

CoVE Attestation Framework

CCC Attestation Meeting - 2023/05/09

Samuel Ortiz, Ravi Sahita

Goals

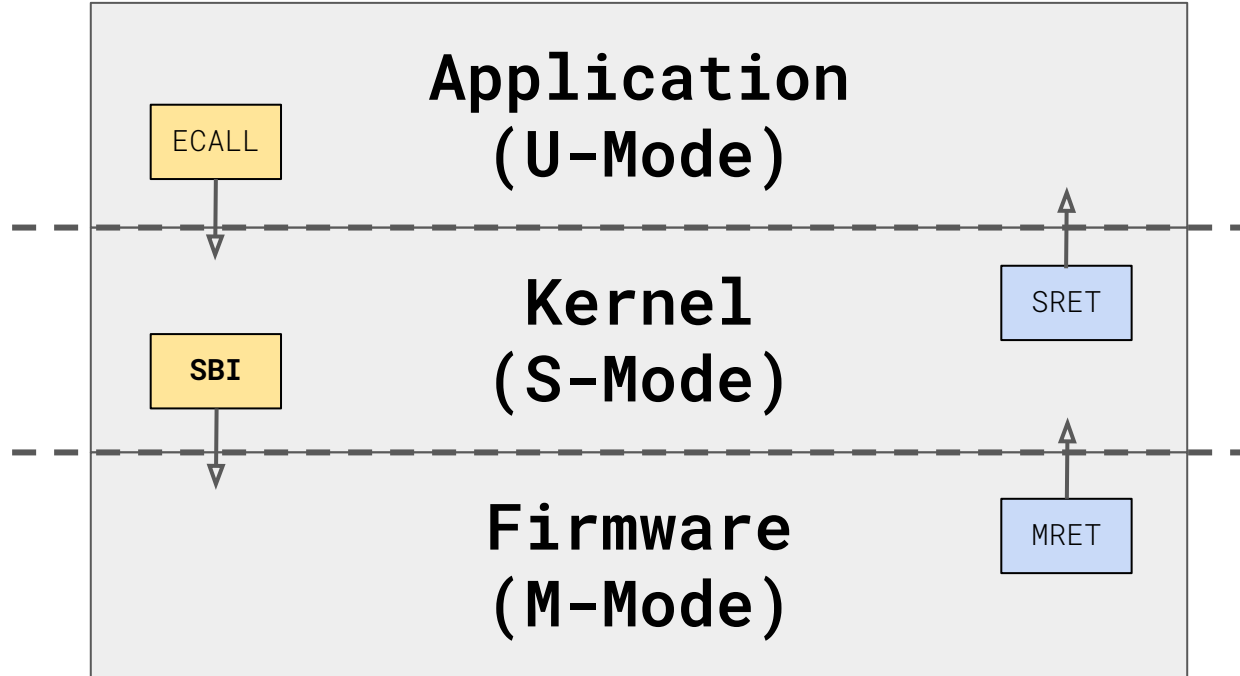
1. **Make the CCC aware of the RISC-V attestation specification ongoing efforts**
2. **Gather feedback from the CCC to improve/enhance the specification**

Agenda

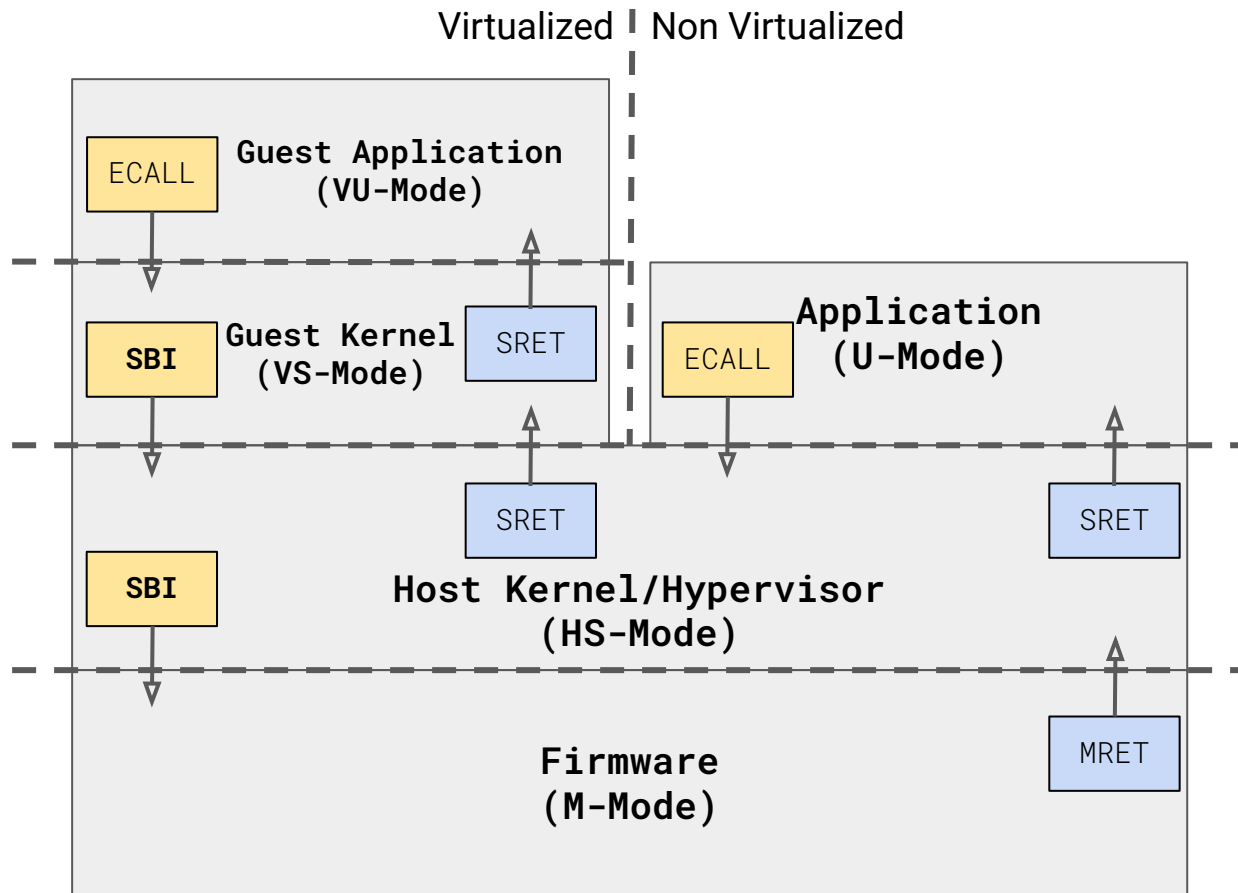
- CoVE (and RISC-V) Refresher
- CoVE Attestation Flows and Formats

RISC-V Primer

- **RISC ISA**
- **Free and Open**
- **Stable but modular ISA**
 - Base ISA and standard extensions are frozen
 - Extensions are optional
 - Hardware virtualization is an extension (H-Extension)
- **Privilege Modes**
 - $M > U > S$
 - M only - Basic embedded systems
 - M + U - Enhanced embedded systems
 - M + S + U - Rich OS (e.g. Linux), Applications
 - M + HS + U - Hypervisor, with Rich OS guests (e.g. Linux), Applications



RISC-V Privilege Modes

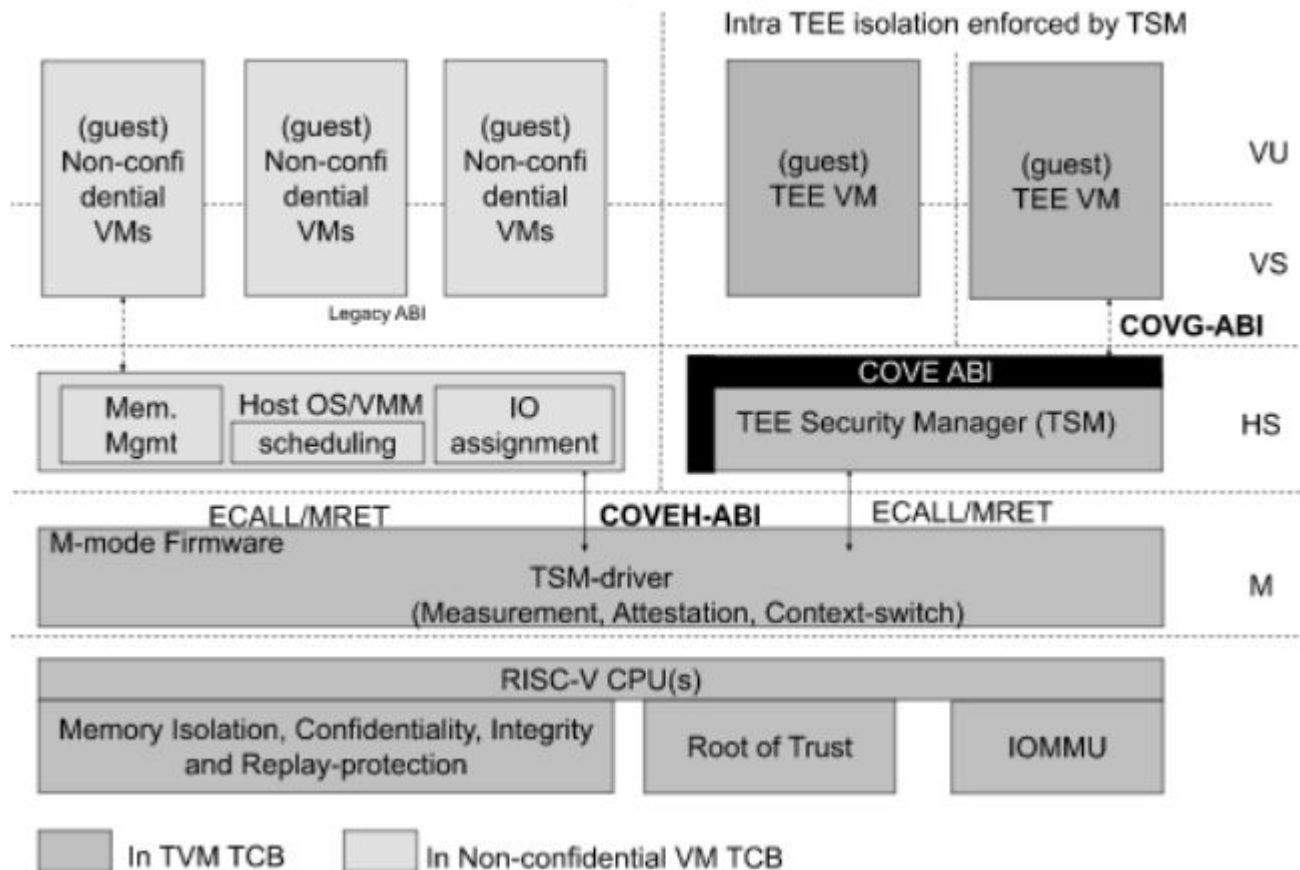


RISC-V Privilege Modes with **Hypervisor Extension**

CoVE

- **Previously known as AP-TEE**
 - Application Processor Trusted Execution Environment
 - Target use case - Confidential Computing for application class RISC-V-based platforms
- **Now known as - Confidential VM Extension**
 - TEE workloads are typically VM guests
 - Similar goals as other Confidential VM e.g. Intel TDX, AMD SEV-ES-SNP or ARM CCA
- **Main Components**
 - TVM - TEE VM
 - TSM - TEE Security Manager, a *TCB* intermediary between the TVMs and the host
 - TSM Driver - TEE Security Manager Driver, a *TCB* M-mode component hosting the TSM ABI
 - Host VMM/Hypervisor - *Untrusted* Virtual Machine Monitor

TEE/non-TEE isolation provided by CPU e.g. MTT

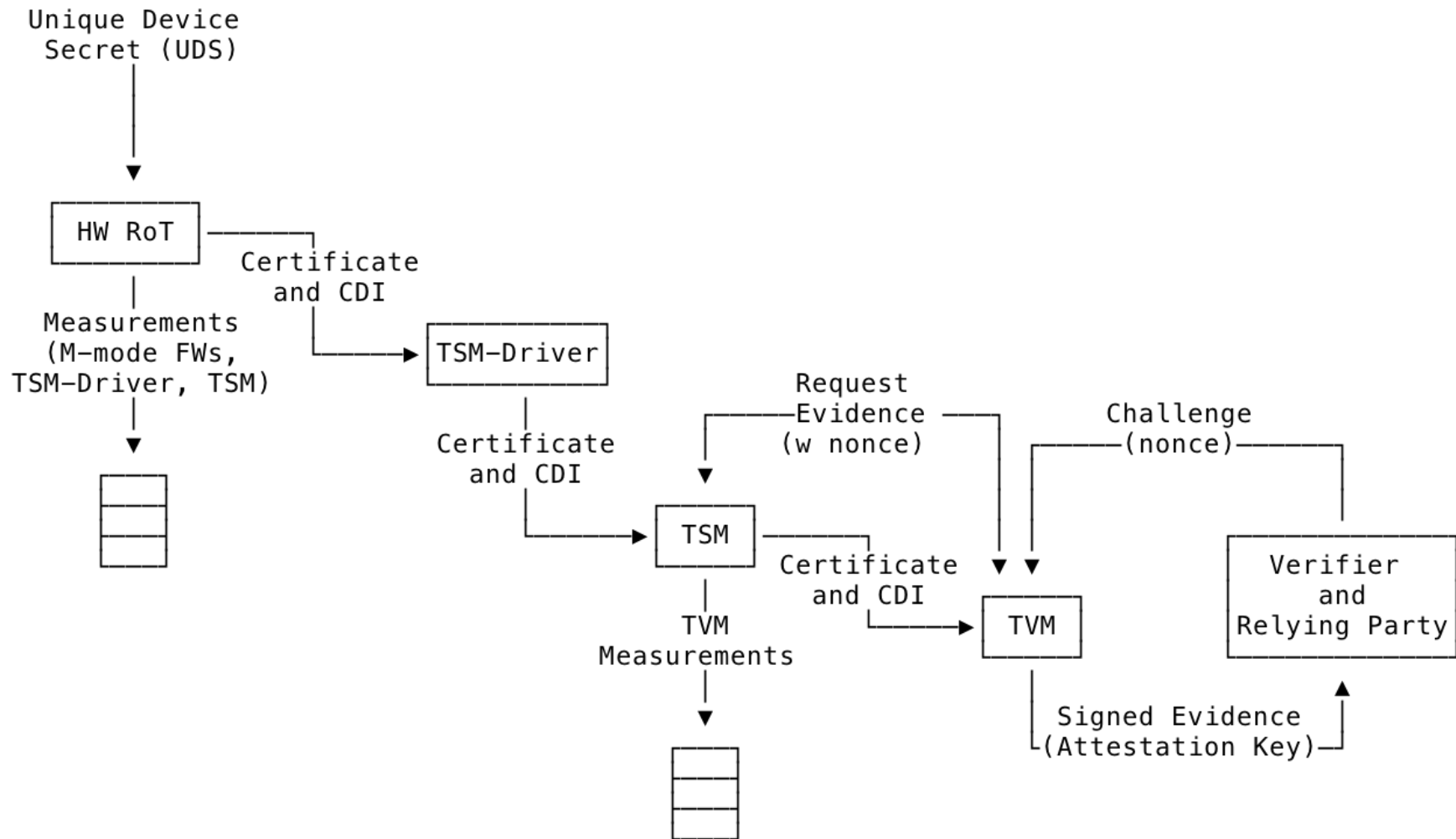


CoVE Attestation

- DICE layered attestation
- EAT-formatted Attestation Evidence

CoVE DICE

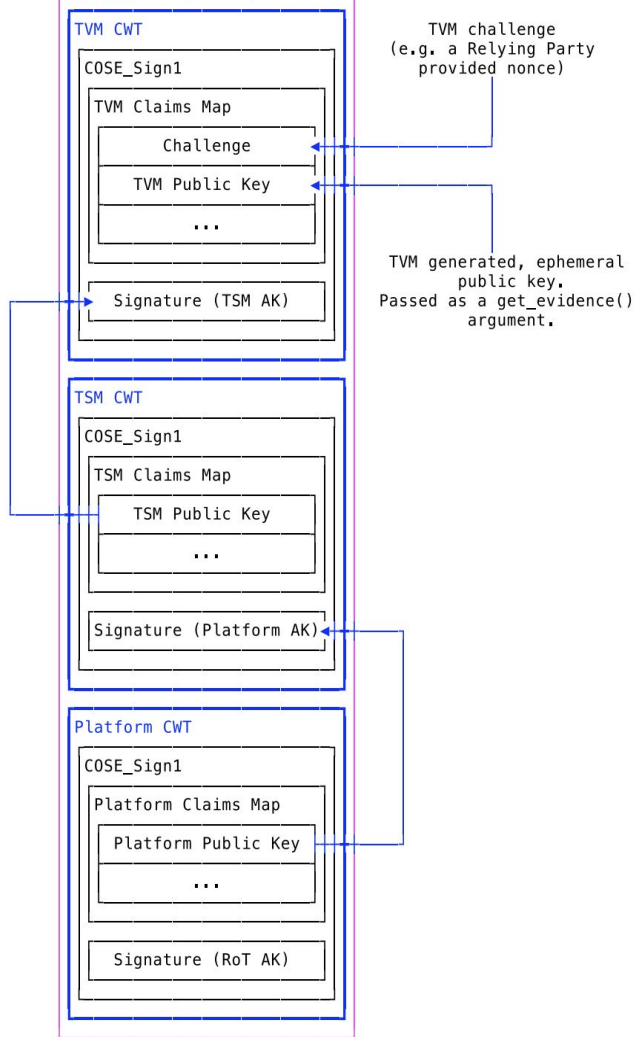
- **A CoVE workload TCB is composed of 3 independent TCBs**
 - a. Platform TCB
 - All platform components participating to the CoVE workload TCB
 - RoT, CPU, all SoC subsystems (Memory, PCIe, PM, etc controllers, IOMMU, etc)
 - All M-mode firmwares, all SoC components firmwares
 - b. TSM TCB
 - TSM and TSM Driver
 - c. TVM TCB
 - TVM measured pages
 - [Assigned TEE-IO devices]
- **DICE starts at platform ROM**
- **The platform RoT derives the platform TCB**
 - a. Provides CDI and Certificate to the TSM Driver
- **TSM generates the final Attestation Evidence**



CoVE Attestation Evidence

- **Evidence payload is a UCCS (Unprotected CWT Claim Set)**
- **One EAT Submodule Claim Set**
 - Each map value is an attestation token
 - One token per TCB component (Platform, TSM, TVM)
- **The payload is embedded into either an CBOR or X.509 certificate**

Submodule Claims Map



CoVE EAT profile claim

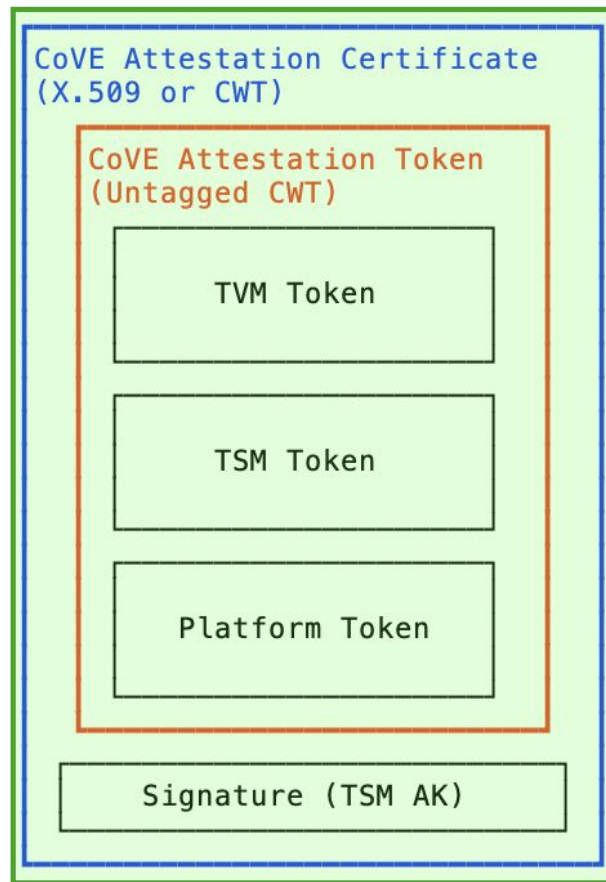
The platform EAT profile claim describes the EAT profile that the CoVE platform implements. The profile should include a description of all three tokens (platform, TSM and TVM) as they are bound together.

EAT Profile Claim

```
riscv-cove-eat-profile-label = 265 ; EAT profile  
riscv-cove-eat-profile-doc = "https://riscv.org/TBD"  
  
riscv-cove-eat-profile = (  
    riscv-cove-eat-profile-label => riscv-cove-eat-profile-doc  
)
```

Evidence Generation

- Defined through the CoVE Attestation ABI
- TVM requests the evidence to TSM
- **sbi_covg_get_evidence()**
 - Inputs
 - TVM public key
 - Nonce,
 - Certificate format (CBOR or X.509)
 - Outputs
 - Attestation Certificate
 - TSM signed



References

[CoVE specification](#) (See pdf in the repo for ease of reference)

[CoVE attestation specification](#)

[Linux CoVE RFC](#)

[RISC-V TSM github](#)

[RISC-V ISA specification](#)