Caliptra: DICE-as-a-Service

Providing a useful cryptographic identity to SoCs

Jeff Andersen (Google)
Problem statement

• Host software wants to wield a cryptographic identity bound to its boot state
  - Use-case: host software wants to be an SPDM Requester and authenticate itself to a Responder
  - Use-case: host software wants to bind TPM secrets using TPM2_PolicySigned

• Host software wants Caliptra to be agnostic as to how that identity is used
  - The high-level flows can evolve independently from Caliptra firmware
  - No need for Caliptra to act as an SPDM Requester or TPM client
Solution: DICE-as-a-Service

- Caliptra holds DICE keys, and wields them on behalf of the host

- Host can extend additional measurements, which get factored into the DICE key derivation
DICE cert hierarchy

Caliptra's own DICE stack

Manufacturer CA cert
Manufacturer CA pub key

Caliptra IDevID cert
(Device serial number)
IDevID pub key

Caliptra alias cert
(TCI = Caliptra fw measurements)
Alias pub key

Host DICE cert (example)
[TCI0 = Vendor {FMC + uCode + …} boot measurements]
[TCI1 = Host IBB boot measurements]
[TCI2 = runtime measurements]
Host pub key

Host pub key
DICE plus SPDM

- SPDM exposed to internal+external callers
- DICE-as-a-Service exposed to internal-only callers
- Measurements can be reported via either channel