Caliptra

An open source, reusable silicon IP block for a Root of Trust for Measurement (RTM)

Andrés Lagar-Cavilla (Google)
Prabhu Jayanna (AMD)
Bryan Kelly (Microsoft)
What is Caliptra

• An OCP specification for a **silicon Root of Trust** internal block
• Targeting **SoCs** and **ASICs** in the **hyperscaler/datacenter** space
• **Goals:**
  - implementation consistency, transparency, openness, reusability
• A **multi-party collaboration** including (today):
  - Google, AMD, Microsoft
• An **open source** implementation of the specification
• The first Security project specification proposing a technology block
• Work in progress!
Targets

- Datacenter devices use by CSPs, hyperscalers
- Not for phones
- Not a discrete or platform RoT
- Key priority are devices handling plaintext user data
  - SoC, GPU, NIC/IPU/DPU
  - Provide a transparency substrate to root confidential compute
  - For example, could fulfill the HES role in Arm [RME spec](#)
- Follow on: devices handling cipher text
  - Storage, [NV] DIMMs, switches
Architectural Role

“Composable Security Architectures” in 2021 OCP summit

Let’s define this today

Examples:
- BMC
- iLO
- iDRAC
- Titan
- Cerberus
- PFR
- CEC1712

Platform Root of Trust

CPU
IPU
Your ASIC
GPU

## What is an RTM?

### NIST 800-193, Platform Firmware Resiliency Guidelines

<table>
<thead>
<tr>
<th>Detection</th>
<th>RTM: RoT for Measurement (a.k.a. RTD)</th>
<th>Integrated Silicon RoT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>• Well and narrowly defined job</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Measure, verify and attest</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In package – best bet against physical attacks on integrity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Limited fuses</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protection</th>
<th>RTU: RoT for Update</th>
<th>Discrete RoT Chip</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>• Mitigate DoS at scale</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• RTU: reject random blobs pushed at scale</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• RTRec: automated recovery against buggy updates</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Ok to be a separate discrete element</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Physical attacks irrelevant to scalable DoS mitigation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Integrated flash for unlimited renewability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Enforce versions, owners, rotations</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recovery</th>
<th>RTRec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical attacks irrelevant to scalable DoS mitigation</td>
<td></td>
</tr>
</tbody>
</table>
# Value of Decoupling

**NIST 800-193, Platform Firmware Resiliency Guidelines**

<table>
<thead>
<tr>
<th>Detection</th>
<th><strong>RTM</strong> : RoT for Measurement (a.k.a. RTD)</th>
<th><strong>Integrated Silicon RoT</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>• Well and narrowly defined job</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Measure, verify and attest</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In package – best bet against physical attacks on integrity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Limited fuses</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Not concerned with update or ownership</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protection</th>
<th><strong>RTU</strong> : RoT for <strong>Discrete RoT Chip</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Decoupling Update and Recovery DoS mitigations usefully simplifies the SoC RTM</td>
</tr>
<tr>
<td></td>
<td>• No need for persistent flash</td>
</tr>
<tr>
<td></td>
<td>• No need for update, fallback, A/B schemes</td>
</tr>
<tr>
<td></td>
<td>• No need for TPM behaviors, or persistent ownership</td>
</tr>
</tbody>
</table>

Our Goals

• RTMs have a well and narrowly defined job
  - Measure, verify and attest
  - No need for update, fallback, A/B recovery, TPM, ownership flows
• Useful simplification leads to easier path for convergence
  - Aligned/converged specification
  - Open Sourcing
  - Transparency
  - Reusability
  - Implementation Consistency
  - These are our goals
<table>
<thead>
<tr>
<th>Identity</th>
<th>Manufacturer Identity aligned to TCG DICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement</td>
<td>Code &amp; configuration posture of the device.</td>
</tr>
<tr>
<td>Lifecycle</td>
<td>Debug mode (ON/OFF), established at reset.</td>
</tr>
<tr>
<td>Ownership</td>
<td>No stateful transfer. Vendor authored firmware only, with stateless Owner Authorization</td>
</tr>
<tr>
<td>Attestation</td>
<td>Identity &amp; Measurement reporting using DMTF SPDM v1.2+</td>
</tr>
</tbody>
</table>

**Mutable Firmware**

- Attestation
- Identity
- Measurement
- Authentication

**Hardware**

- Fuses
- ROM
- SRAM
- CPU
- Cryptos
- Analog, TRNG
- IOs
- SoC
- RoT

---

**Caliptra: Behavioral Elements**

[Image of Caliptra's architectural elements]
Summary of Key Behaviors

• **Measurement**  
  - Load fw, measure fw, and release fw target from reset

• **Configuration**  
  - Measure relevant security configuration state  
  - JTAG enablement, GPIO/straps/fuses

• **Attestation**  
  - Form an attestation and sign it with unforgeable entropy

• **Identity**  
  - Provide and protect unique asset entropy (DICE UDS)

• **Identity Service**  
  - For example provide derived keys from UDS to core
Critical Decisions

- Not interested in differentiation
  - Caliptra is not a landing pad for vendor “value adds”
- Support only manufacturer signed RTM code
  - Code roots back to open source fw development
- Leans on DICE
  - FMC mixed into UDS to generate an Alias keypair
  - Derived Alias key signs attestation
- Stateless ownership enforcement
  - A silicon owner can additionally sign code with their key
  - The public signing key is reported in attestation
  - Until next power-on, owner key enforces runtime upgrades
- SPDM responder
  - GET_CERTIFICATES 1.2 (or 1.3) responder for attestation
- Facilitate SoC integration
High Level Block Structure

RISC-V CPU

Tightly coupled memories

Integration: mailbox to the SoC

Cryptos: accelerate boot path

IOs: Debug, boot path, SPDM path

Making It SOC Independent

- Caliptra boots first, reads its fw, creates identity
- Copies and measures SOC FMC firmware into SRAM buffer
- Releases SOC ROM from reset to boot
- SOC ROM loads FW using current flows and authentication
- Simplifies integration, preserves existing security flows.
- Attestation is always done through, and rooted to, Caliptra
Ownership & Implementation Consistency

DC owner Authorization / Dual Signing

- Open Source Caliptra FW
- Open Source Caliptra HW

Sig: 1 HW MFG
Sig: 2 DC Owner

Manufacturer signed fw

Open Source Caliptra FW
Open Source Caliptra HW

Allowing owner signed custom FW impacts implementation consistency
- Perils of fragmentation and forking
- Gratuitous scope-creep

Proprietary FW
Proprietary/OEM-signed forked fw

BSP
Caliptra HW

Not Allowed
DICE Identity

- Co-signs Caliptra firmware
- Latched into Caliptra RAM on cold-boot
- Authorizes runtime Caliptra updates

Manufacturers CA

DC owner CA

UDS

FMC

ROM

FMC CDI

Runtime fw CDI

Runtime Alias

Signs

Signs

Signs

ROM
Commitments

- Google silicon target 2024
- AMD silicon target 2026+
- Microsoft silicon since 2024
Work In Progress

- Three parties in CLA
- v0.5 spec by end of May, then publish to community
- Partners welcome
  - Must commit to integrate and contribute!
- RTL in progress
  - fw to follow on
- Committed to open sourcing RTL, FW and Specifications
FAQ (don’t panic)

• Not a whole chip! It’s an IP block

• Not intending to over-specify how you should build your SoC
  - Not mandating backend IP synthesis methods
  - Not mandating certification criteria
  - Not mandating analog IPs, or counter-measures
  - Not mandating a fuse technology or a process node
  - Not mandating manufacturing operational security processes

• Not a datapath element or general purpose crypto accelerator!
Caliptra: Take Home

- Silicon IP block for integration into SoC
- Hyperscaler/datacenter device targets
- Public specification, open source logic and fw
- RTM: Measurement, attestation, identity
- Goals are implementation consistency, portability, transparency, openness
- Explicitly decoupled other security functions to achieve goals
- Google, AMD, Microsoft
  - Contributors who will integrate are welcome!
Caliptra

An open source, reusable silicon IP block for a Root of Trust for Measurement (RTM)
Andrés Lagar-Cavilla (Google)
Prabhu Jayanna (AMD)
Bryan Kelly (Microsoft)

Thanks!
Q&A