TEEを中心とするCPUセキュリティ機能の動向 (RISC-V, ARM, etc)

国立研究開発法人 産業技術総合研究所
サイバーフィジカルセキュリティ研究センター
須崎有康

Do you know TEE?

- I asked same question at RISC-V Day Tokyo 2018 and MICRO51 RISC-V workshop 2018.
- Only 10% attendees know it.

Do you know there words?

- Real Product
  - TPM (Trusted Platform Module)

  - ARM TrustZone
  - ARM SCP (System Co Processor)
  - Intel SMM (System Management Mode)
  - Intel TXT (Trusted Execution Technology)
  - Intel SGX (Software Guarded Extension)
  - Intel ME (Management Engine)
  - AMD PSP (Platform Security Processor)

  - Google Titan
  - MS Azure Pluton
  - Apple Secure Enclave
  - Apple T2

- Research *
  - IBM 4765 Secure Coprocessor
    - FIPS 140-2 level 4
      - Wikipeida: FIPS 140-2 Level 4 makes the physical security requirements more stringent, and requires robustness against environmental attacks.
  - MIT Aegis Secure Processor [ICS'03]
  - MIT Sanctum [USENIX Sec'15]

* Secure Processors Part I: Background, Taxonomy for Secure Enclaves and Intel SGX Architecture [2017, Srinivas Devadas]

Contents

- What is CPU Security?
- What is TEE?
- Implementation
- Related topic (TEEP)

- Intel ME, Google NERF, Google Titan, MS Azure, etc. (Messy)

Why is CPU Security needed?

- Application is not trustable.
  - Quality is not managed.
- OS kernel is not trustable.
  - Too large for TCB.
  - Security function (e.g.,Reference Monitor) is a part of OS.
  - Hardware isolation (privilege) is not enough.
- Hypervisor is not trustable.
  - Hardware isolation (virtualization) is not enough.

CPU which runs apps, kernel, and hypervisor is not trustful.

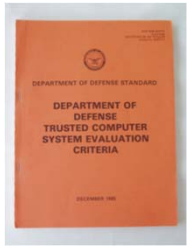
HIEE (Hardware-assisted Isolated Execution Environments) is required.

* SoK：A Study of Using Hardware-assisted Isolated Execution Environments for Security[HASP16]

History of HIEE

1990 2000 2010

Orange Book
"Trusted Computer System Evaluation Criteria"
(83,85,05)



MS NGSCB: Next-Generation Secure Computing Base (2002) ----------→ TPM of TCG: Trusted Computing Group
2006 Intel Mac
2009 as ISO/IEC 11889
Ver1.2 March 3, 2011
Ver2.0 Sep 29, 2016

Static Root of Trust →

Dynamic Root of Trust
Intel TXT(2007)
Intel ME(2008)

Intel SMM
i386SL (1990)
Suspend/Resume
20 MHz clock

TEE

Intel SGX
(2015 Skylake)
(2014 QEMU by GIT)

ARM TrustZone (02～)
Cortex-A
Cortex-M (ARMv8-M)

GlobalPlatfrom's TEE
July 2010

Comparison of HIEE

Table 2: Summary of HIEEs

	SMM	ME	PSP	DRTM	SGX	TrustZone
Timelines	~1993	~2007	~2013	~2005	~2013	~2002
Supported hardware	x86	Intel	AMD	Intel/AMD	Intel	ARM
Sharing main CPU	✓			✓	✓	✓
High privilege	✓	✓	✓			✓
Zero overhead		✓	✓			
Designed for security		✓	✓	✓	✓	✓

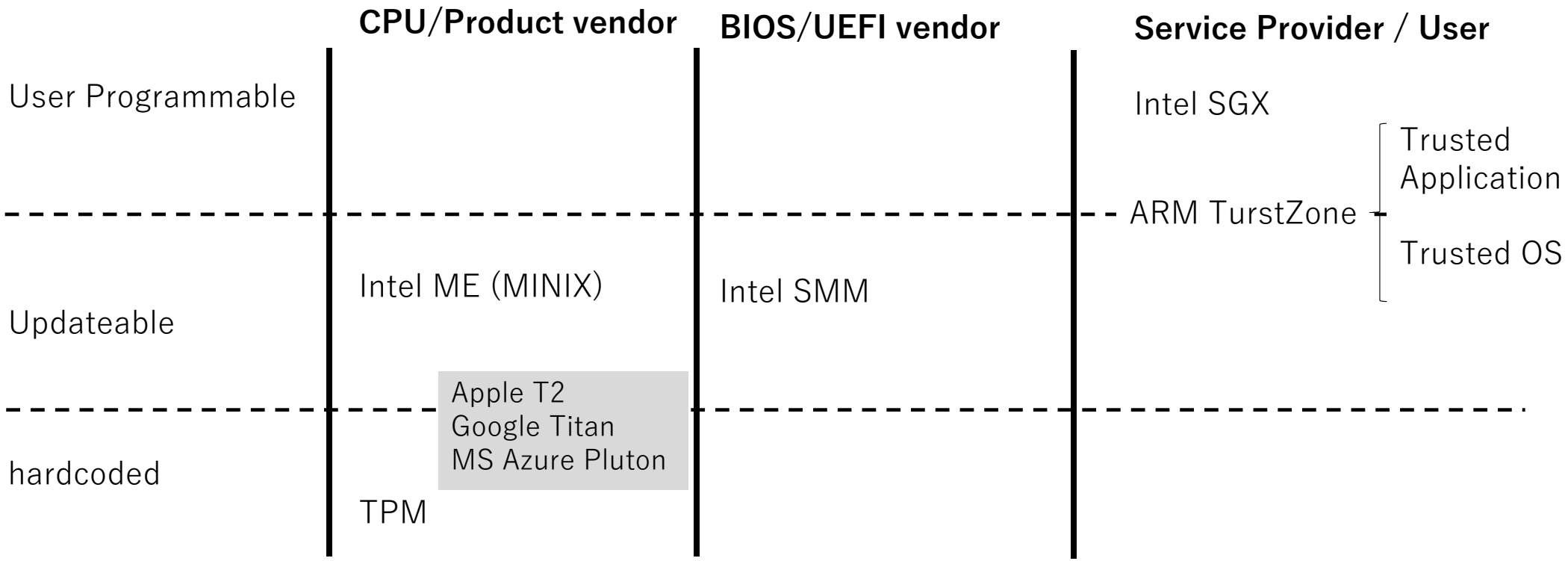
From: SoK：A Study of Using Hardware-assisted Isolated Execution Environments for Security[HASP16]

Comparison of CPU mechanism

Attack	TrustZone	TPM	TPM+TXT	SGX	Aegis	Sanctum
Malicious containers (direct probing)	N/A (secure world is trusted)	N/A (The whole computer is one container)	N/A (Does not allow concurrent containers)	Access checks on TLB misses	Security kernel separates containers	Access checks on TLB misses
Malicious OS (direct probing)	Access checks on TLB misses	N/A (OS measured and trusted)	Host OS preempted during late launch	Access checks on TLB misses	Security kernel measured and isolated	Access checks on TLB misses
Malicious hypervisor (direct probing)	Access checks on TLB misses	N/A (Hypervisor measured and trusted)	Hypervisor preempted during late launch	Access checks on TLB misses	N/A (No hypervisor support)	Access checks on TLB misses
Malicious firmware	N/A (firmware is a part of the secure world)	CPU microcode measures PEI firmware	SINIT ACM signed by Intel key and measured	SMM handler is subject to TLB access checks	N/A (Firmware is not active after booting)	Firmware is measured and trusted
Malicious containers (cache timing)	N/A (secure world is trusted)	N/A (Does not allow concurrent containers)	N/A (Does not allow concurrent containers)	×	×	Each enclave its gets own cache partition
Malicious OS (page fault recording)	Secure world has own page tables	N/A (OS measured and trusted)	Host OS preempted during late launch	×	×	Per-enclave page tables
Malicious OS (cache timing)	×	N/A (OS measured and trusted)	Host OS preempted during late launch	×	×	Non-enclave software uses a separate cache partition
DMA from malicious peripheral	On-chip bus bounces secure world accesses	×	IOMMU bounces DMA into TXT memory range	IOMMU bounces DMA into PRM	Equivalent to physical DRAM access	MC bounces DMA outside allowed range
Physical DRAM read	Secure world limited to on-chip SRAM	×	×	Undocumented memory encryption engine	DRAM encryption	×
Physical DRAM write	Secure world limited to on-chip SRAM	×	×	Undocumented memory encryption engine	HMAC of address, data, timestamp	×
Physical DRAM rollback write	Secure world limited to on-chip SRAM	×	×	Undocumented memory encryption engine	Merkle tree over HMAC timestamps	×
Physical DRAM address reads	Secure world in on-chip SRAM	×	×	×	×	×
Hardware TCB size	CPU chip package	Motherboard (CPU, TPM, DRAM, buses)	Motherboard (CPU, TPM, DRAM, buses)	CPU chip package	CPU chip package	CPU chip package
Software TCB size	Secure world (firmware, OS, application)	All software on the computer	SINIT ACM + VM (OS, application)	Application module + privileged module + containers	Application module + security kernel	Application module + security monitor

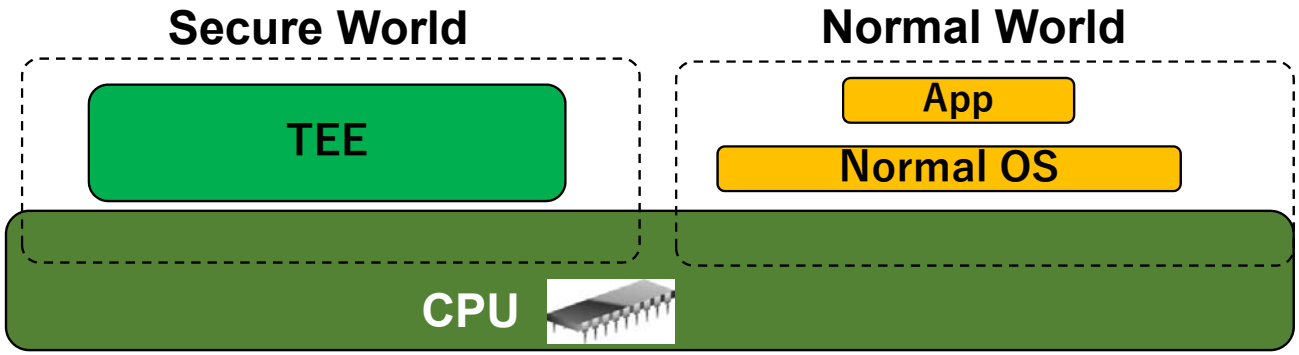
From: Secure Processors Part I: Background, Taxonomy for Secure Enclaves and Intel SGX Architecture [2017, Srinivas Devadas]

Stakeholder map



What is TEE?

- TEE: Trusted Execution Environment.
  - TEE separates computing world into “normal” and “secure”.
    - Secure world is used to run a critical code (e.g., authentication, DRM, etc).



  - GlobalPlatform defines TEE specification.
    - https://globalplatform.org/technical-committees/trusted-execution-environment-tee-committee/

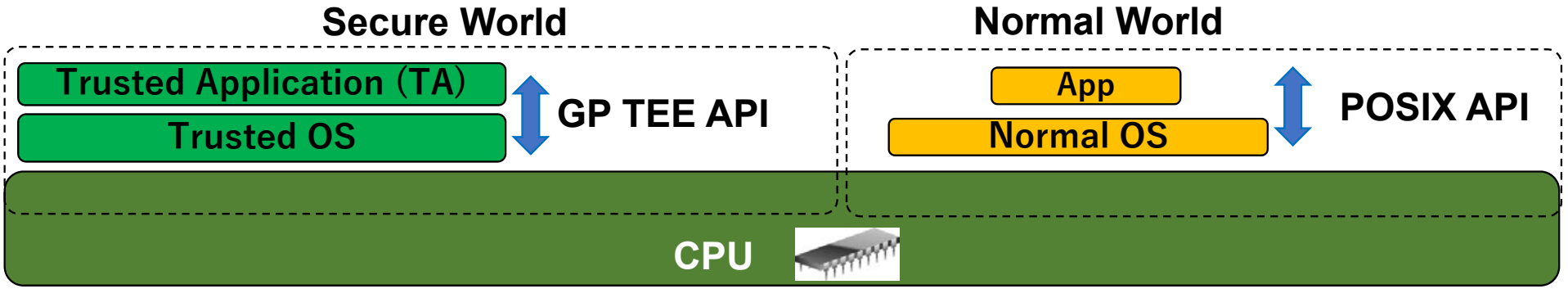
GlobalPlatfrom TEE

- Requirements:
  - https://globalplatform.org/wp-content/uploads/2018/05/Introduction-to-Trusted-Execution-Environment-15May2018.pdf

1. Isolation from the Rich OS
   - all trusted applications and their related data are separated from the rich environment.
2. Isolation from other TAs
   - TAs are isolated within the TEE, and from the TEE itself.
3. Application management control
   - any modification of the TA and the TEE can only be performed by the authenticated entity.
4. Identification and binding
   - where the boot process is bound to the System-onChip (SoC), enforcing authenticity and integrity of TEE firmware and TAs.
5. Trusted storage
   - TA and TEE data is stored security to ensure integrity, confidentiality and binding to the TEE (or anti-cloning).
6. Trusted access to peripherals
   - the TEE offers APIs access to trusted peripherals such as the screen, biometric sensors and SEs, under the control of the TEE.
7. State of the art cryptography
   - random number generation, cryptography and monotonic time stamps are key assets for value added services.

Privileges for TEE

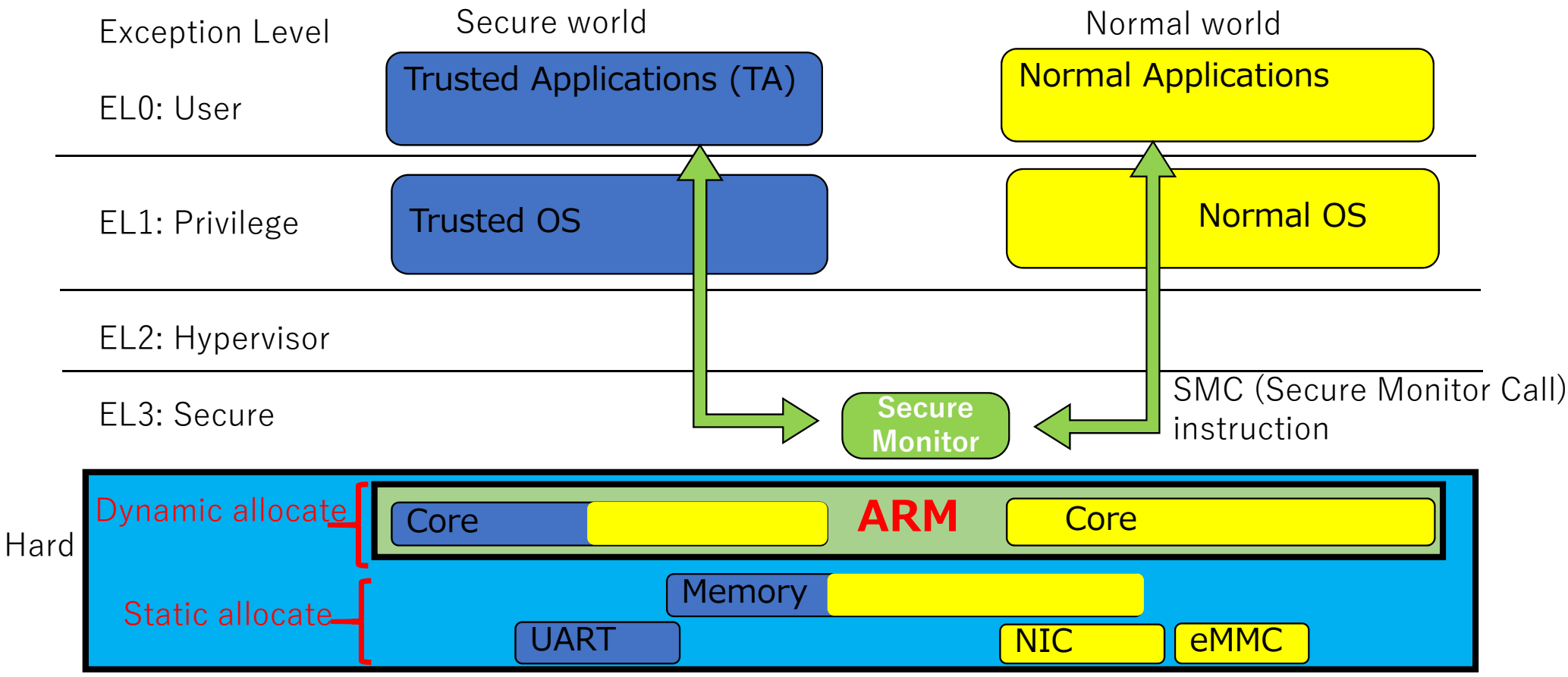
- Global Platform's TEE specification assumes plural privileges on both worlds.
  - Normal world runs normal applications on a normal OS.
  - Secure world runs trusted applications (TAs) on a trusted OS.



- ARM Trust Zone offers same privileges to normal and secure world.
- Intel SGX has only one privilege (enclave).
  - Enclave is different from Ring architecture.

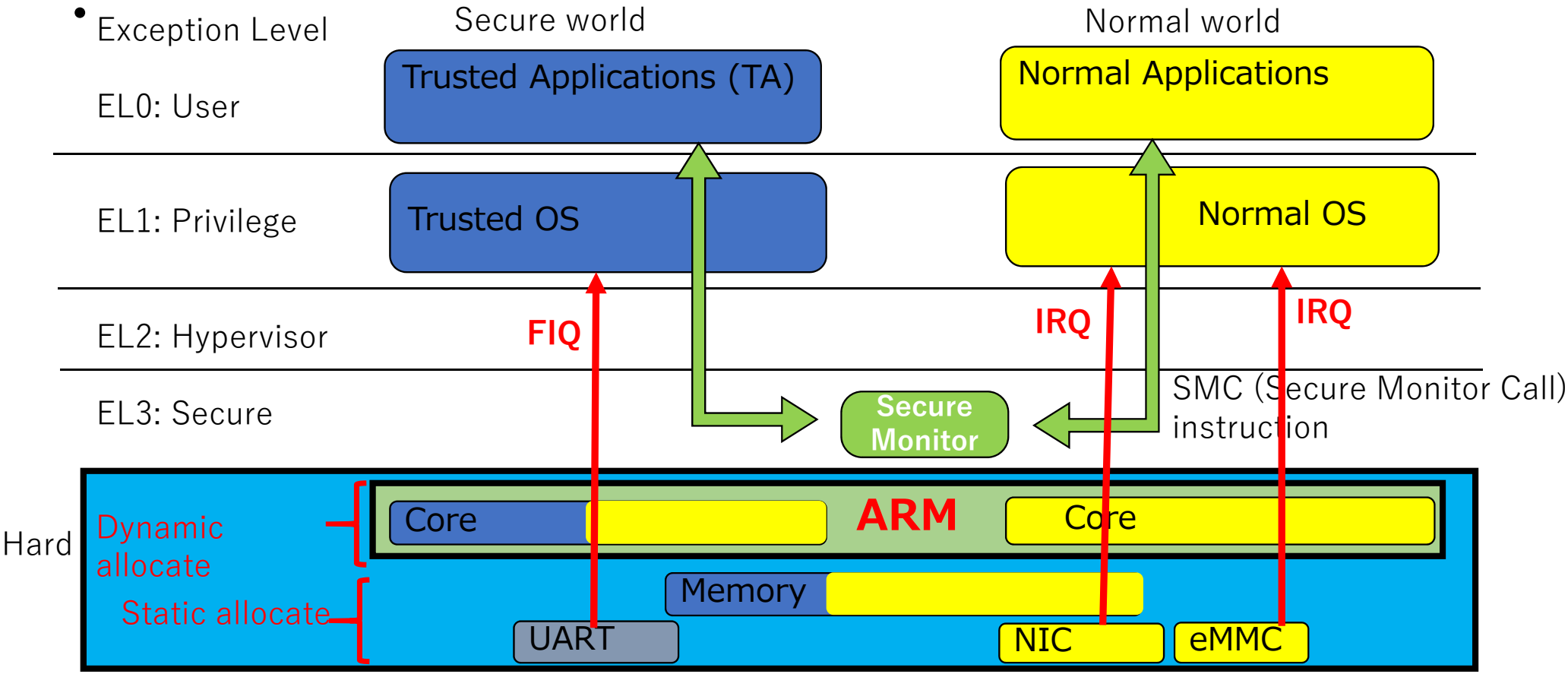
Trusted OS on ARM Trust Zone

- GlobalPlatform model



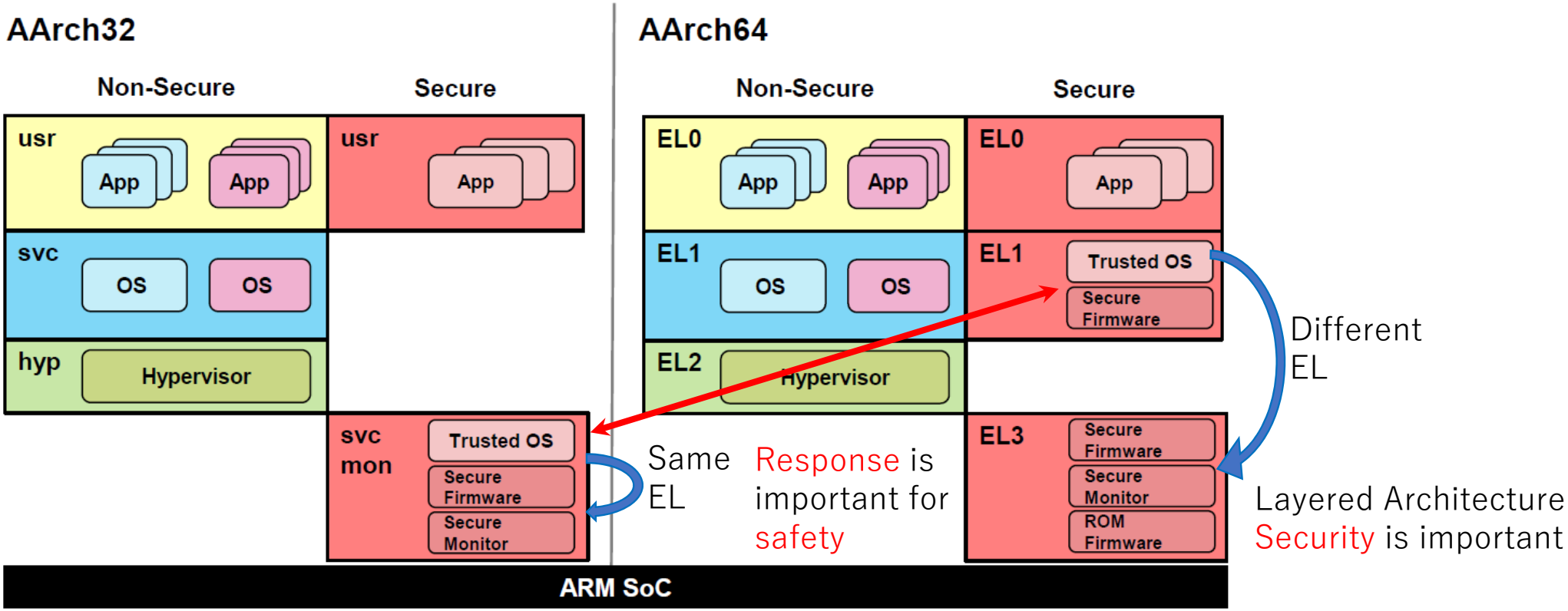
Trusted OS on ARM Trust Zone

- GlobalPlatform model
  - Interrupt is also separated. (depending on configurations)
  - Exception Level



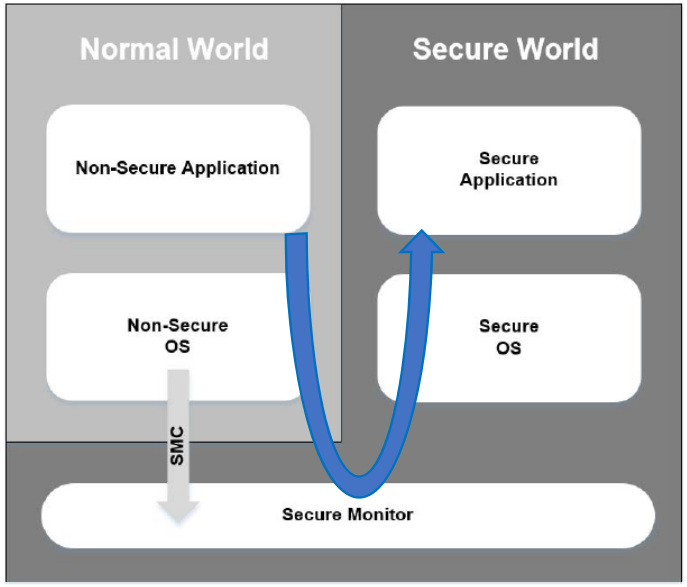
Difference of Implementation of Trusted OS

- Cortex-A 32bit (ARMv7) and 64bit (ARMv8)

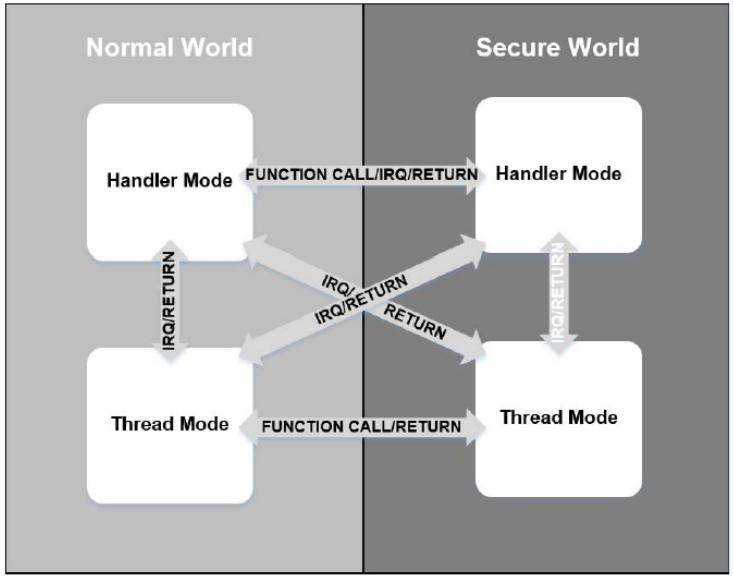


https://www.slideshare.net/linaroorg/arm-trusted-firmareforarmv8alcu13

Comparing Cortex-A and Cortex-M



Cortex-A



Cortex-M

- Cortex-A follows the layer architecture of GlobalPlatform TEE.
- Cortex-M's mode (thread or handler) can be privilege or unprivileged.
- Cortex-M TrustZone doesn't provide monitor mode, because latency is important for safety.

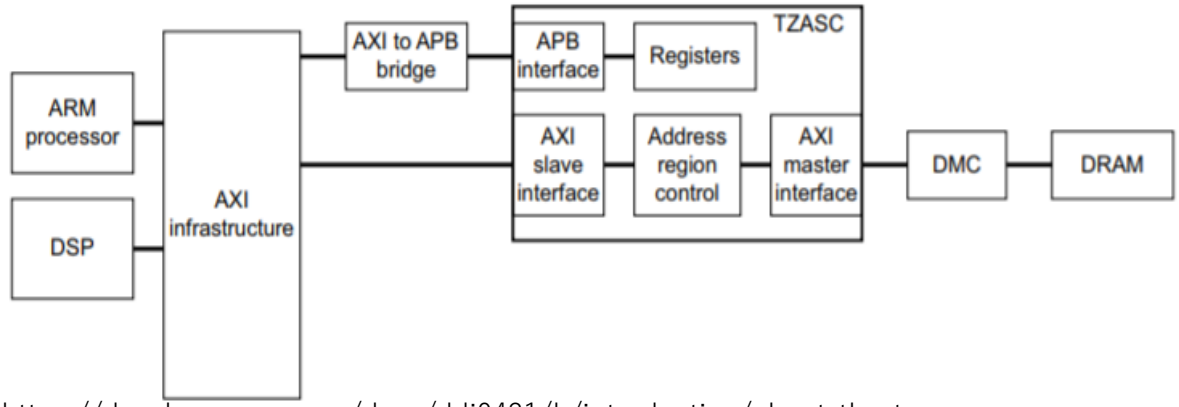
Bernard Ngabonziza, Daniel Martin, Anna Bailey, Haehyun Cho and Sarah Martin, "TrustZone Explained: Architectural Features and Use Cases",IEEE International Conference on Collaboration and Internet Computing (2016)

Hardware components to build TrustZone

- TZASC: TrustZone Address Space Controller
- TZPC: TrustZone Protection Controller
- TZMA: TrustZone Memory Adapter

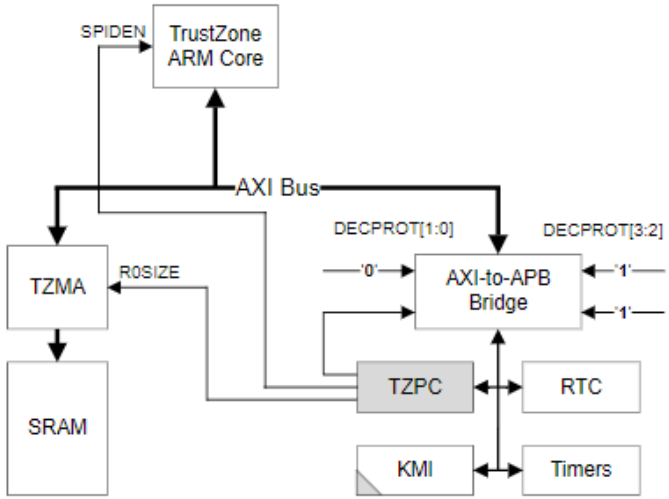
- Each Peripheral has **Non-Secure bit**
  - IRQ for Normal World
  - FIQ for Secure World

Memory managed by TZASC



https://developer.arm.com/docs/ddi0431/b/introduction/about-the-tzasc

Peripheral managed by TZASC

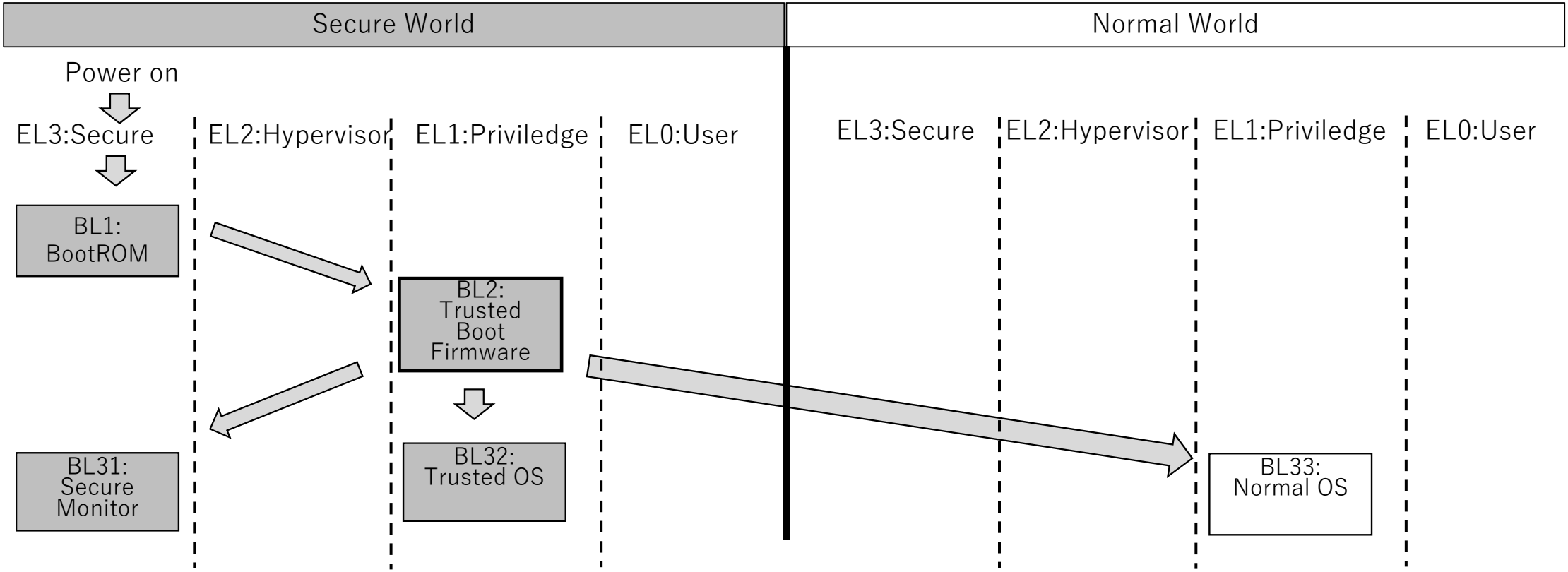


http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.prd29-genc-009492c/ch04s01s01.html

OP-TEEのソースでは Address Space ControllerにTZC-380とTZC-400の記述がありました。
ARM Trusted FirmwareではTZC-400はありますが、TZC-380が見つからない。

Boot Sequence on ARM Trust Zone

- BL: Boot Loader
- EL: Exception Level



Trusted OS

- Trusted OS is not a normal OS
  - Trusted OS is TCB (Trusted Computing Base). It must be secure (small).
  - No POSIX API, No dynamic link library
    - TA becomes a static linked binary.
- Trusted OS needs the help of normal OS
  - Because Trusted OS has no File System, no device driver (except some special devices, e.g., UART)
    - When a TA want to save a data, the data is encrypted and saved on FS of normal OS.

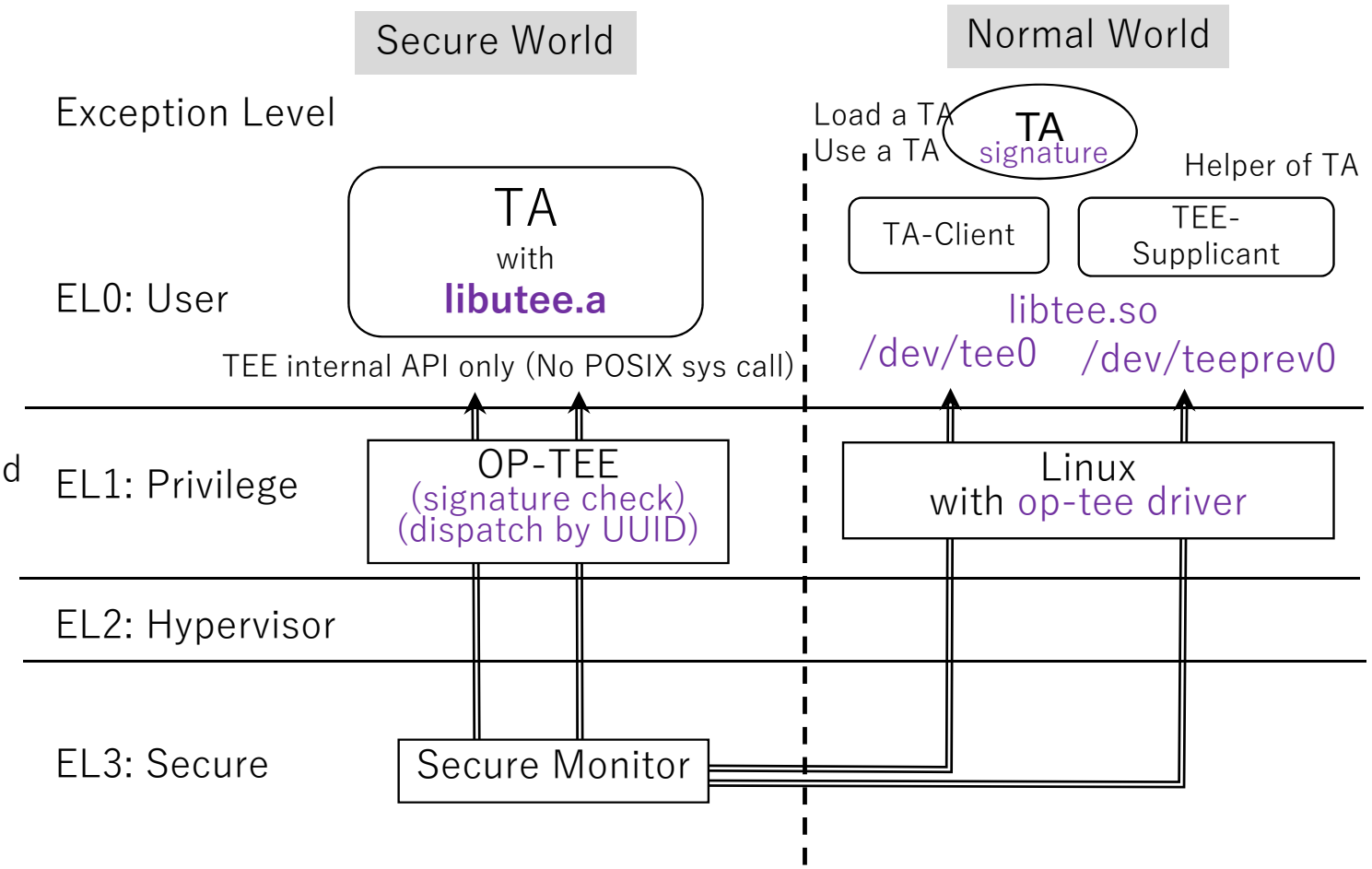
Implementation of Trusted OS

- Open Source Trusted OS
  - OP-TEE (Linaro) https://github.com/OP-TEE
  - Open-TEE (Aalto University[TrustCom15]) https://open-tee.github.io/
  - Trusty (Google) https://source.android.com/security/trusty/index.html
  - SierraTEE (Sierra) https://www.sierraware.com/open-source-ARM-TrustZone.html
  - SafeG (Nagoya University) https://www.toppers.jp/en/safeg.html
- Enterprise Trusted OS
  - Apple's Secure Enclave
  - Qualcomm's QTEE, ex. QSEE https://www.qualcomm.com/solutions/mobile-computing/features/security
  - Samsung's Knox https://www.samsungknox.com/en
  - Samsung's Teegris http://developer.samsung.com/teegris
  - Trustonic's Kinibi OS, ex. Mobicore/t-base/G&D
  - Huawei's TrustedCore

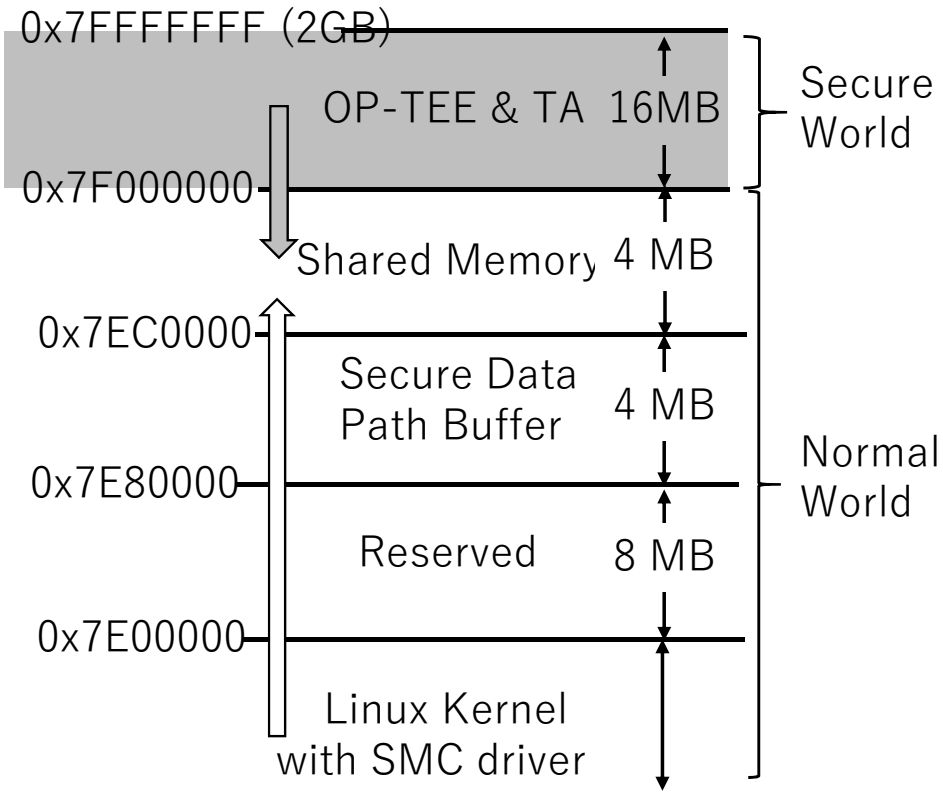
How to run a TA on OP-TEE

Major Roll

- TA-Client loads a TA on OP-TEE
  - TA needs a signature to load on OP-TEE
  - TA has a UUID to communicate with TA-Client and TEE-Supplicant
- TA-Client sends a request to TA
- TA replies an answer to TA-Client
- TA sends a request to TEE-Supplicant (e.g., to save an encrypted data on File System)



Memory Map of OP-TEE

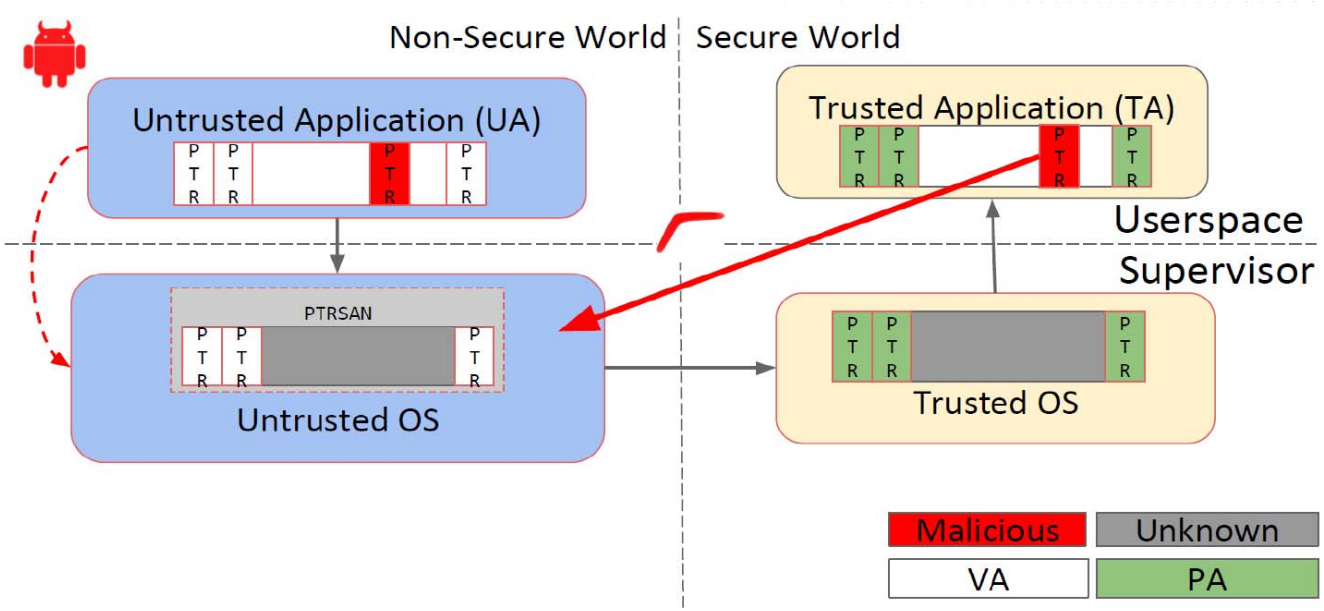


- ARM 96Board Hikey 2GB
  - SoC: Kirin 620
  - Cortex-A53 Octa-core 64-bit 1.2GHz (ARM v8 instruction set)
- Software size: Our experience
  - Secure world
    - Secure Monitor 33KB
    - OP-TEE 281KB
    - TA 1,200KB
  - Normal World (on Linux)
    - TA-Client 17KB
    - TEE-Supplicant 197KB

TEE Vulnerabilities

- Many attacks exist on software and hardware.
  - Software
    - **Boomerang [NDSS'17]**
      - Pointer exploit. TA can access any memory region. Attacker exploits TA to get sensitive data on normal world.
    - QSEE TrustZone Kernel Integer Overflow [BlackHat14]
    - Exploiting Trustzone on Android [BlackHat15]
  - Hardware
    - Foreshadow [USENIX Sec'18] (aka L1TF: L1 Terminal Fault)
      - Intel SGX Vulnerability of out-of-order execution
      - Microcode update mitigate this vulnerability
    - Prime+Count [ACSAC'18] by Samsung
      - ARM Trust Zone Cross-world Covert Channels on using cache.
    - Cache Attack [EuroSec'17] https://www1.cs.fau.de/sgx-timing
      - Intel SGX Cache Timing Attack

Boomerang Flaw[NDSS18]



TEE name	Vendor	Impact	Bug Detail
TrustedCore	Huawei	Arbitrary write	CVE-2016-8762
QSEE	Qualcomm	Arbitrary write	CVE-2016-5349
Trustonic	As used by Samsung	Arbitrary write	
SierraTEE	Sierraware	Arbitrary write	
OP-TEE	Linaro	Write to other application's memory	

*Security issues with ARM TrustZone [TestingStage18]

HIEE on RISC-V

- SMM: System Management Mode
  - Used by BIOS/UEFI for ACPI, etc.
- Intel's ME: Management Engine.
  - Run MINIX. Used for remote wakeup.
- Intel SGX
- ARM TrustZone

On RISC-V

⇒ Machine Mode

⇒ ???

⇒ ・ Sanctum of MIT
  ・ Keystone of UCB
⇒ ・ MultiZone of Hex-Fife
  ・ TEE WG of RISC-V Foundation

They are not programmable for a user.

They are programmable for a user.
They are used for TEE.

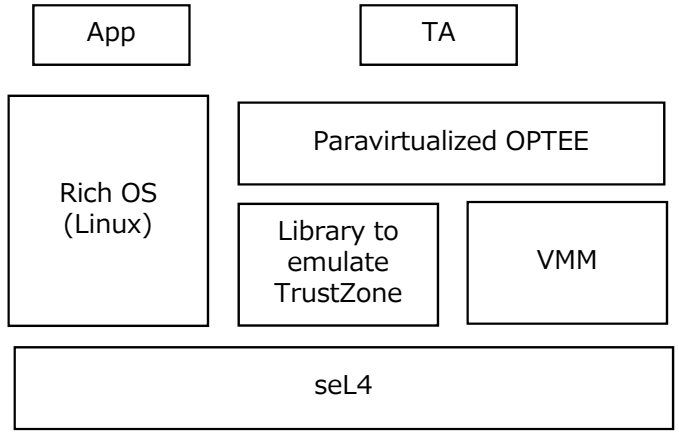
RISC-V TEE project

- Rahul Mahadev's OP-TEE on seL4 [Google Summer of Code 2016]
- Sanctum [USENIX Sec'16]
  - KeyStone [2018]
- MultiZone of Hex-Five[Sep/2018]
- TERP of SiFive[RISC-V Summit Dec/2018]
- TEE Working Group of RISC-V foundation

OP-TEE on RISC-V using seL4

- Rahul Mahadev's Google Summer of Code 16
- http://mahadevrahul.blogspot.com/

  - The TrustZone features and secure monitor must be implemented as a seL4 library.
  - OPTEE is paravirtualized, all calls referencing ARM Trusted Firmware and secure monitor are replaced with new calls.



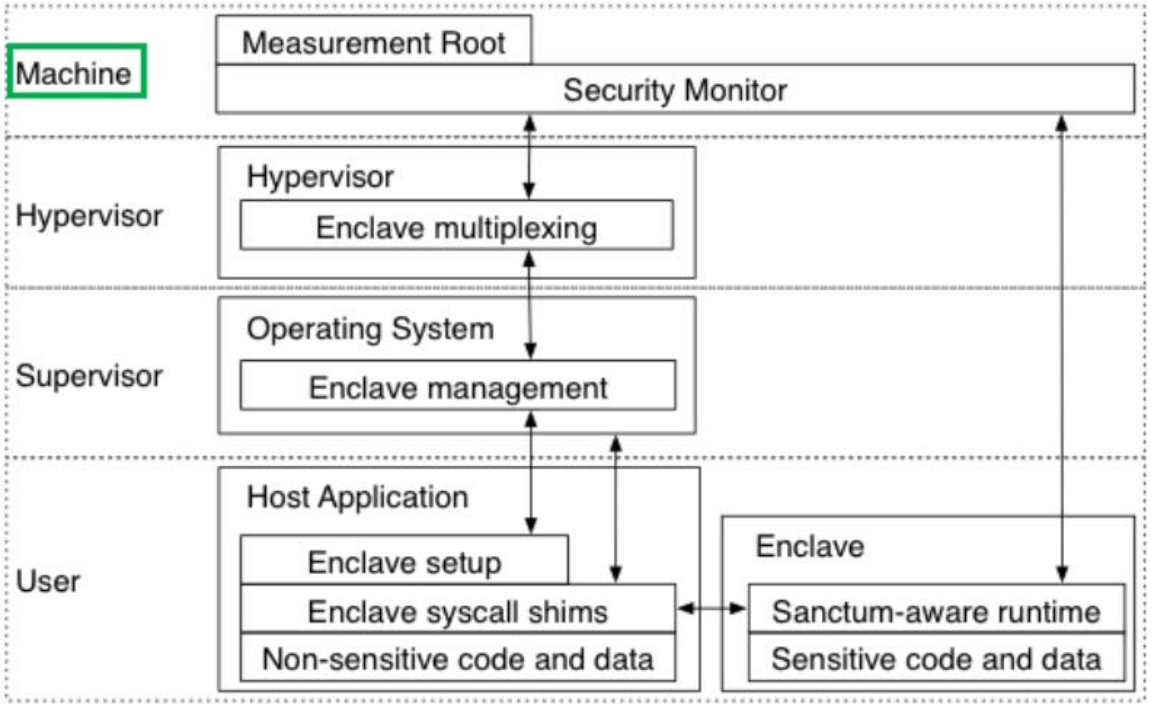
Sanctum [USENIX Sec'16]

- Figure of software stack

https://www.usenix.org/sites/default/files/conference/protected-files/security16_slides_costan.pdf

- Enclave is created o User Mode.
- Secure Monitor on Machine mode helps the secure creation of enclave.

- Successor project "KeyStone" of UCB and MIT.
- https://keystone-enclave.org/



MultiZone of Hex-Five

- MultiZone is announced


The Register
Biting the hand that feeds IT

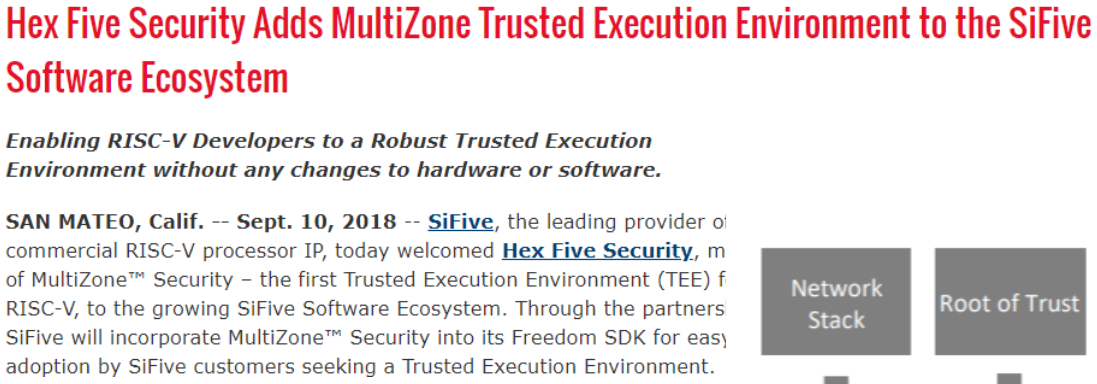
CENTRE SOFTWARE SECURITY DEVOPS BUSINESS PERSONAL TECH SCIENCE

Security

Arms race: SiFive, Hex Five build code safe houses for RISC-V chips

Those developing custom CPUs can now tap a TrustZone-ish trusted execution environment

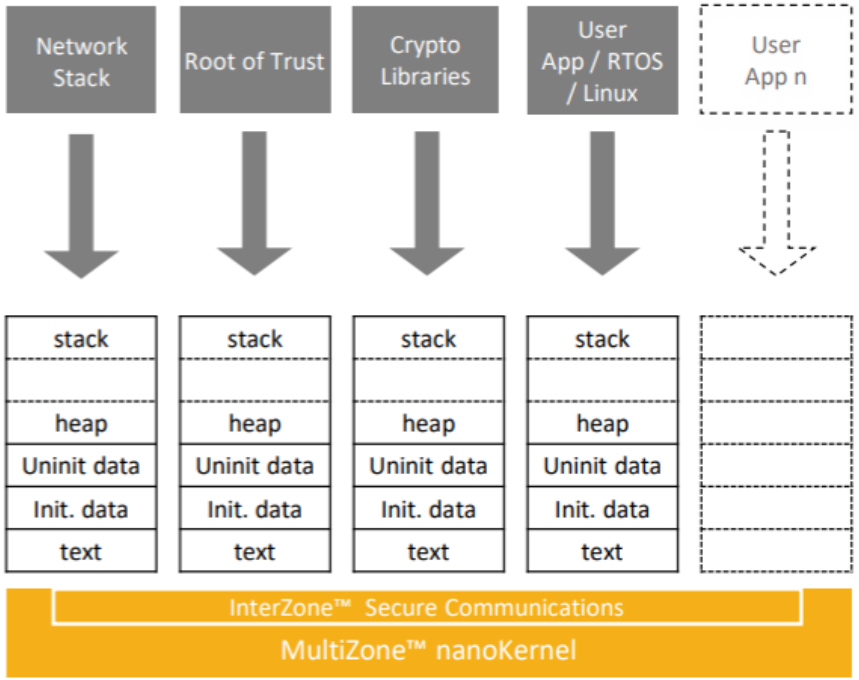
By Thomas Claburn in San Francisco 10 Sep 2018 at 20:08 2 SHARE


Hex Five Security Adds MultiZone Trusted Execution Environment to the SiFive Software Ecosystem

Enabling RISC-V Developers to a Robust Trusted Execution Environment without any changes to hardware or software.

SAN MATEO, Calif. -- Sept. 10, 2018 -- SiFive, the leading provider o commercial RISC-V processor IP, today welcomed Hex Five Security, m of MultiZone™ Security – the first Trusted Execution Environment (TEE) f RISC-V, to the growing SiFive Software Ecosystem. Through the partners SiFive will incorporate MultiZone™ Security into its Freedom SDK for easy adoption by SiFive customers seeking a Trusted Execution Environment.

- MultiZone is based on nanokernel.
  - https://hex-five.com/wp-content/uploads/2018/09/hex_five_multizone_datasheet.20180920.pdf
  - System Requirements
    - 32 bit or 64 bit RISC-V ISA with 'S' or 'U' extensions
    - Physical Memory Protection compliant with Ver. 1.10
    - 4KB FLASH and 1KB RAM



SiFive TERP: A Trusted Execution Reference Platform for Embedded Secure Applications

- The goal of TERP is to describe all the components necessary to build an embedded RISC-V processor which provides isolated multi-tenancy.

- It will be open at **RISC-V Summit 5/Dec/2018**.

TEE Working Group of RISC-V foundation

- Remote conference every week
  - Discuss memory protection, privilege mode, etc.

- When we implement OP-TEE on RISC-V, we must develop
  - Boot sequence: Trusted Boot Firmware, Secure Monitor
  - Linux kernel driver
  - Libraries (libutee.a for TA and libtee.so for Linux Apps)
  - Linux application to assist TA (TEE-supplicant)

Other Implementation of TEE

- Hardware
  - FPGA TEE “Iso-X” (SUNY at Binghamton) [Micro47 2014]
  - GPU TEE “Graviton” (Microsoft Research) [OSDI’18]
    - Requires NVIDIA GPU extension
- Software
  - TrustZone virtualization “vTZ” (Shanghai Jiao Tong University) [USENIX Sec’17]
    - Virtualize TrustZone for VMs
  - TEE delegation “DelegaTEE” (ETH Zurich) [USENIX Sec’18]
    - DelegaTEE is implemented by Intel SGX
  - TEE Migration (INRIA) [IFIP WISTP’15]
    - privacy-preserving TEE profile migration protocol

IETF's TEEP

- Trusted Execution Environment Provisioning
  - https://datatracker.ietf.org/wg/teep/about/
  - Protocol to manage TA: Trusted Application.
    - TAM(Trusted Application Manager) controls life cycle of TA (create, update, and delete).

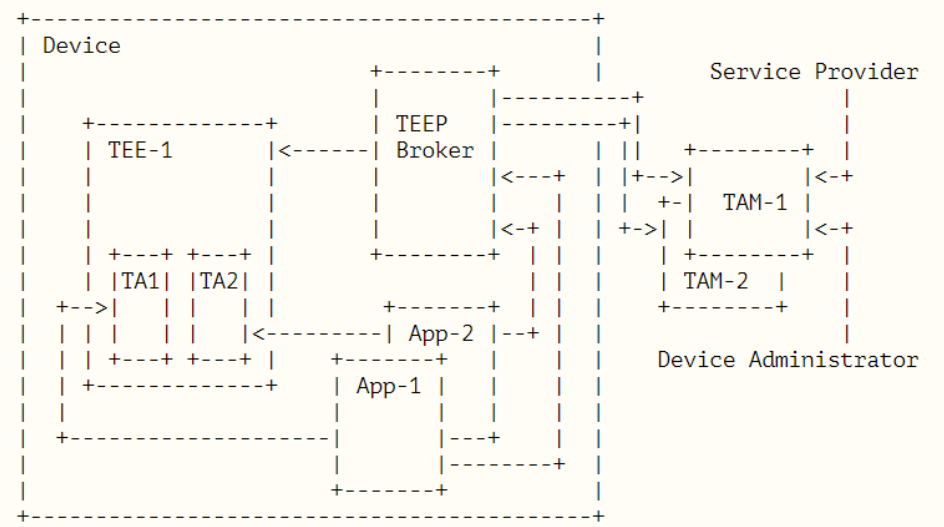


Figure 1: Notional Architecture of TEEP

    - TEE's API (Trusted OS) is important.

Survey papers

- Secure Processors Part I: Background, Taxonomy for Secure Enclaves and Intel SGX Architecture [2017, **Srinivas Devadas**]
- SoK：A Study of Using Hardware-assisted Isolated Execution Environments for Security[HASP16]
- Security issues with ARM TrustZone [TestingStage18]

- Trusted Execution Environment: What It is, and What it is Not [TrustCom15]
- TrustZone Explained: Architectural Features and Use Cases [CIC16]

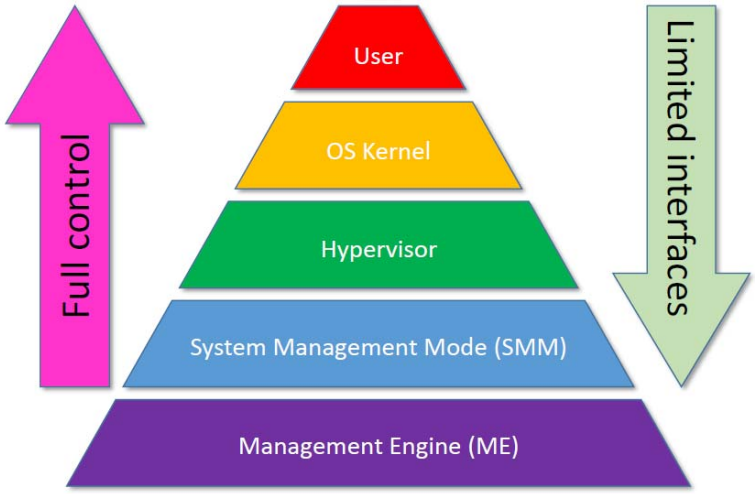
日本語参考

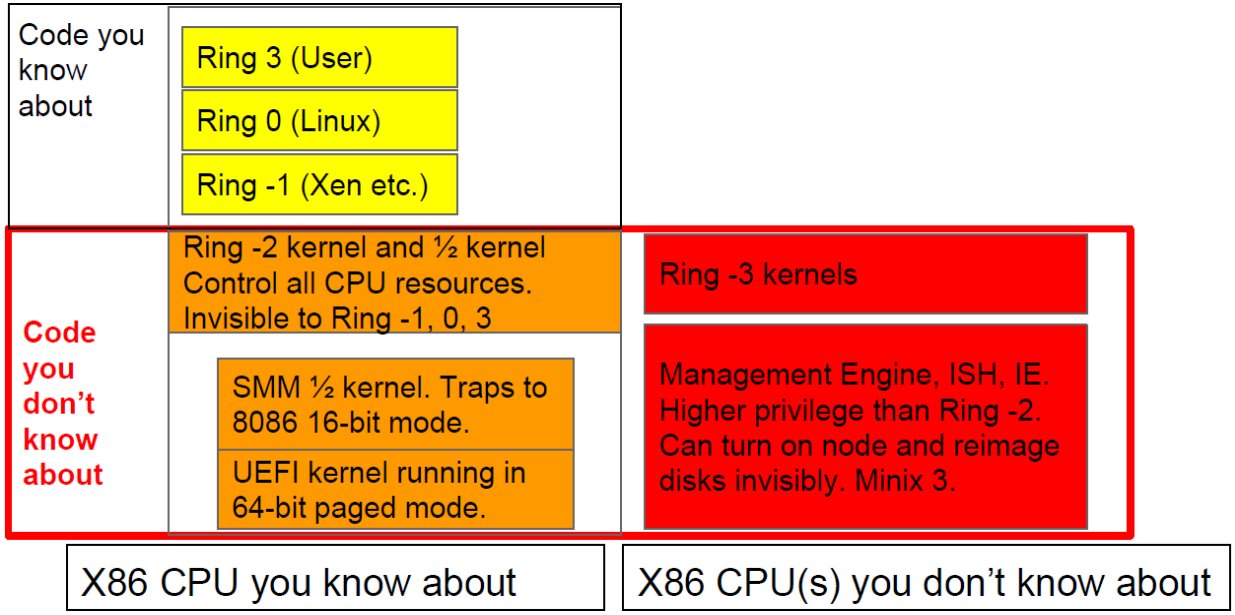
- FFRI Monthly Research 「ARMv8-M TrustZone：組み込みデバイス向けアーキテクチャとセキュリティ機能」
  - https://www.ffri.jp/blog/2016/03/2016-03-18.htm
- セキュアハードウェアの登場とその分析
  - https://www.ffri.jp/assets/files/monthly_research/MR201303_TrustZone.pdf
- TrustZone のユースケースと動向
  - https://www.ffri.jp/assets/files/monthly_research/MR201703_TrustZone_use_case_and_trend_JPN.pdf

Intel ME (Management Engine)

- Micro controller on chipset
- MINIX runs
  - HTTPS server runs
- Intel AMT (Active Management Technology)
  - Remote boot which works as IPMI
- Update with BIOS (Size info https://github.com/corna/me_cleaner)
  - Generation 2 (Nehalem-Broadwell, ME version 6 -10)
    - 1.5 MB (non-AMT firmware) 5 MB (AMT firmware)
  - Generation 3 (from Skylake onwards, ME version >= 11)
    - 2 MB (non-AMT firmware) 7 MB (AMT firmware)

Position of Intel ME

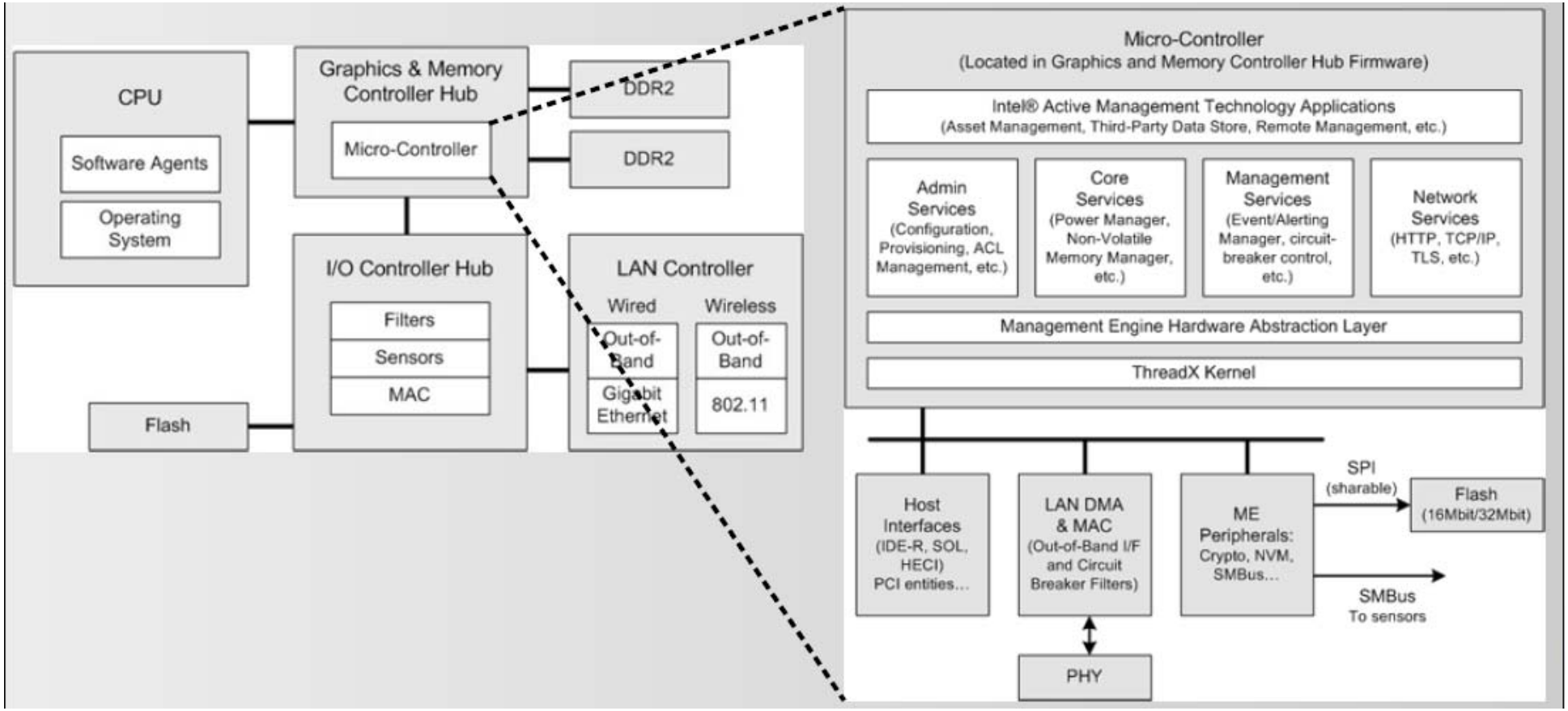




Intel ME: The Way of the Static Analysis [TROOPERS17]

Replace your exploit-ridden firmware with a Linux kernel [LinuxCon17]

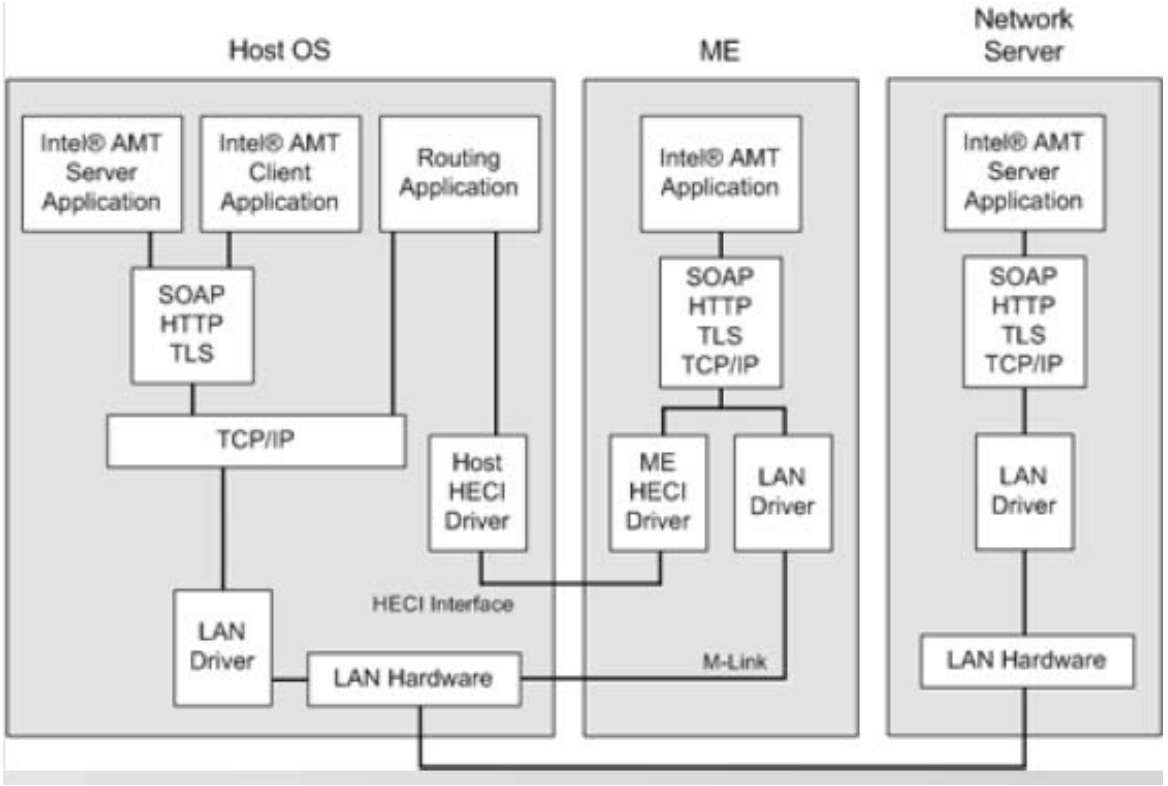
Overview of Intel ME



- https://itsfoss.com/fact-intel-minix-case/

Igor Skochinsky, Intel ME Secrets, CODE BLUE 2014

Network of Intel ME



- HECI: Host Embedded Controller Interface
  - communication using a PCI memory-mapped area
- Network protocol is SOAP(HTTP or HTTPS)

Igor Skochinsky, Intel ME Secrets, CODE BLUE 2014

Vulnerability of Intel ME

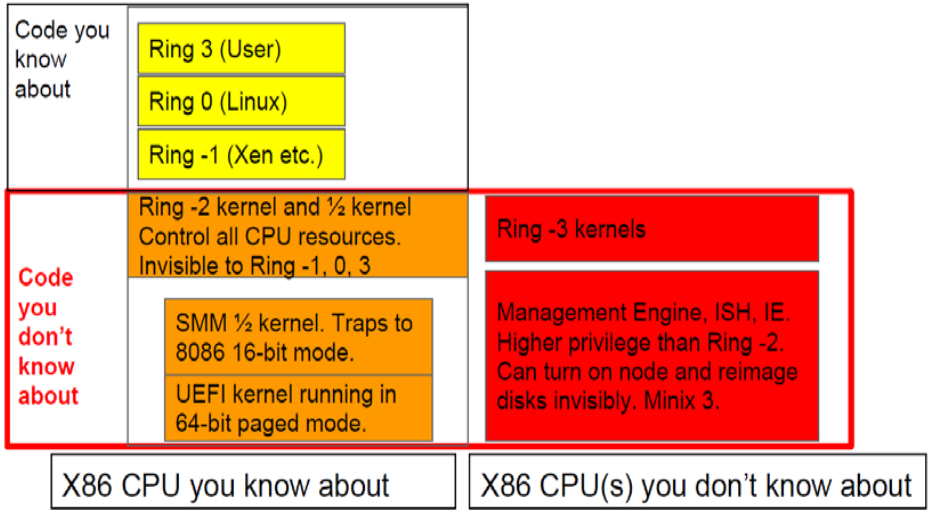
- Intel ME Manufacturing Mode: obscured dangers and their relationship to Apple MacBook vulnerability, CVE-2018-4251
  - http://blog.ptsecurity.com/2018/10/intel-me-manufacturing-mode-macbook.html
- Intel ME 11.x Firmware Images Unpacker
  - https://github.com/ptresearch/unME11
- Vulnerability INTEL-SA-00086 allows to activate JTAG for Intel Management Engine core.
  - https://github.com/ptresearch/IntelTXE-PoC
- ”Silent Bob is Silent”, Escalation of privilege vulnerability on Intel AMT, CVE-2017-5689

Google's stance

- Ring2-3 has many functions (in MINIX)
  - IP stacks (4 and 6)
  - File systems
  - Drivers (disk, net, USB, mouse)
  - Web servers

They work even if the main OS is terminated.

They increase attack surface.



Replace your exploit-ridden firmware with a Linux kernel [LinuxCon17]

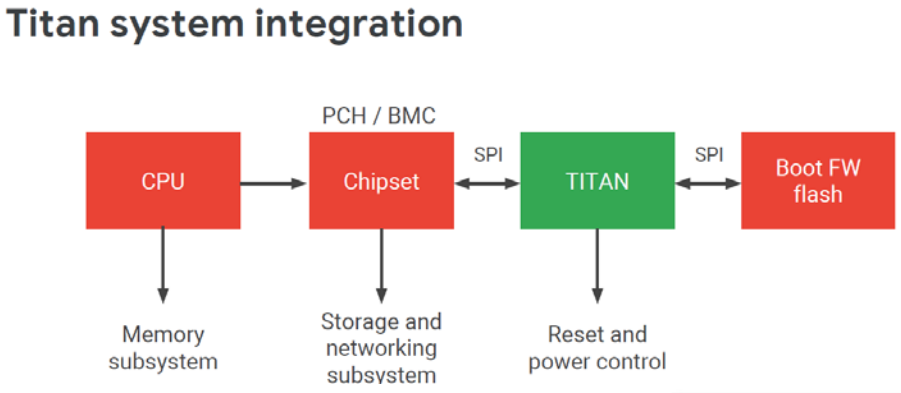
Googles Answer
NERF: Non-Extensible Reduce Firmware

- De-blobbed ME
- UEFI reduced to its most basic parts
- SMM disabled or vectored to Linux
- Userland written in Go (http://u-root.tk)
  - u-root [USENIX'15]

Replace your exploit-ridden firmware with a Linux kernel [LinuxCon17]

Google's Titan

- Google proposes "secure chip" which integrated between chipset and boot flash.



Titan: enabling a transparent silicon root of trust for Cloud [HotChips18]

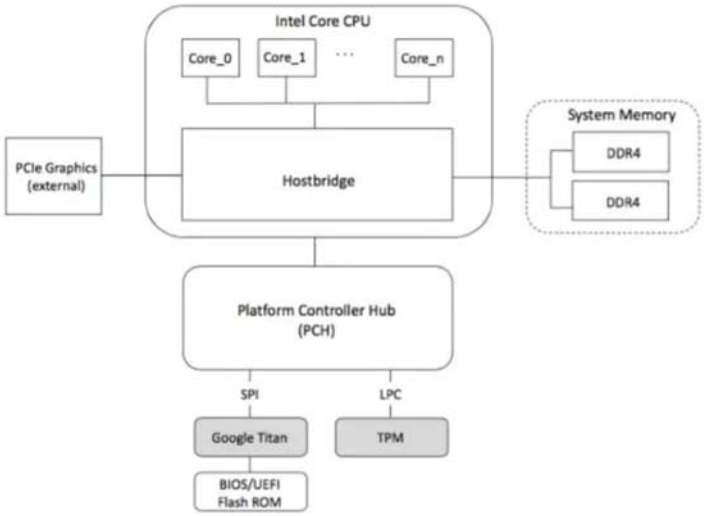


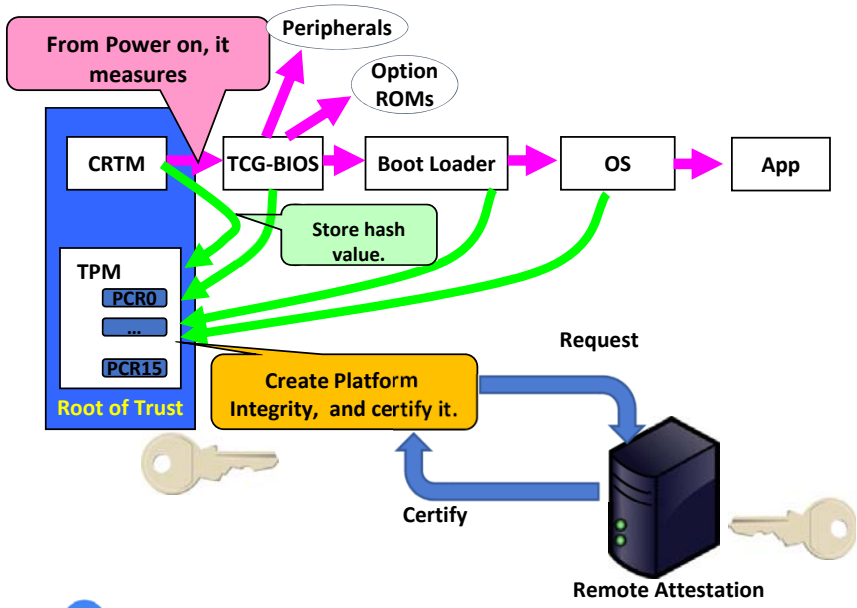
Table 1: Comparisons between Titan and TPM.

		Titan	TPM
Topological Location		On SPI bus. Between PCH/BMC and the boot firmware (BIOS/UEFI) flash	Either on LPC bus or SPI bus
Hardware Features	Application Processor	Yes	Yes
	Cryptographic Co-processor	Yes	Yes
	Random Number Generator	Yes	Yes
	SRAM	Yes	No
	Non-volatile Memory	Yes	Yes
	Read-only Memory	Yes	Yes
	Monotonic Counter	Yes	Yes
	PCRs	Maybe	Yes
Security Features	Secure Boot	Yes	Yes
	Remote Attestation	Maybe	Yes
	Remediation	Yes	No
	First-instruction Integrity	Yes	No

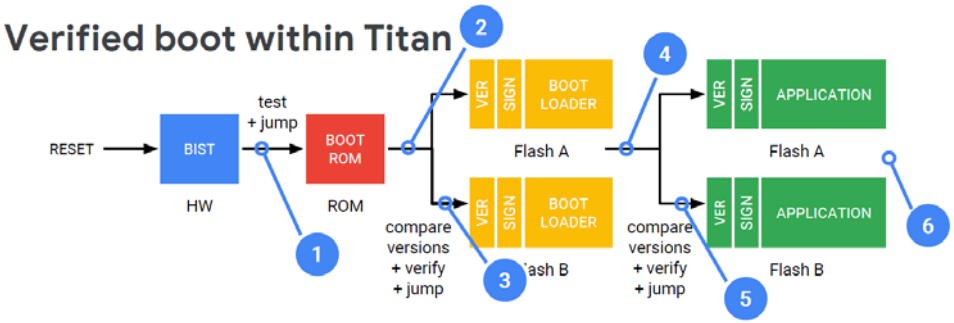
- It resembles to TPM.
  - "Towards Understanding Google Titan"
  - https://harry.uno/post/google-titan.html

TPM v.s. Google's Titan

- TPM's trusted boot (measured boot)
  - No mechanism to stop



- Titan's verified boot
  - Stop if verification fails

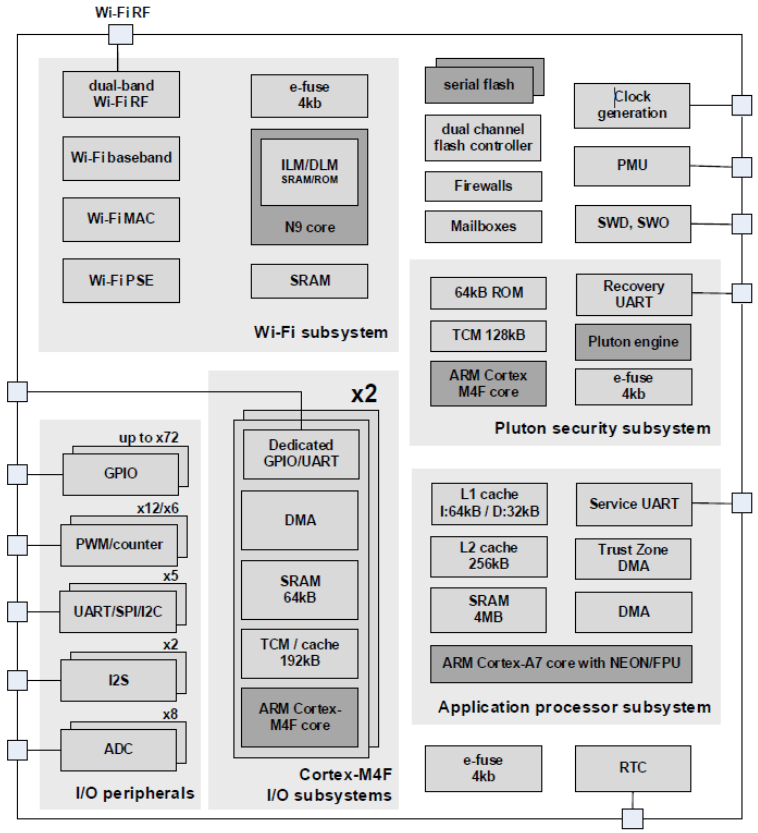


1. Test logic (LBIST) and ROM (MBIST); if fail ⇒ stay in reset; else jump to ROM
2. Compare bootloader (BL) versions A + B; choose most recent
3. Verify BL signature; if fail, retry with other BL; if fail, freeze
4. Compare firmware application (FW) versions A + B; choose most recent
5. Verify FW signature; if fail, retry with other FW; if fail, freeze
6. Execute successfully verified FW

Titan: enabling a transparent silicon root of trust for Cloud [HotChips18]

MS Azure Pluton

- MediaTek MT3620The first Azure Sphere class Microcontroller
  - Securely isolated subsystems.
  - Units has HW firewalls.
  - HW based attestation
  - **Security processor is first to boot**
    - **Initial code is in ROM.**
  - **Software is signed.**
  - **SW rollback protection.**



The Hardware Security Platform Behind Azure Sphere [HotChips 18]

Conclusions

- Many CPU security faculties.
  - The common feature is “HIEE: Hardware-assisted Isolated Execution Environments”.

- Related Talks
  - 29/Nov 第7回サイバーセキュリティ国際シンポジウム@慶応大学
    - RISC-V Panel
    - https://cysec-lab.keio.ac.jp/sympo1811/index-j.html
  - 13/Dec ハードウェアセキュリティフォーラム2018　@東大
    - http://www.ieice.org/~hws/