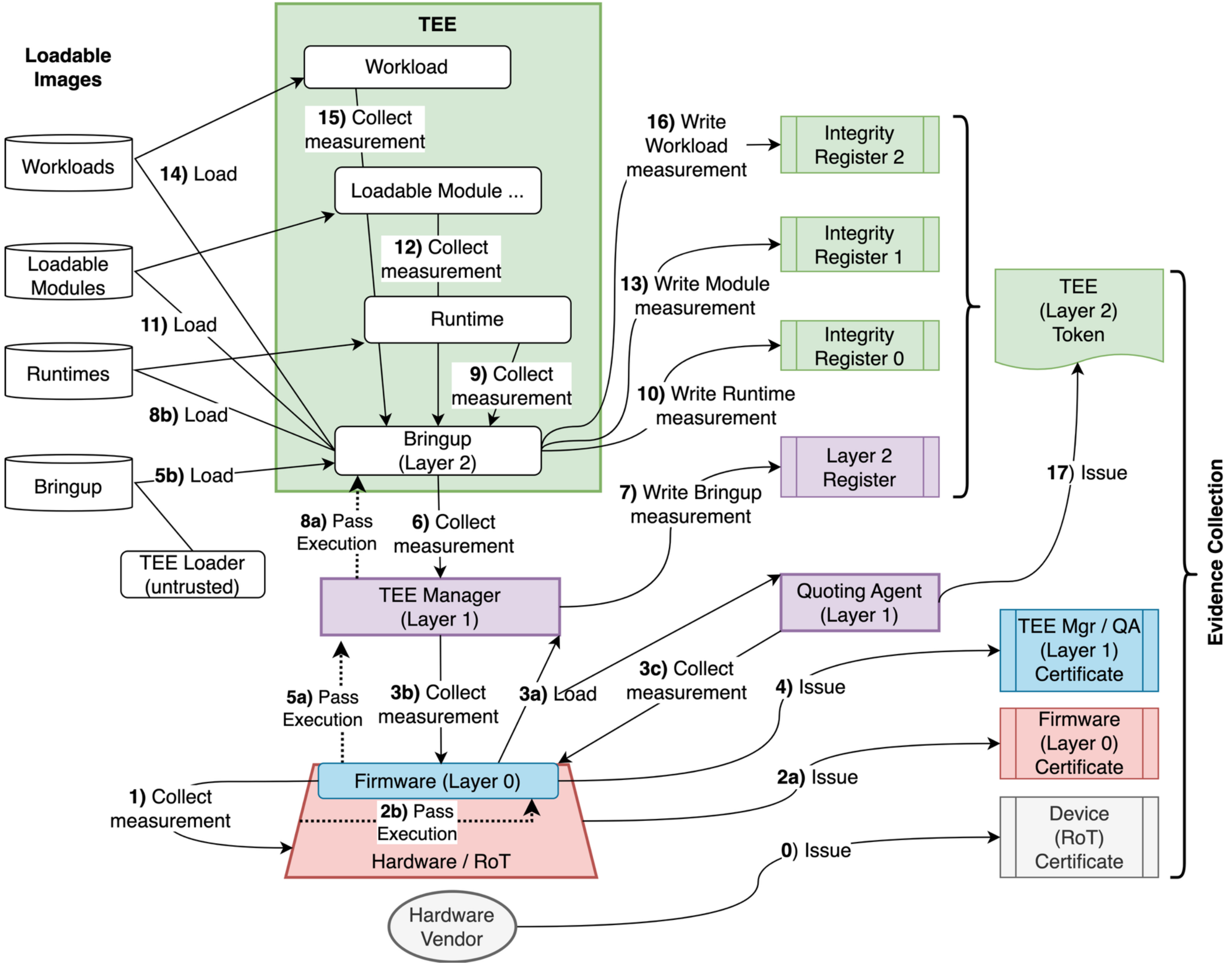
CoRIM Domain Dependency Triple

Ned Smith

February 4, 2026

Example TEE Layering Use Case

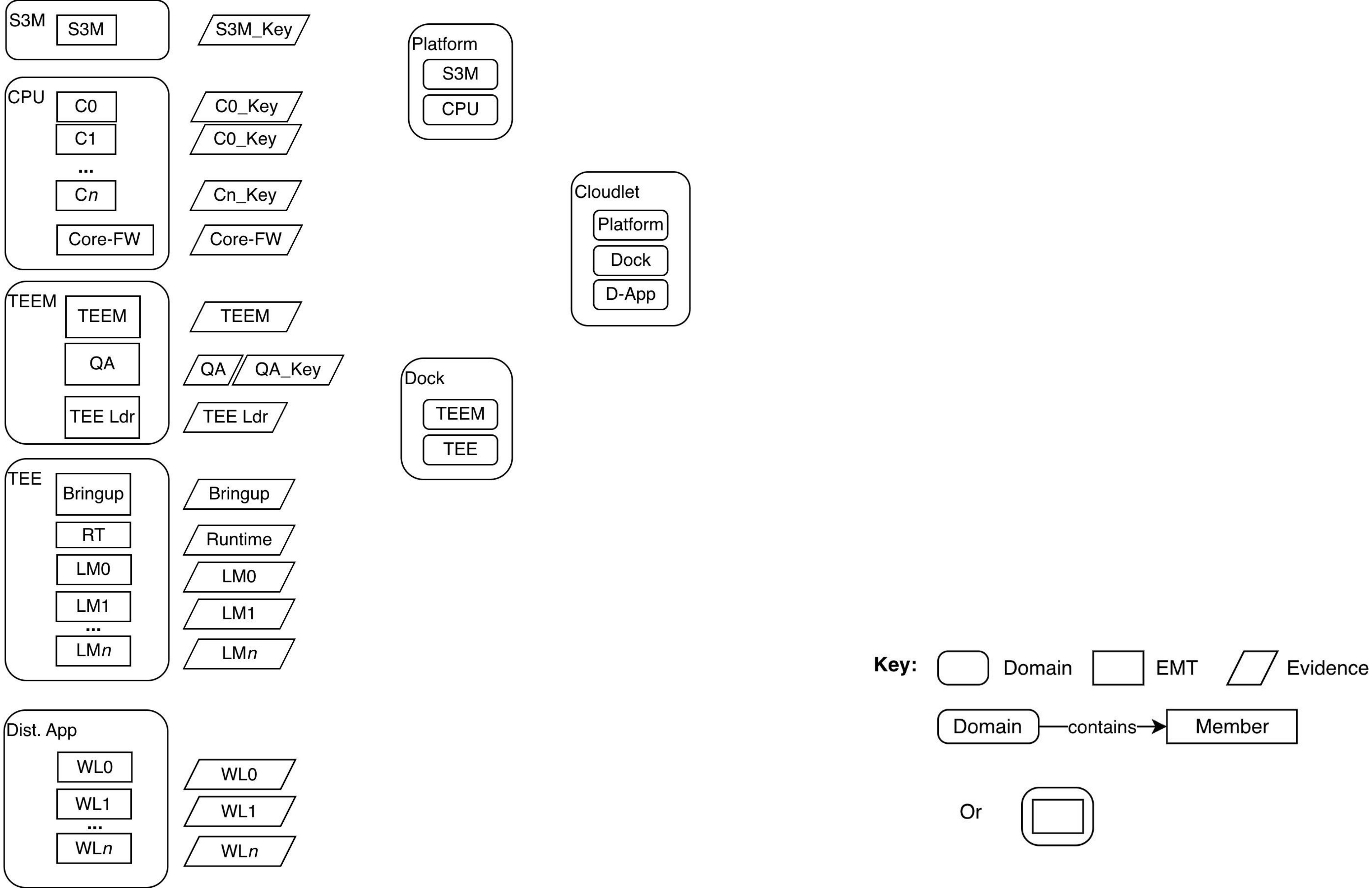


Use Case Summary

- There are many ecosystem entities
  - Req: multiple domain triples likely needed to capture domain dependencies
- Example TEE is a hybrid of DICE layering and dynamic composition
  - Req: multiple membership triples likely needed to capture membership
- Hardware can be partitioned (multi-core) and each core can have its own keys
  - Req: keys can be reported as evidence (i.e., env-meas tuple - EMT)

Membership Triples Design

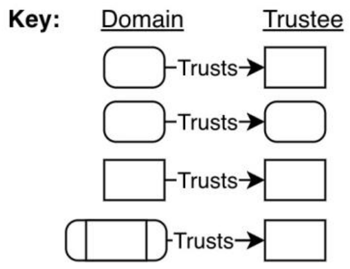
- Simplifying assumptions
  - Membership triple design roughly follows color coding (slide 2)
  - Different vendors likely supply differently colored boxes
  - Additionally, workloads are likely supplied by yet other vendors

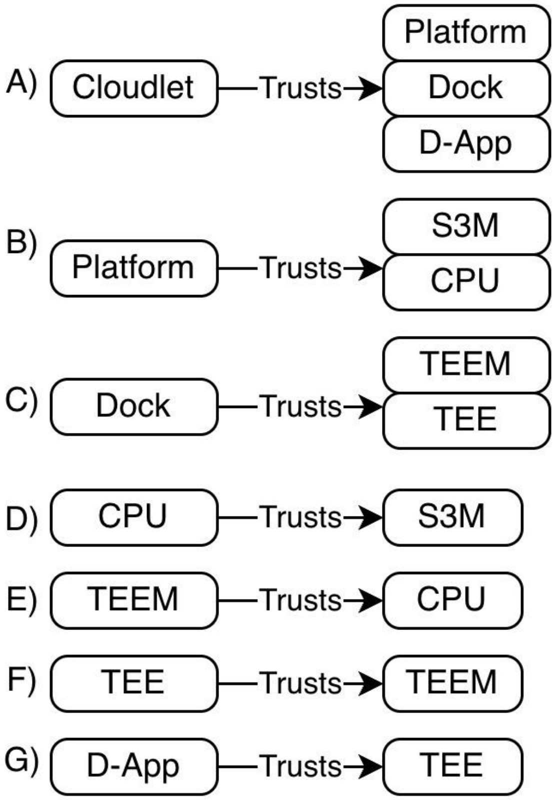

S3M
S3M
S3M_Key
CPU
C0
C0_Key
C1
C0_Key
...
C*n*
Cn_Key
Core-FW
Core-FW
TEEM
TEEM
TEEM
QA
QA
QA_Key
TEE Ldr
TEE Ldr
TEE
Bringup
Bringup
RT
Runtime
LM0
LM0
LM1
LM1
...
LM*n*
LM*n*
Dist. App
WL0
WL0
WL1
WL1
...
WL*n*
WL*n*
Platform
S3M
CPU
Cloudlet
Platform
Dock
D-App
Dock
TEEM
TEE
Key:
Domain
EMT
Evidence
Domain
contains
Member
Or

Domain Dependency Triple Design

- Goals
  - Establish membership before seeking trust dependencies
  - Describe the path through a DMT graph that defines trust dependency
  - Disallow cycles – trust terminates at Roots of Trust, peers are uninteresting
  - Support multi-vendor ecosystem – multiple CoRIM authors

Domain to Domain Dependency Triples Design





: Because Platform, Dock and D-App are contained in Cloudlet

: Because S3M and CPU are contained in Cloudlet

: Because TEEM and TEE are contained in Dock

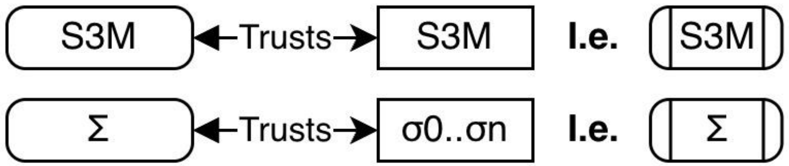
: Because S3M initializes CPU

: Because TEEM calls into CPU

: Because TEE calls into TEEM

: Because D-App calls into / is hosted by TEE

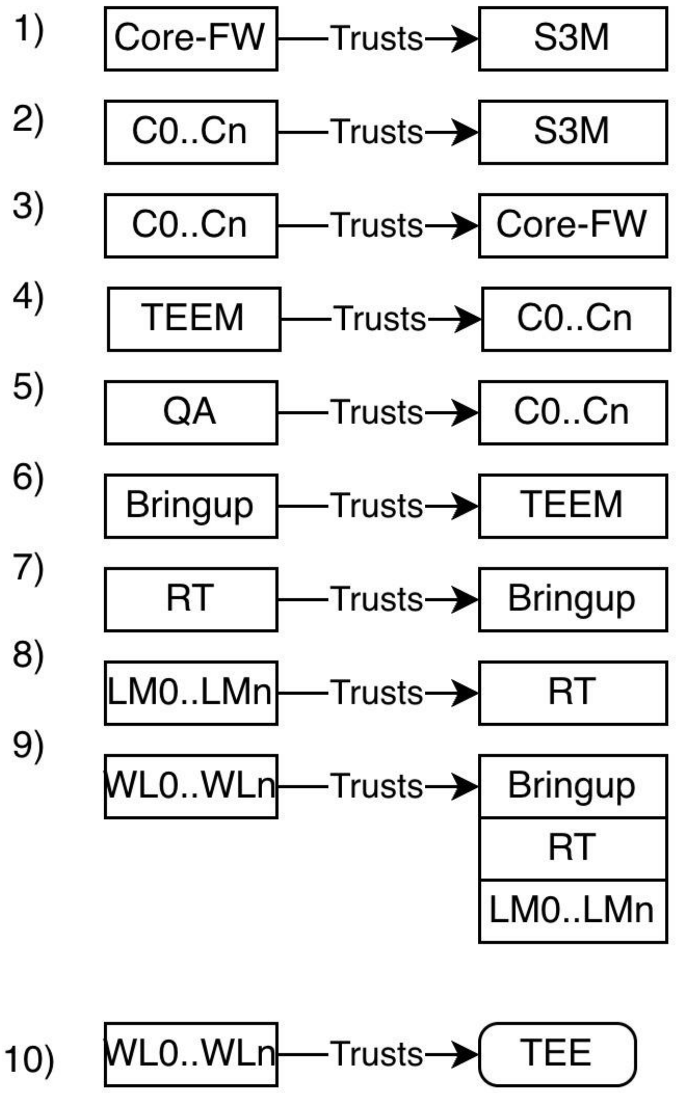
Implied Trust Cases



: Because S3M domain trusts S3M environment and vise versa

: This is generalizable

Environment to Environment Dependency Design



1) Core-FW —Trusts→ S3M : Because S3M loaded Core-FW

2) C0..Cn —Trusts→ S3M : Because S3M initialized C0..Cn

3) C0..Cn —Trusts→ Core-FW : Because S3M loaded Core-FW on C0..Cn

4) TEEM —Trusts→ C0..Cn : Because TEEM calls into C0..Cn

5) QA —Trusts→ C0..Cn : Because QA calls into C0..Cn

6) Bringup —Trusts→ TEEM : Because Bringup calls into on TEEM

7) RT —Trusts→ Bringup : Because RT is loaded by Bringup

8) LM0..LMn —Trusts→ RT : Because LM0..LMn calls into RT

9) WL0..WLn —Trusts→ Bringup
RT
LM0..LMn : Because WL0..WLn calls into LM0..LMn

10) WL0..WLn —Trusts→ TEE : Because WL0..WLn calls into LM0..LMn and domain TEE is eqivalent to the sum of its environments (see previous relation) and reaffirms above relation (D-App trusts TEE)