Measurement and Attestation RootS (MARS)

Global Platform – Cybersecurity Vehicle Forum Virtual Update

Sep 25, 2023

Tom Broström

Cyber Pack Ventures, Inc.

Who am I?

- Previously
  - Technical Director at NSA (33 years)
  - Adjunct Professor, NSA Cryptologic School
  - Adjunct Professor, University of Maryland Baltimore County
- Currently
  - Technical Director at Cyber Pack Ventures, Inc.
  - Contracting to NSA's Laboratory for Advanced Cybersecurity Research, Trust Mechanisms Office
  - Chair of TCG MARS Work Group

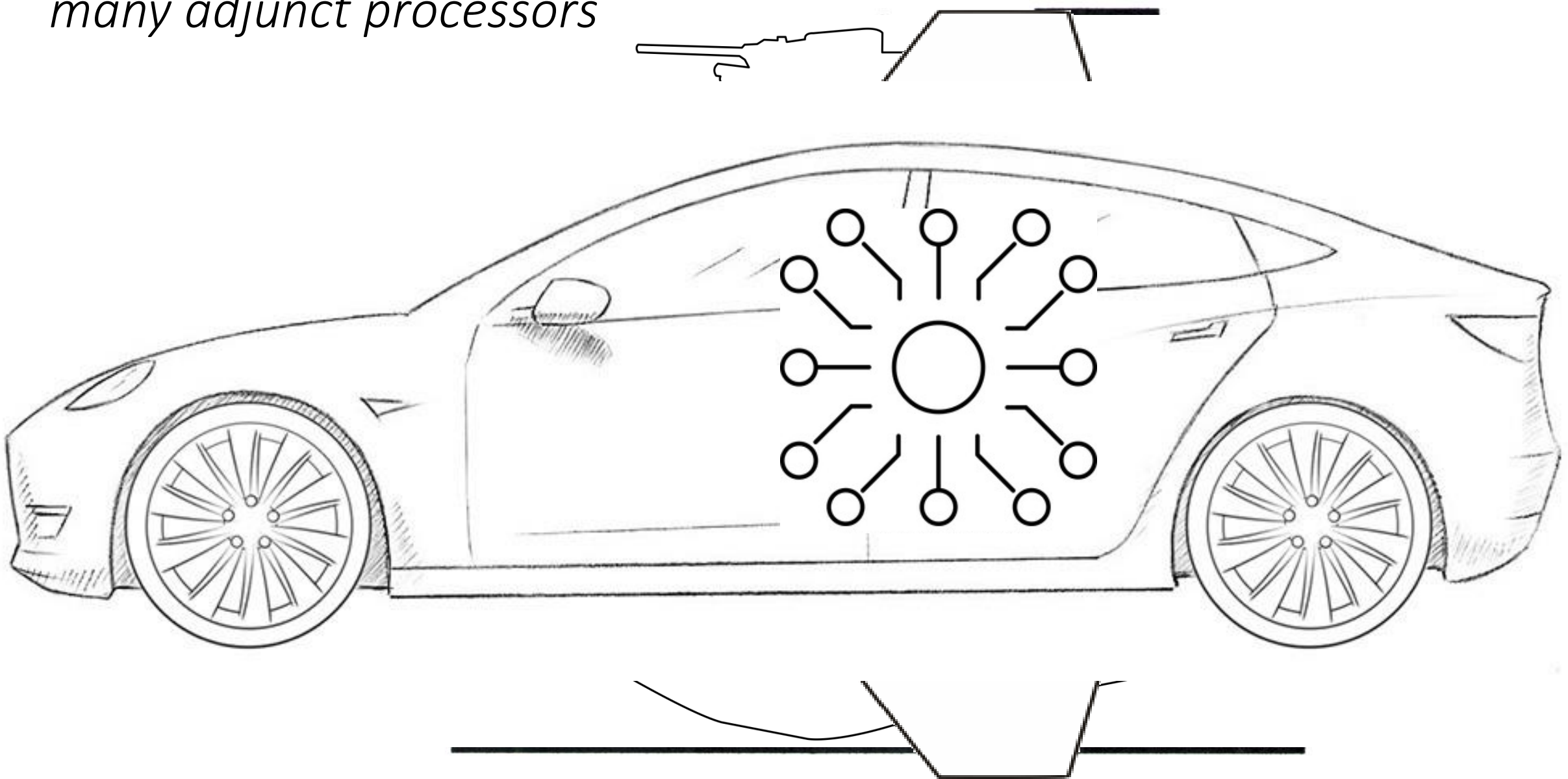

IDENTITY

Identity

- A handle by which a thing is definitely recognizable or known
- E.g., VIN, SSN, fingerprint, public key, digest
- Important “things”
  - Embedded device, IoT, adjunct processing element
  - Configuration
  - Firmware
- Evidence
  - identities of and in a device
  - Collected via *measured boot*
  - Reported via *attestation*
  - Facilitated by Root of Trust – Measurement and Attestation RootS (MARS)

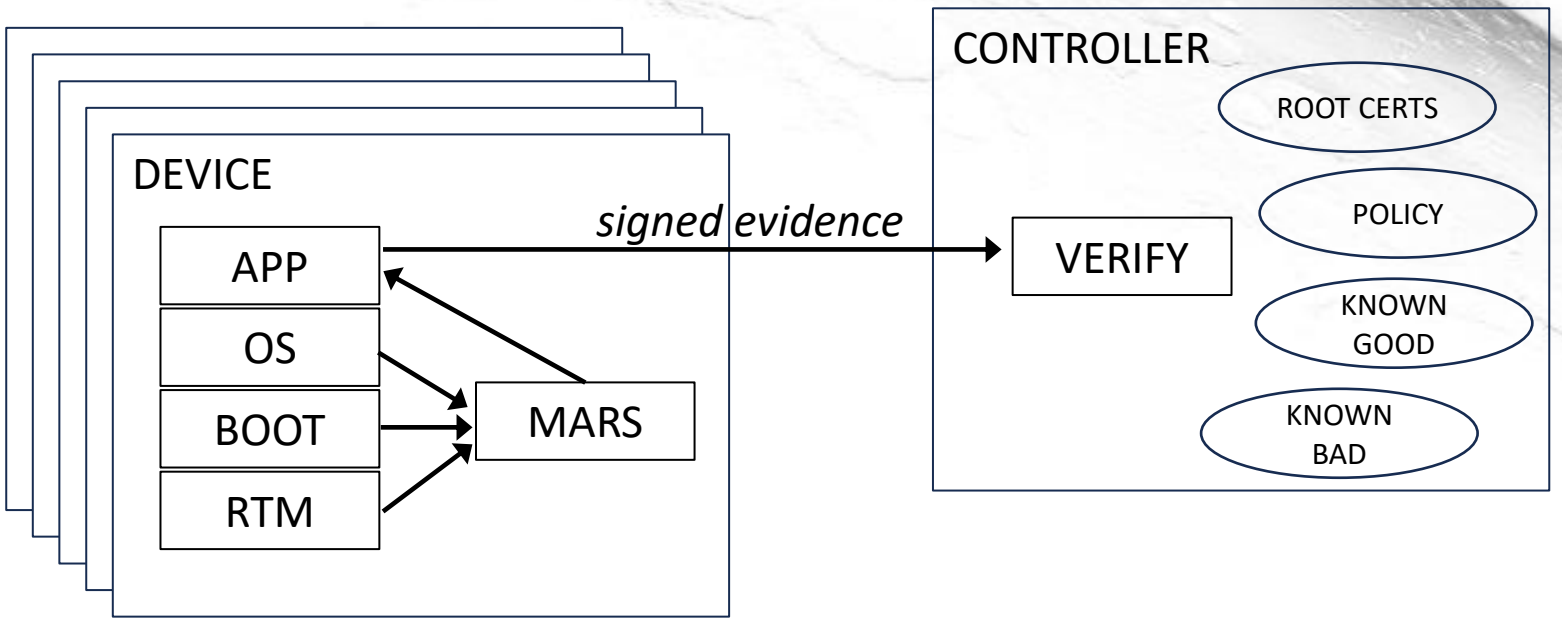
IoT, embedded devices

many adjunct processors



Controller making informed access control decision

does NOT rely on device verifying for itself



Concerns

- How does central controller determine adjuncts’
  - Identity?
  - Health (trustworthiness)?
- Need secure mechanisms to
  - Measure
  - Report

Measurement

- Uniquely identifies block of bytes (code, data)
- Mechanism is cryptographic ***hash***
- hash produces a ***digest***, used as the measurement
- Common hash algorithm is ***SHA256***
- Measuring before use during boot is an ***event***
- Event digests are conveyed in Canonical Event Log, ***CEL***

SHA256 Properties

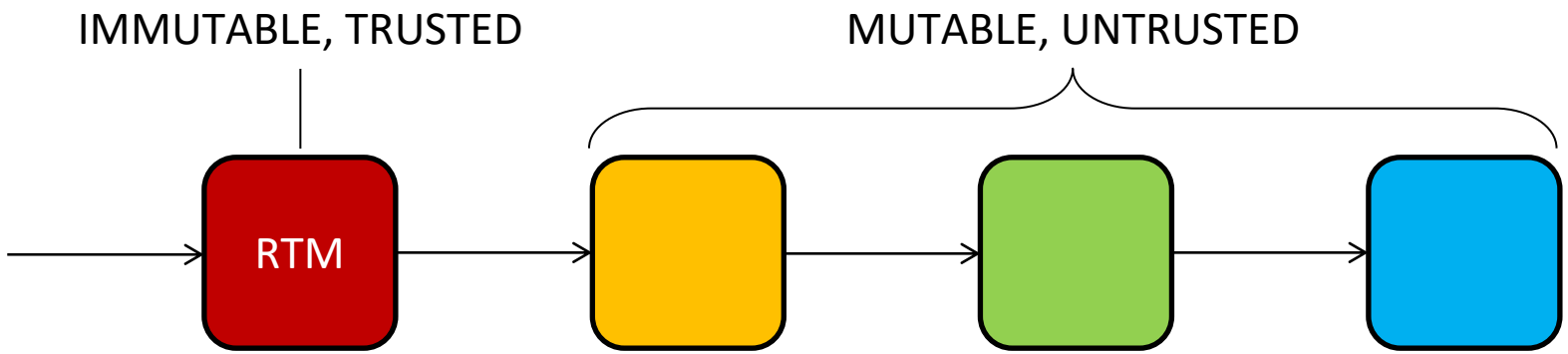
- Input message < 2^{64} bits
- Output digest is always 256 bits
- Infeasible to create an input to produce a given digest
  - Preimage attack
- $2^{256} \sim= 10^{77}$
- 10^{80} protons in universe, the Eddington Number

- `SHA256("Global Platform") = 9edd780028e79f85898f08bfed331777503cc74403a61d77715fe059b28772ed`
- `SHA256("Global P1atform") = d4694d2527a0ab79d8fc1a7d51fe541ef872608d2b53380ea64415fa873187b9`

Verified vs. Measured Boot

- Both compute a measurement
- Both verify measurement against known good

Measured Boot Process



1. LOAD
2. MEASURE
3. EXTEND
4. EXECUTE

# # # # # # # # # # # # # # # # # #

Event Log

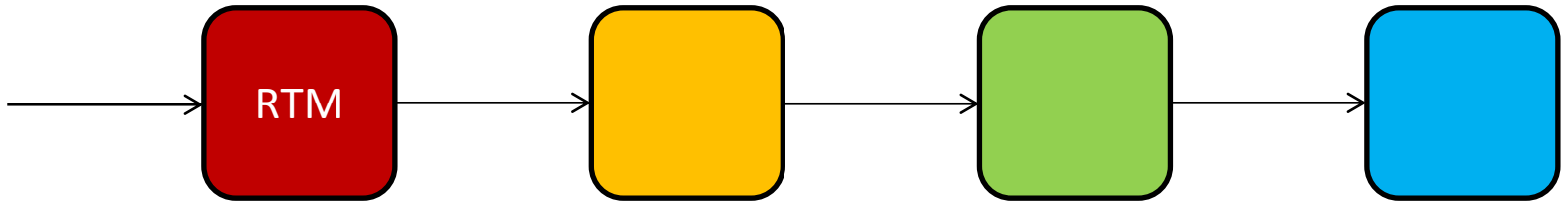
0 0 0 0 0 0

Platform Configuration Register

MARS

ROOTS OF TRUST:
- STORAGE
- REPORTING

Attestation Process



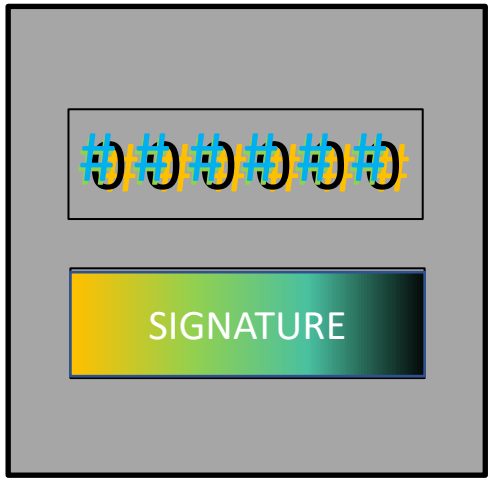
N O N C E

Event Log

# # # # # #

# # # # # #

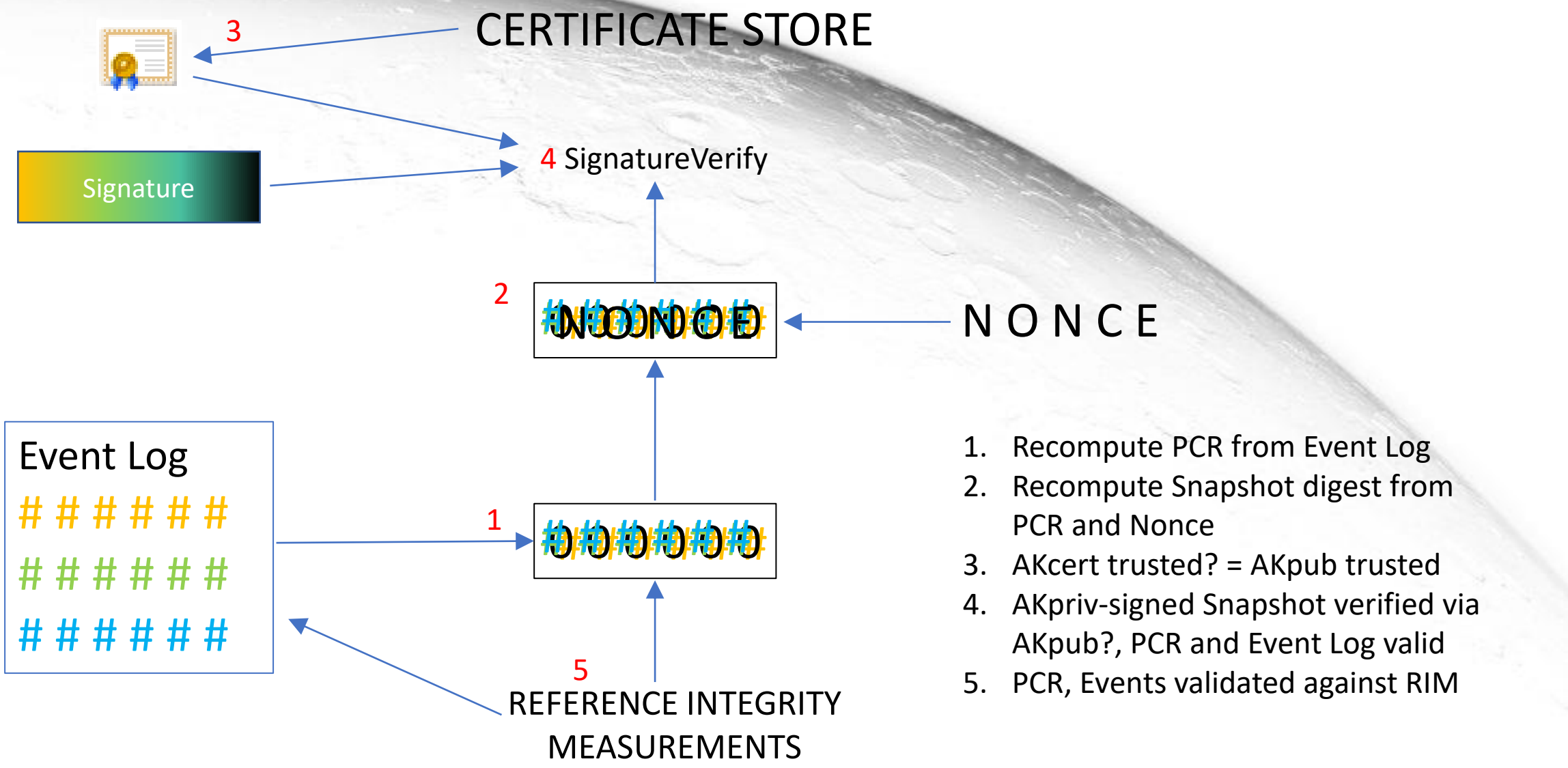
# # # # # #



MARS

Signature
AK Certificate
Event Log

Simplified Assessment Process



1. Recompute PCR from Event Log
2. Recompute Snapshot digest from PCR and Nonce
3. AKcert trusted? = AKpub trusted
4. AKpriv-signed Snapshot verified via AKpub?, PCR and Event Log valid
5. PCR, Events validated against RIM

Platform Configuration Register

- Can only “EXTEND” with another event digest
  - `PCR = HASH( PCR || DIGEST )`
- Integrity check for events
- Malware (already measured) can’t find a digest such that EXTENDing produces a benign PCR (preimage attack)
  - PCR_{GOOD} = HASH(PCR_{BAD} || $DIGEST_{FAKE}$) ??

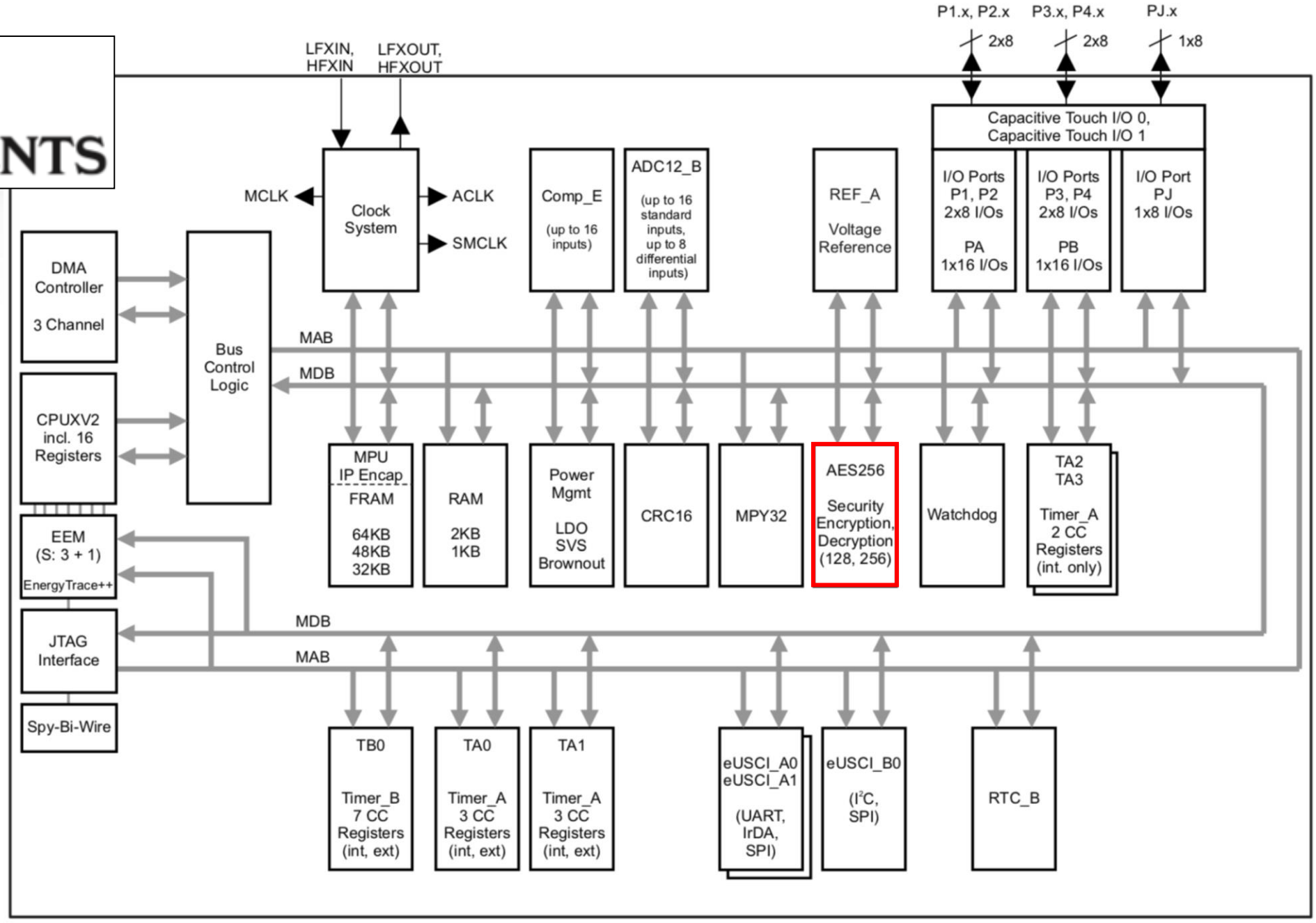
Proposal

- Need HASH to measure and extend
- Need HASH and SIGN to quote PCR
- Need KDF to create attestation key
- Implement around crypto acceleration HW
  - Typically AES and/or SHA
  - Can make crypto-agnostic
- Lightweight state machine (aka *wrapper*) drives crypto accelerator

CORE	HASH	SIGN	KDF
SHA-2	native	HMAC	NIST SP 800-108
AES	ISO/IEC 10118-2	CMAC	NIST SP 800-108
Ascon	native	native	?

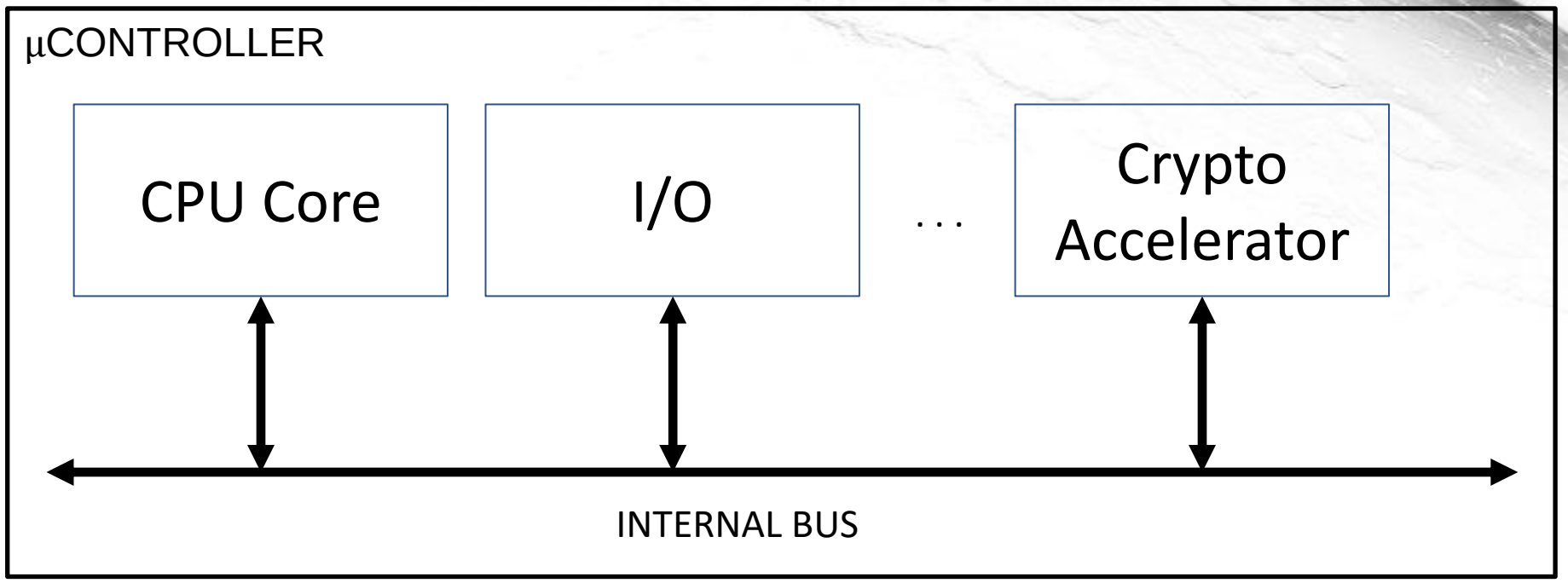
TEXAS INSTRUMENTS

MSP430

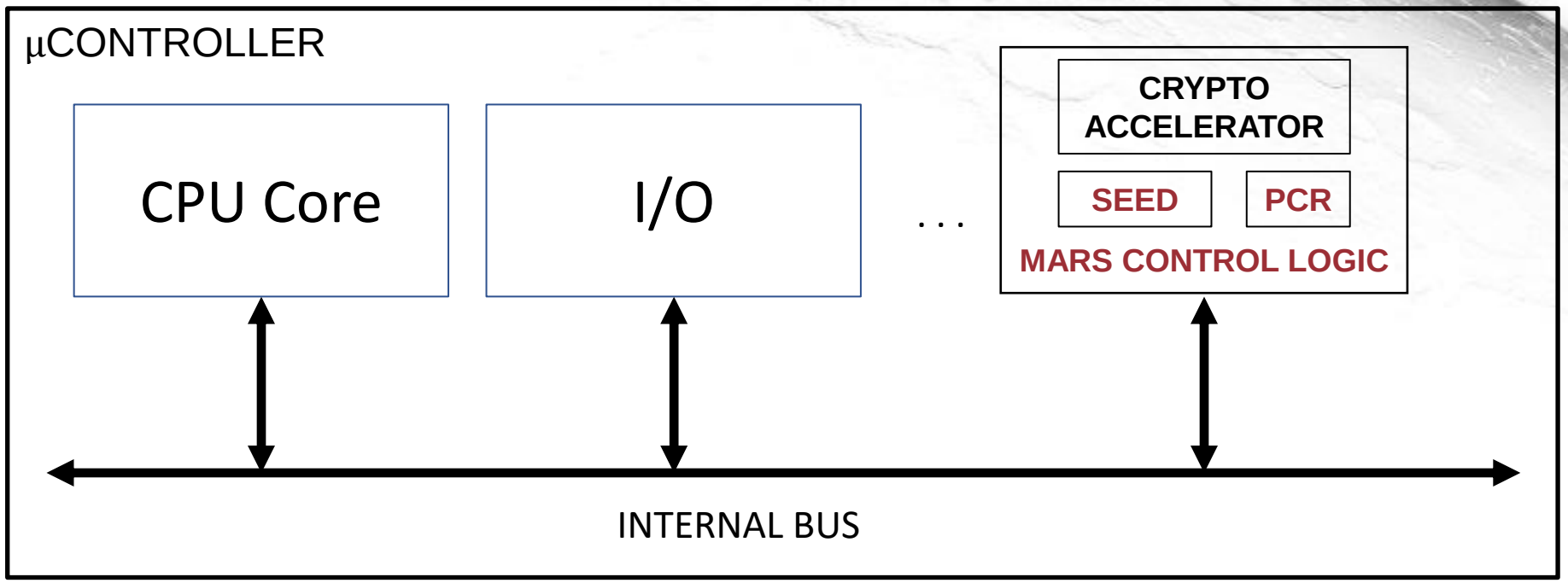


http://www.ti.com/lit/ds/slas704f/slas704f.pdf

Typical Microcontroller



Trusted Platform Architecture with MARS



Quick Look

Measurement and Attestation

COMMAND	M / R / O
MARS_SelfTest	R
MARS_CapabilityGet	M
MARS_SequenceHash	R
MARS_SequenceUpdate	R
MARS_SequenceComplete	R
MARS_PcrExtend	M
MARS_RegRead	M
MARS_Derive	R
MARS_DpDerive	O
MARS_PublicRead	M*
MARS_Quote	M
MARS_Sign	R
MARS_SignatureVerify	R

Mandatory
Recommended
Optional

Hashing thoughts

- Used for measured boot and verified (trusted) boot
- Concern over potential increase to boot time
- MARS – fast hardware on fast internal bus
- Ascon example
  - Performance measured in clock cycles per bytes, c/B
  - Software alone: 203.8 c/B
  - Hardware assist: 2.5 c/B – faster than host processor can feed?

https://ascon.iaik.tugraz.at/implementations.html

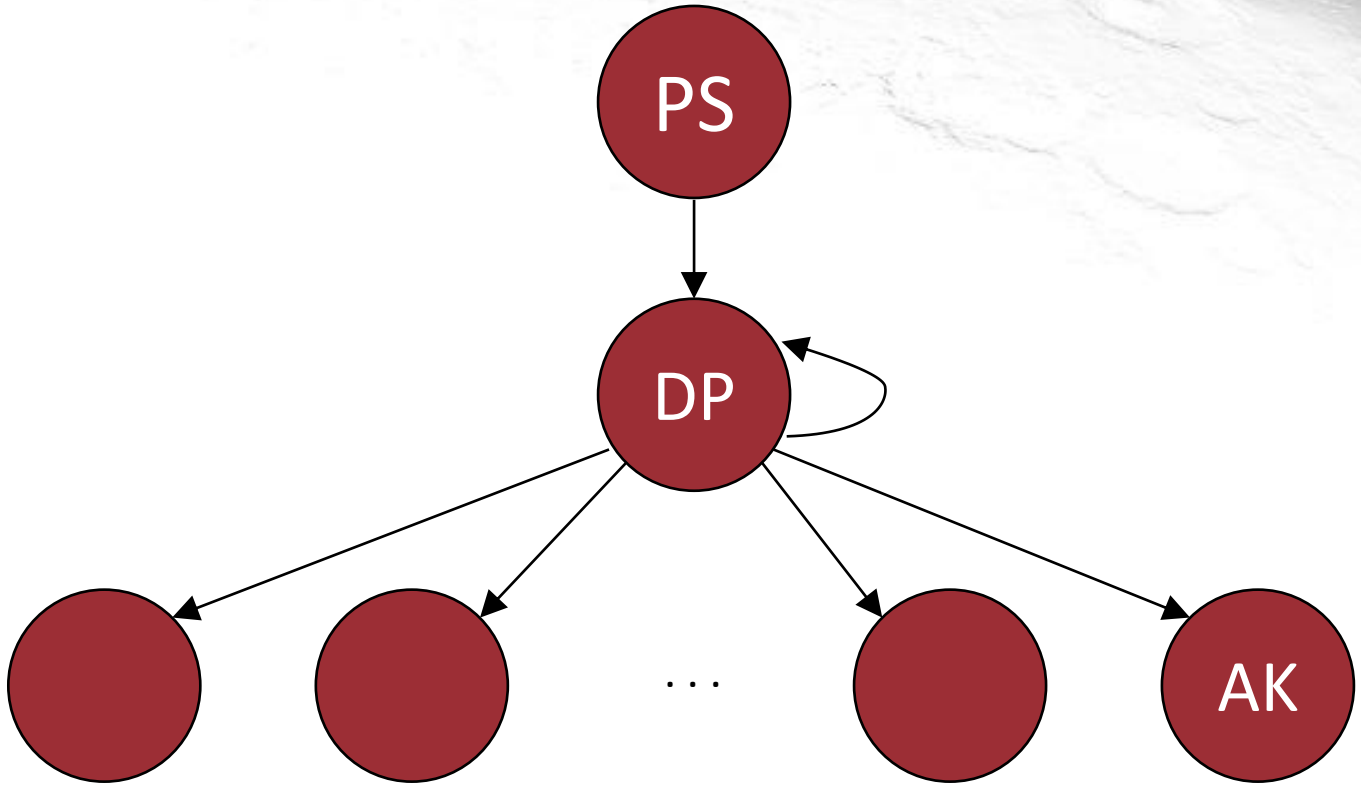
Load and Measure

```
// Load text segment
count = ((uint32_t)&TEXT_ROM_size + 3) >> 2;
p1 = (uint32_t *)&TEXT_ROM_start;
p2 = (uint32_t *)&TEXT_RAM_start;
while (count--)
  *p2++ = *p1++;
```

```
// Load and measure text segment
count = ((uint32_t)&TEXT_ROM_size + 3) >> 2;
p1 = (uint32_t *)&TEXT_ROM_start;
p2 = (uint32_t *)&TEXT_RAM_start;
while (count--)
  *hasher = *p2++ = *p1++;

// assumes hasher is fast enough
```

MARS Key Hierarchy

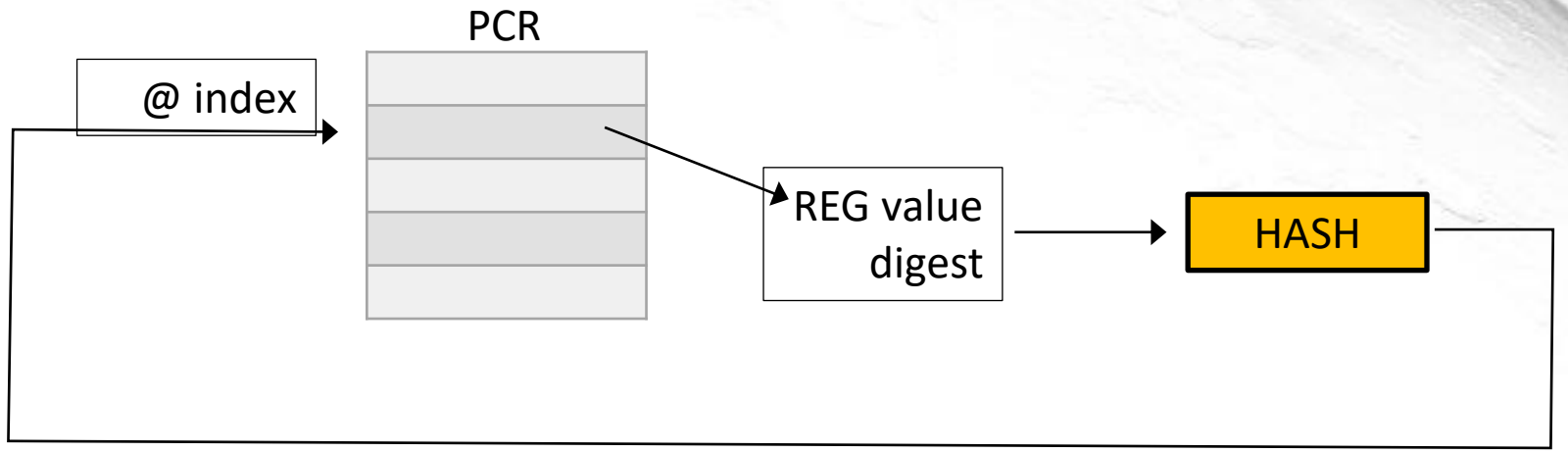


PS = Primary Seed
- Only 1
- Provisioned

DP = Derivation Parent
- Derived from PS at init
- All other values implicitly derived from DP
- DP can be derived from itself
- DP can be re-init'd

AK = Attestation Key
- example

MARS_PcrExtend(index, digest)



TSR – Trusted Sensor Register

- Adjacent to PCR
- Can Quote, but NOT Extend
- Copied from external HW when building snapshot
- Device cannot lie about sensor reading

MARS_Quote(regSelect, nonce, context)

Sampling logic

PCR, TSR

regSelect
REG value
…
REG value
nonce

DP
'R'
context

KDF

AK

HASH

snapshot

SIGN

signature

Pre-MARS hardware

- 1 PCR
- Ascon AEAD cipher
- OpenTitan
- Memory-mapped I/O
- RISC-V µProcessor
- FreeRTOS

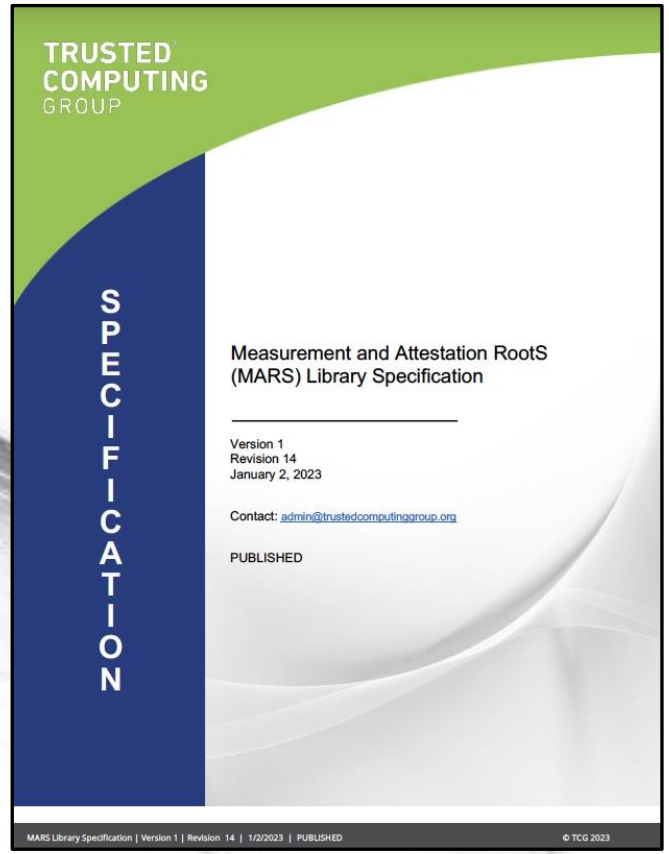
SPECIFICATIONS	FPGA	ASIC
DEVICE	CMOD A7-35T Xilinx Artix-7	45 nm 3mm x 3mm die BGA304 package
ROM *	16 KB	16 KB
RAM	64 KB	256 KB
FLASH *	48 KB	128 KB
CLOCK	50 MHz	100 MHz

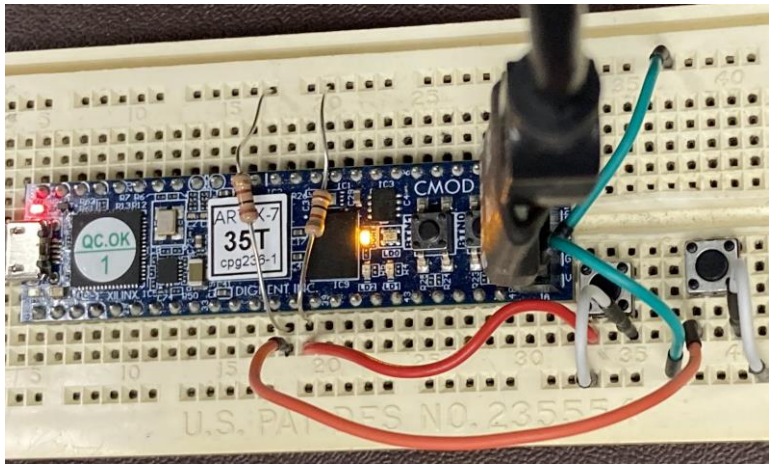
* emulated via RAM

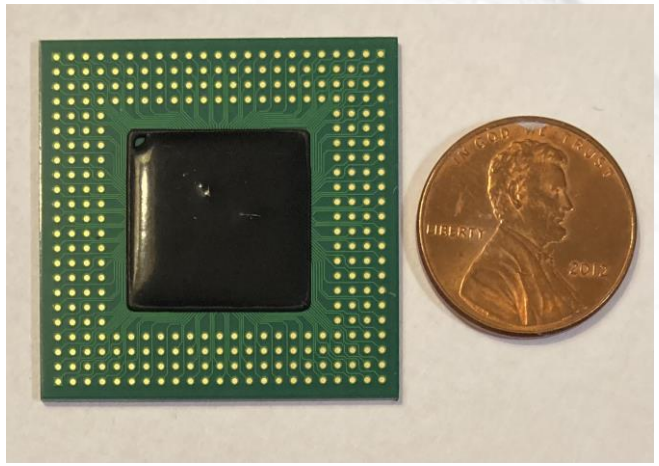
MARS Resources

- Use Cases and Considerations
- Library Specification
- Errata for Library Spec
- FAQ
- API Specification
- Register Interface Specification (in progress)
- Serialization Interface Specification (in progress)
- Profile guidance (on hold)
- Briefing at TPM.dev 2023

- Emulators
  - C via TCG GitHub
  - Python via TCG GitHub
  - Rust (in progress)
- CDDL via TCG GitHub
- Pre-MARS Hardware
  - VHDL (via UMBC)
  - FPGA (via UMBC)
  - ASIC (via USG)







Q&A