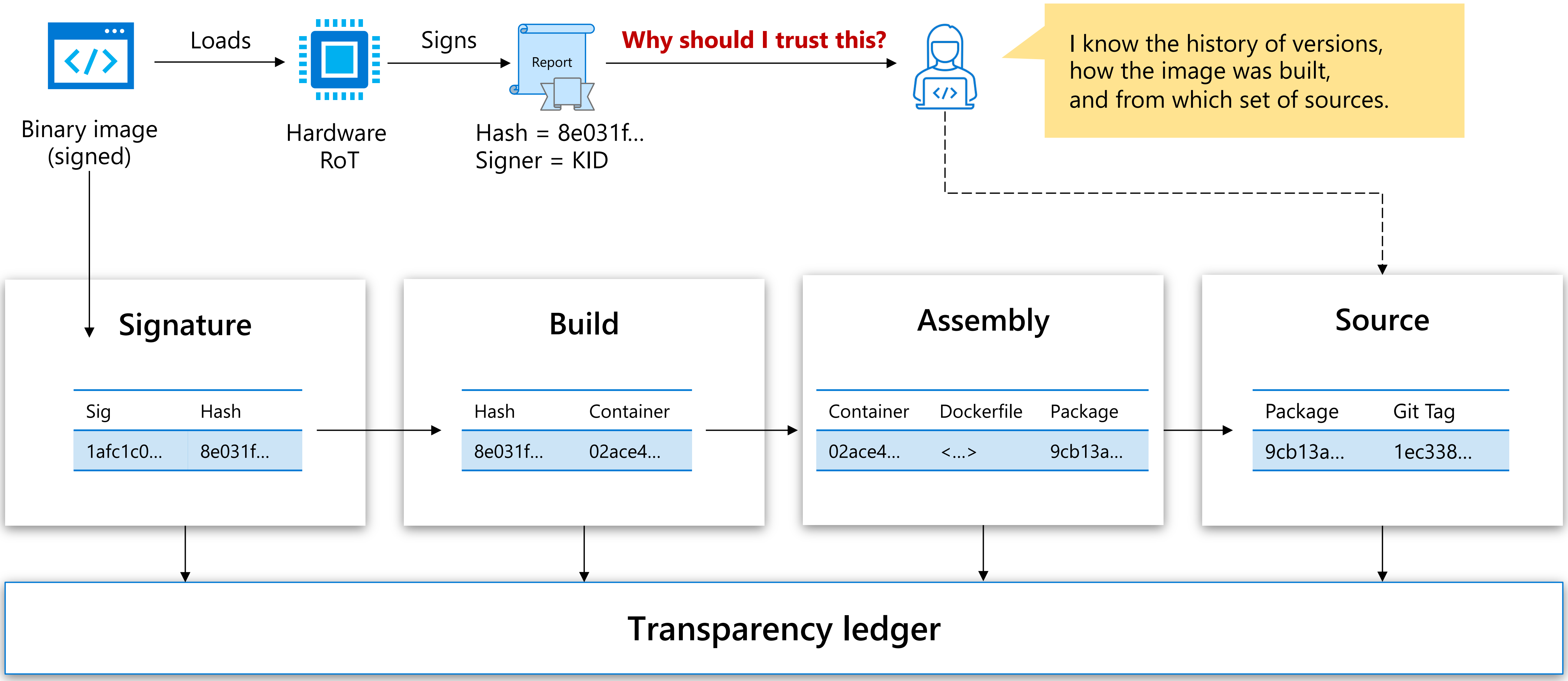
Firmware
Signed apps
…

Signing

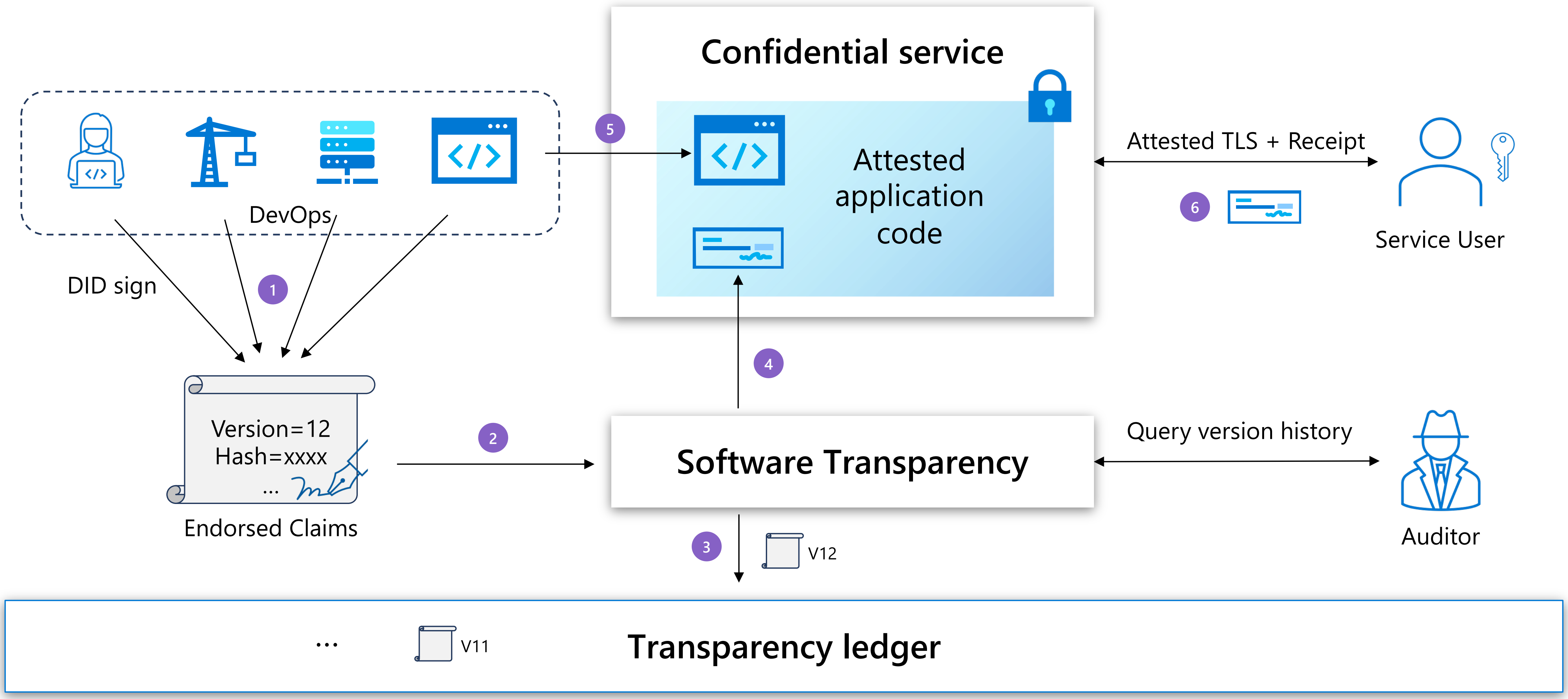
Audits

Endorsements

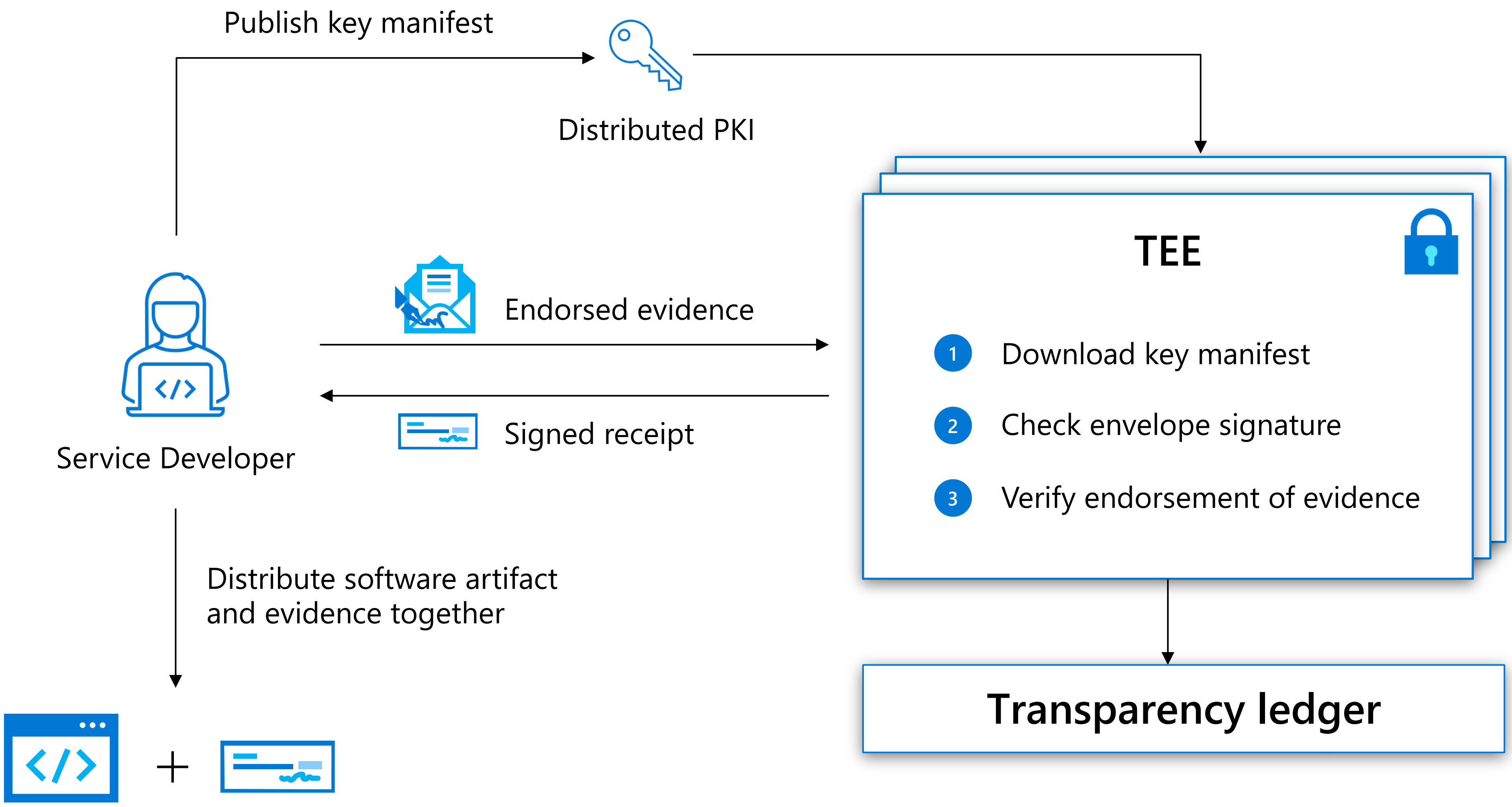
Example: Trusting a hardware measurement



Example: Updatable confidential services

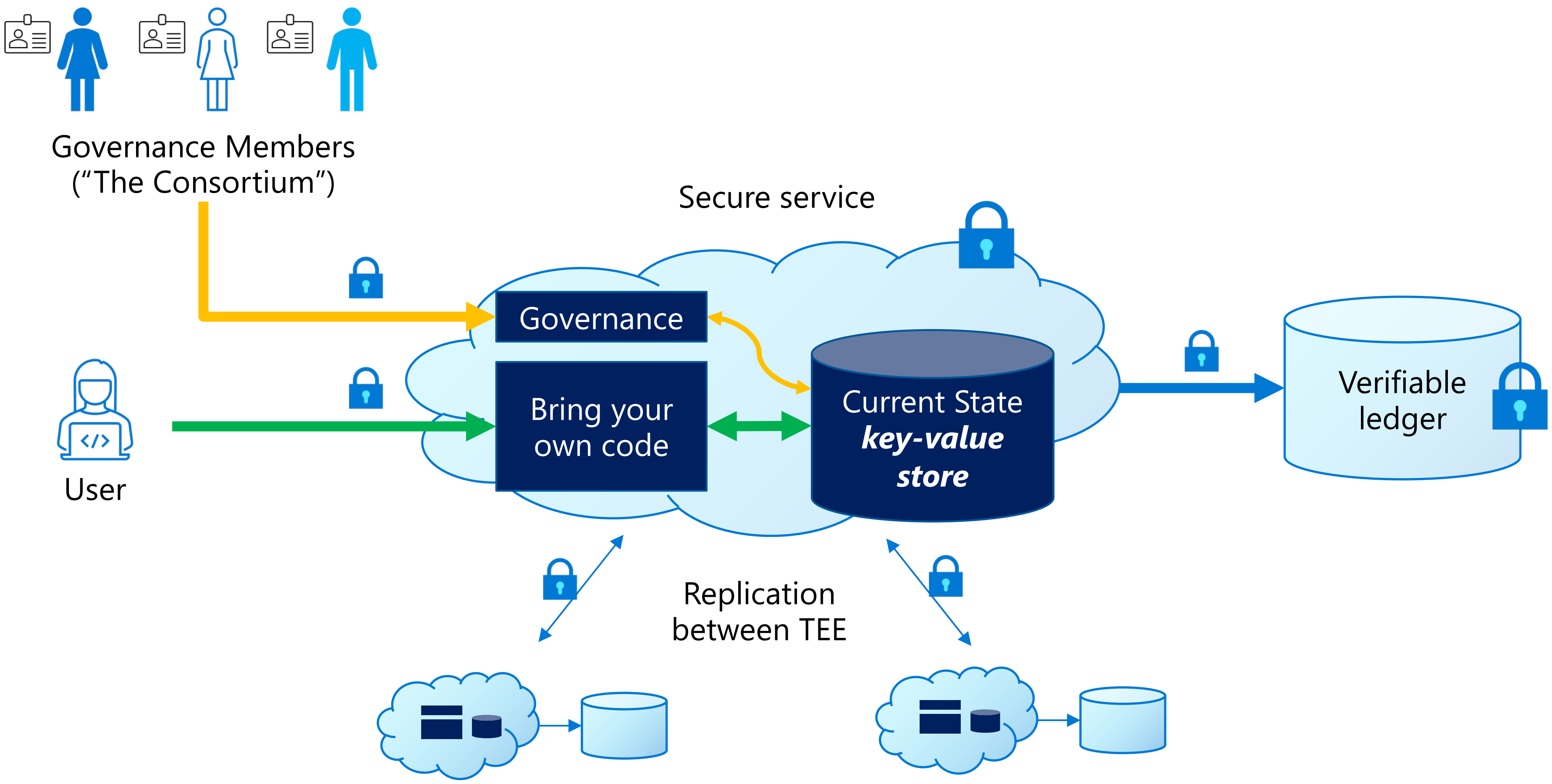


SCITT - Transparency as a Service



CCF - Confidential Consortium Framework

https://github.com/Microsoft/CCF



CCF ledger integrity

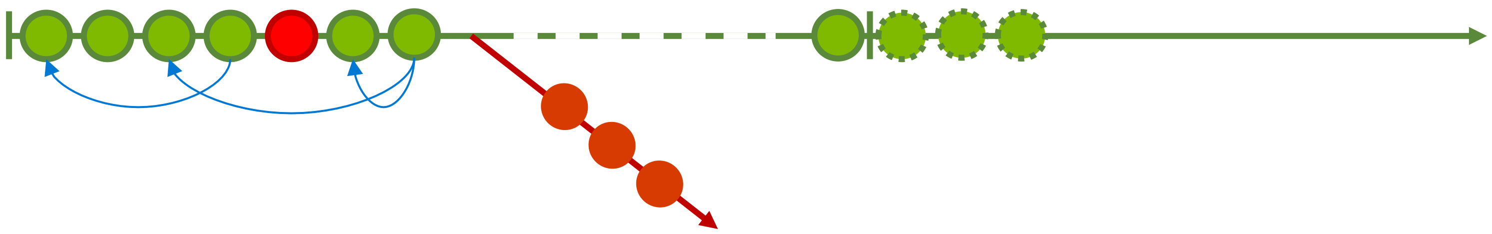
Attacks are **prevented** while a quorum of replicas remain honest contents

Attacks can be **detected** and **blamed** on corrupt replicas based on their signed ledger

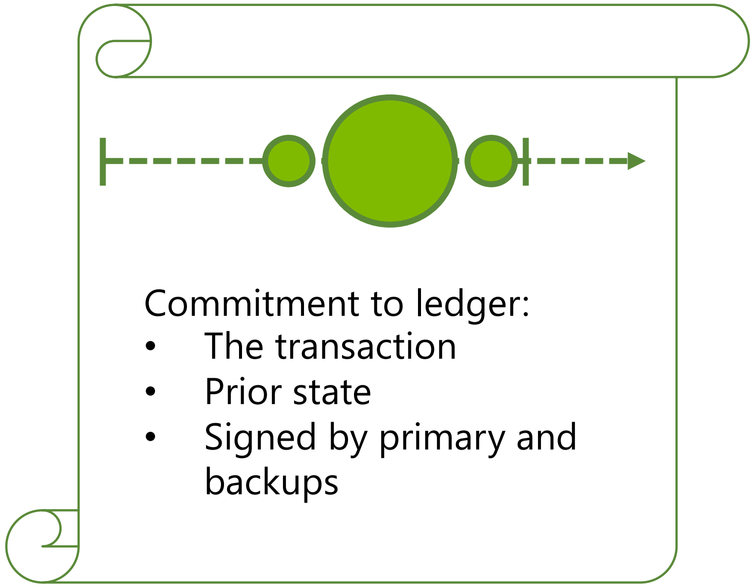
Every transaction is the **correct** replayable result of a command executed on the current state

Tamper proof:
no rollback, change, or reordering of committed transactions

Universally **verifiable receipts** for any committed transaction



The ledger is **consistent**:
no forks on committed transactions



Merkle Trees & Receipts

Receipts

Signing the root of the binary Merkle tree over the whole ledger contents

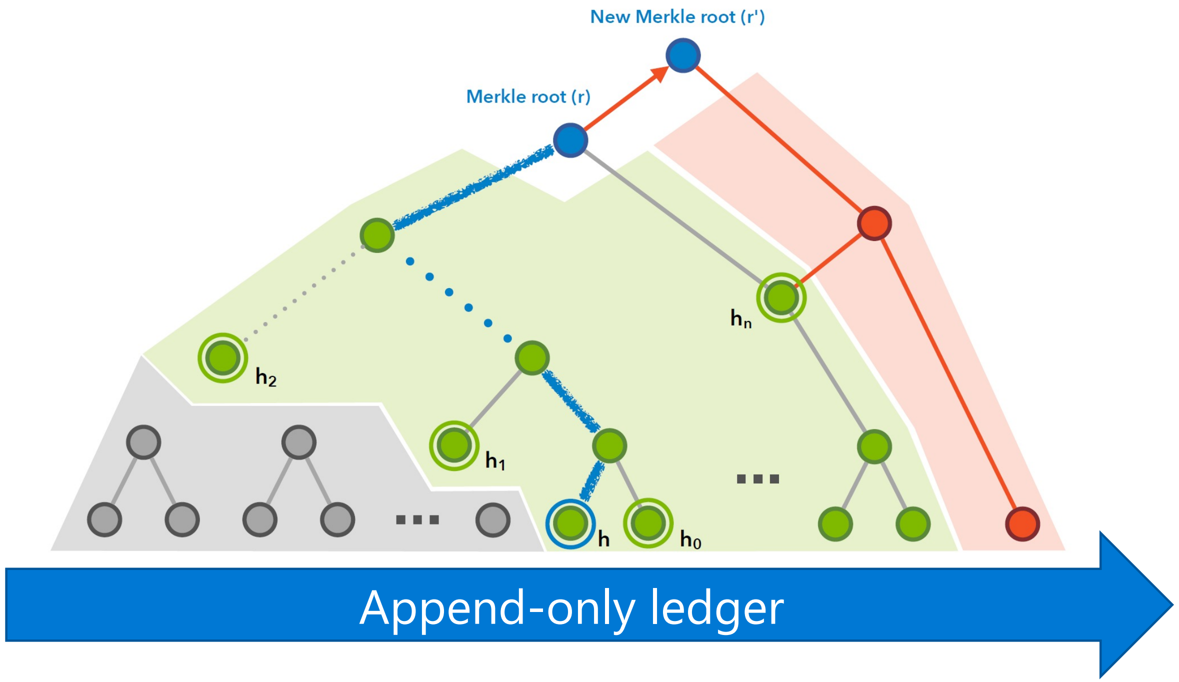
One hash per transaction

One signature per transaction batch

Signing key

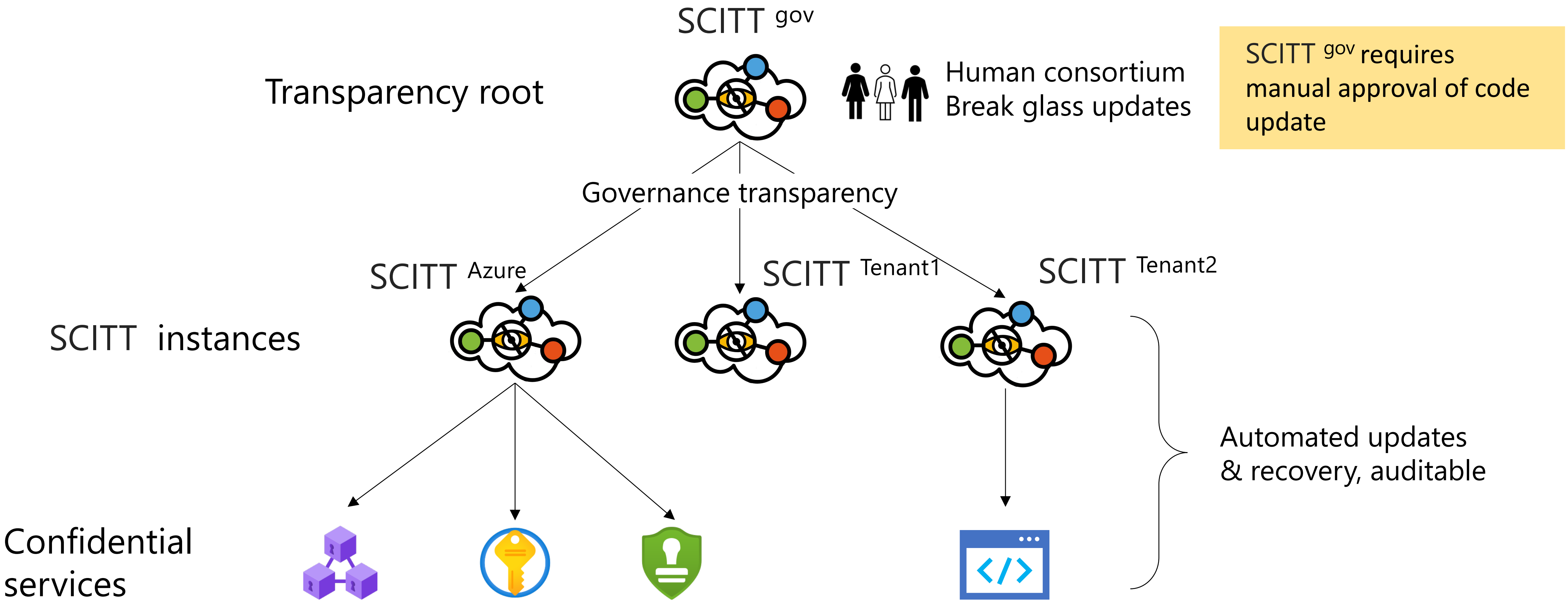
Supported by attestation reports and governance transactions

Recorded in the ledger




```
SCITT_CounterSignature = {
    "serviceId"     => bstr          ; Hash of public key of CCF service
    "transactionId" => tstr          ; CCF transaction id
    "alg" => int                     ; Signature algorithm
    "signature" => bstr              ; Signature over tree root
    "proof" => [+ ProofElement]      ; Intermediate hashes (Merkle path)
}
```

Hierarchical governance & bootstrapping



Confidential Computing in the Cloud

1. Overview
2. Hardware acceleration for ML
3. Software Transparency

Cédric Fournet
Confidential Computing - Microsoft Research