第32回コンピュータシステム・シンポジウム（ComSys2020）招待講演
2020/12/1

TEE (Trusted Execution Environment)は第二の仮想化技術になるか？

須崎有康(1,2)

(1) セキュアオープンアーキテクチャ・エッジ基盤技術研究組合(TRASIO)

(2) 国立研究開発法人 産業技術総合研究所 (AIST)

サイバーフィジカルセキュリティ研究センター (CPSEC)

おことわり

- TEEや仮想化を中心に話します。Root of Trustやremote Attestationも面白い話題がありますが、今回は詳細を割愛させていただきます。

- 参考資料
  - Trusted Execution Environmentによるシステムの堅牢化, 情報処理2020/06
    https://ci.nii.ac.jp/naid/40022255769
  - Trusted Execution Environmentの実装とそれを支える技術,電子情報通信学会基礎・境界ソサイエティ Fundamentals Review, 2020/10
    https://www.jstage.jst.go.jp/article/essfr/14/2/14_107/_article/-char/ja/

研究背景

- TPM
  - 振興調整費「組込みシステム向け情報セキュリティ技術（研究代表者：柴山先生、2006-08）」
  - 日本IBMからの再委託(METI新世代情報セキュリティ研究開発事業、2007)
- ARM TrustZone
  - JST日台研究交流「偽造困難なデバイスを用いたIoTセキュリティ管理システム」(2015-17)
- RISC-V TEE
  - NEDO AIエッジ(FY2018-2022)でRISC-V TEEのソフトウェア開発を担当

    セキュアオープンアーキテクチャ・エッジ基盤技術研究組合 (TRASIO)が2019/08に設立

TPMによるRemote attestationを実現した。

ACSAC2020論文
Reboot-Oriented IoTに結実

今ここ

Open Source Summit Japan、
IEEE TrustCom論文に結実

Outline

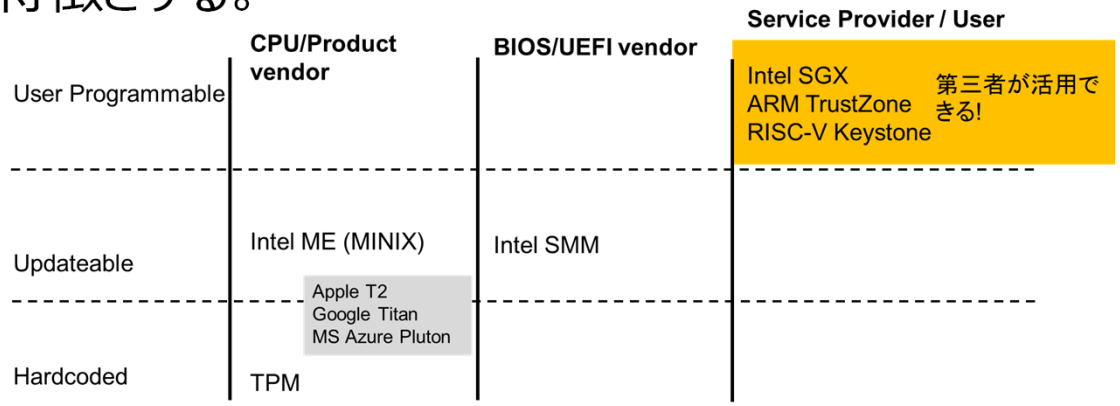
- TEEとは
  - ハードウェアの特長

    Arm TrustZone, Intel SGX, RISC-V Keystone(TRASIOのRISC-V TEE)
  - システムソフトの実装の違い
- 新しい方向
  - 仮想化対応

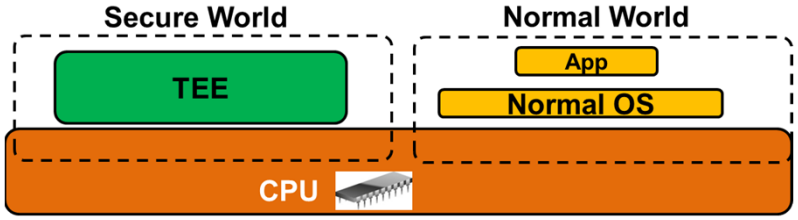
    Arm-v8.4A TrustZone, AMD SEV, Intel TDX, Amazon Nitro
- その他
  - アンチテーゼ
  - 関連組織、規格
- まとめ

TEEとは

- HIEE(Hardware-assisted Isolated Execution Environments)の一つだが、TEEは第三者がプログラミング可能であることを特徴とする。



- 特徴：（極端に言えば）隔離実行されるのみで、単体ではRoot of Trustになりえない。
  - Root of Trustには安全に鍵・証明書を保存する耐タンパハードウェア(Secure ElementかSecure CoProcessor)が必要
  - **これを信頼の基点に*Remote Attestation*が行われる**
- 利用できるハードウェア
  - ARM TrustZone, Intel SGX, RISC-V Keystone



この図はあくまでTEEの一例

TEEの応用

- 機密情報処理
  - 鍵管理
    AndroidのKeyMaster
  - DRM処理
    スマホのWidevine(Google)
  - 個人情報管理
    指紋認証処理
- コード・データの隠蔽
  - 機械学習の重み付けデータ
  - プライバシー保護
  - 遺伝子解析
- 暗号処理の前提条件軽減
  - IRON[CCS'17]

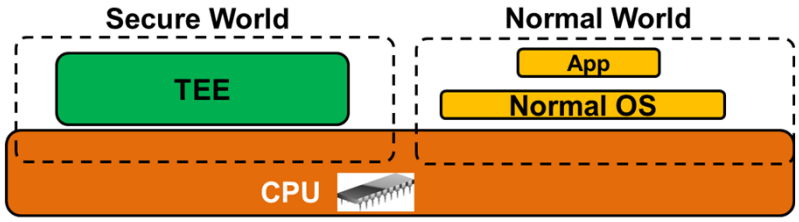
- メモリ消費が少ない
- スマートフォン
- Arm TrustZone向き

- メモリ消費が大きい
- サーバ・クラウド
- Intel SGX、AMD SEV 向き

- RISC-V?

TEE分類

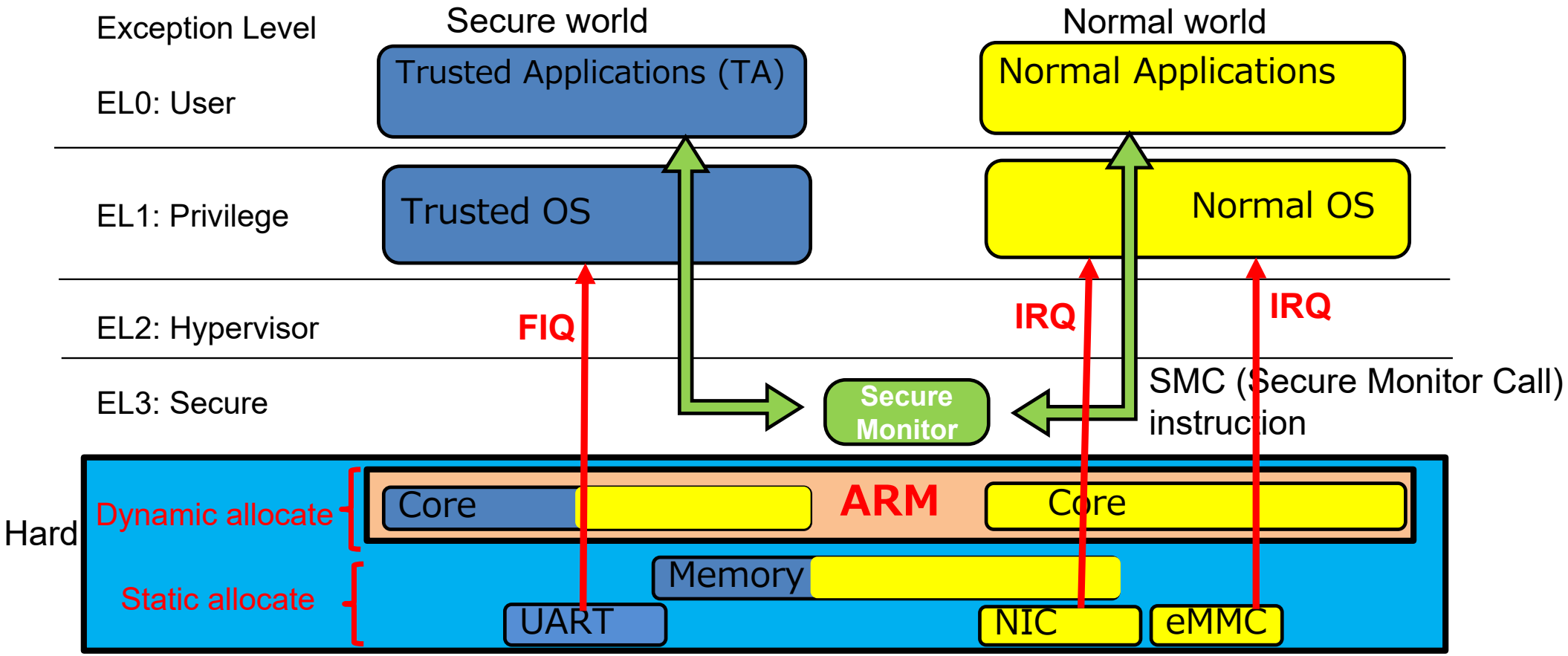
A Tale of Two Worlds[CCS19]での分類



- **2 World View Model**
  - Normal WorldとSecure Worldに分けられる。
  - 各Worldでユーザ・カーネルモードがある。
  - GlobalPlatformが定義しているアーキテクチャ
  - 対応アーキテクチャ　ARM TrustZone, RISC-V Keystone
- **Single Address Model**
  - ユーザ空間のプログラムから直接TEEのEnclaveに切り替わる。
  - 対応アーキテクチャ　Intel SGX

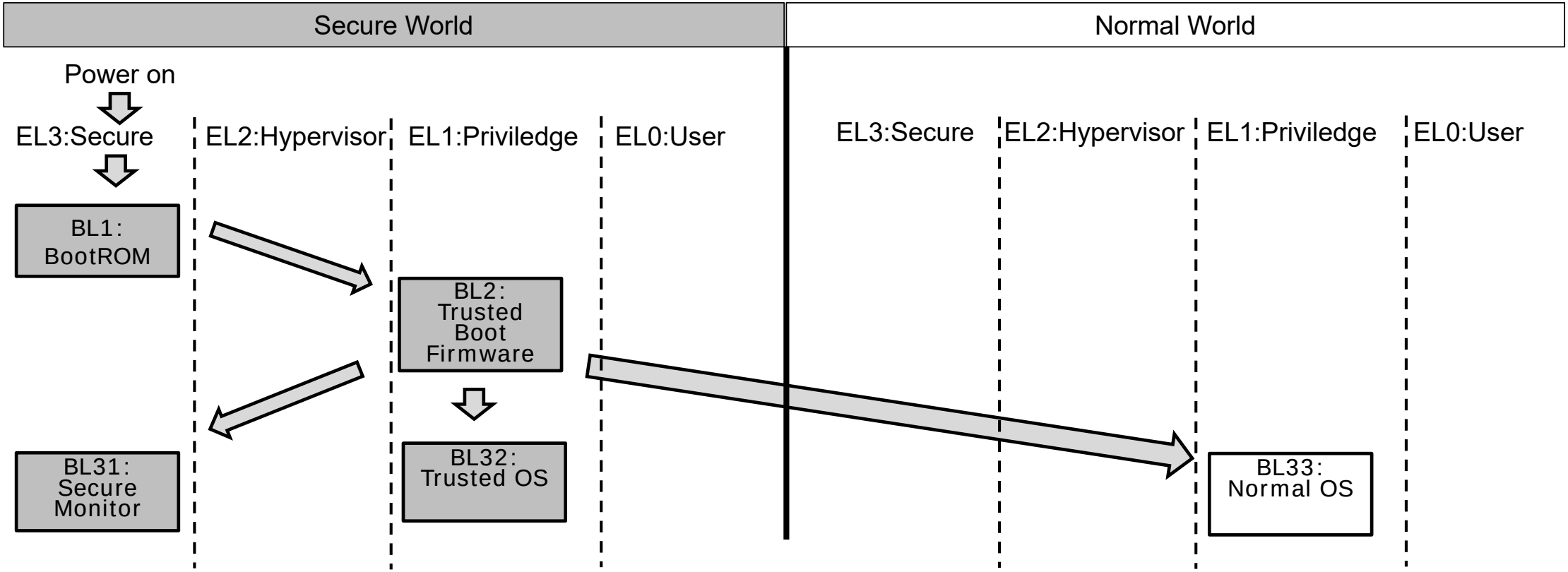
7

Arm Cortex-A TrustZone



Boot Sequence on ARM Trust Zone

- BL: Boot Loader
- EL: Exception Level



9

TRASIO

AIST CPSEC | Cyber Physical Security Research Center

Intel SGX

- TEEであるEnclave内はRing3のみ。ユーザアプリのみの実行ができる
  - Intel SDKに従えばUntrusted側のプロセスの一部のライブラリ関数として実行される
  - OS相当の機能は単純に入れられない。（苦労して入れている）
- SGX実装の大部分がMicro Code (詳細は特許、講演資料から推測 Secure Processors Part I & II)
  - 追加ハードは PMH(Page Miss Handler)とMEE (Memory Encryption Engine)
  - Miro Codeでは実装が難しい部分には特殊Enclaveがある

    Provisioning Enclave 鍵管理、Launch Enclave 生成管理、Quoting Enclave リモートアテステーション管理
- 使えるメモリは１Enclaveで96MB以下。（BIOSで確保は128MB以下）
  - 暗号化したメモリをEnclave外にswap outする仕組みあり

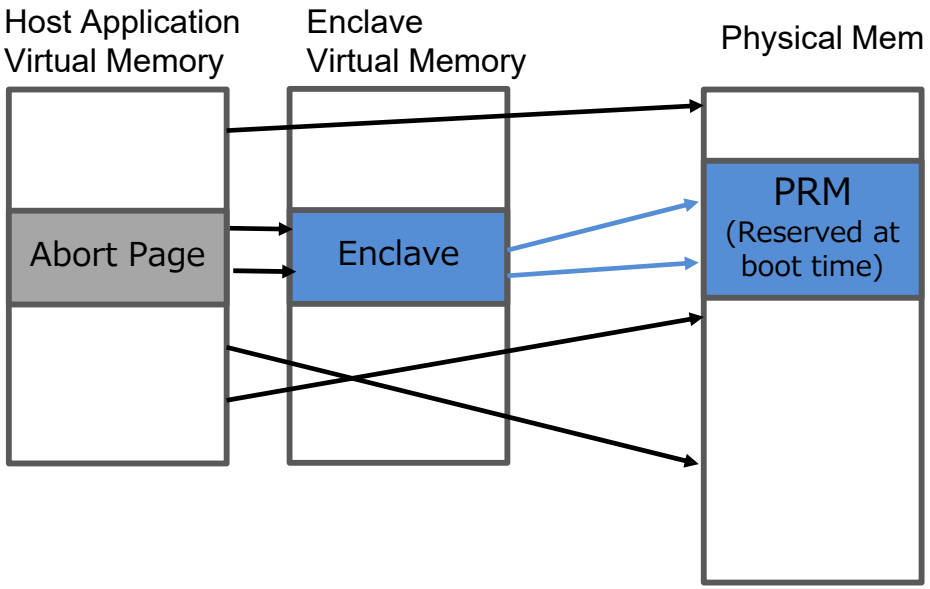
参考資料

Secure Processors Part I & II, Victor Costan, Ilia Lebedev and Srinivas Devadas **これはバイブル！**

https://people.csail.mit.edu/devadas/pubs/part_1.pdf https://people.csail.mit.edu/devadas/pubs/part_2.pdf

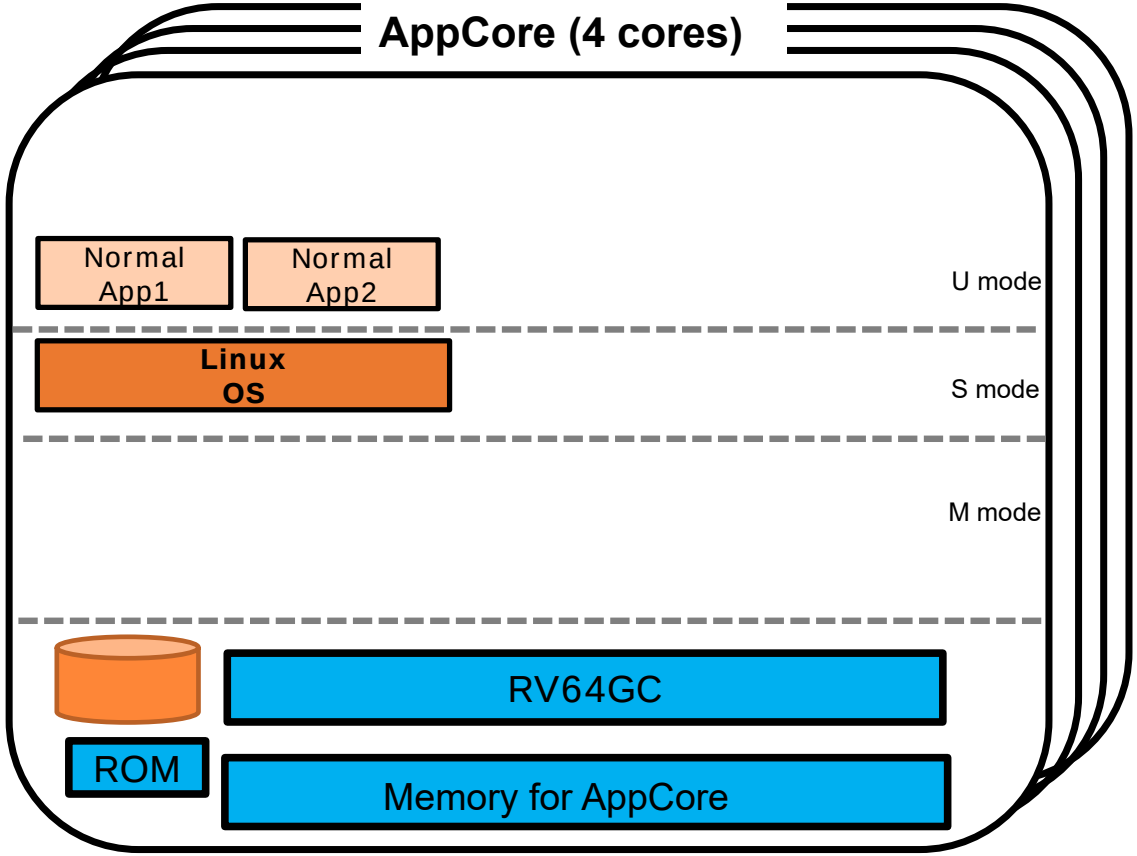
Intel SGX

- 通常のアプリの一部（ライブラリ関数）としてEnclaveで実行される
- Enclaveのメモリは別仮想空間となる。物理メモリとしてはEnclave用に確保した部分が使われ、暗号化される



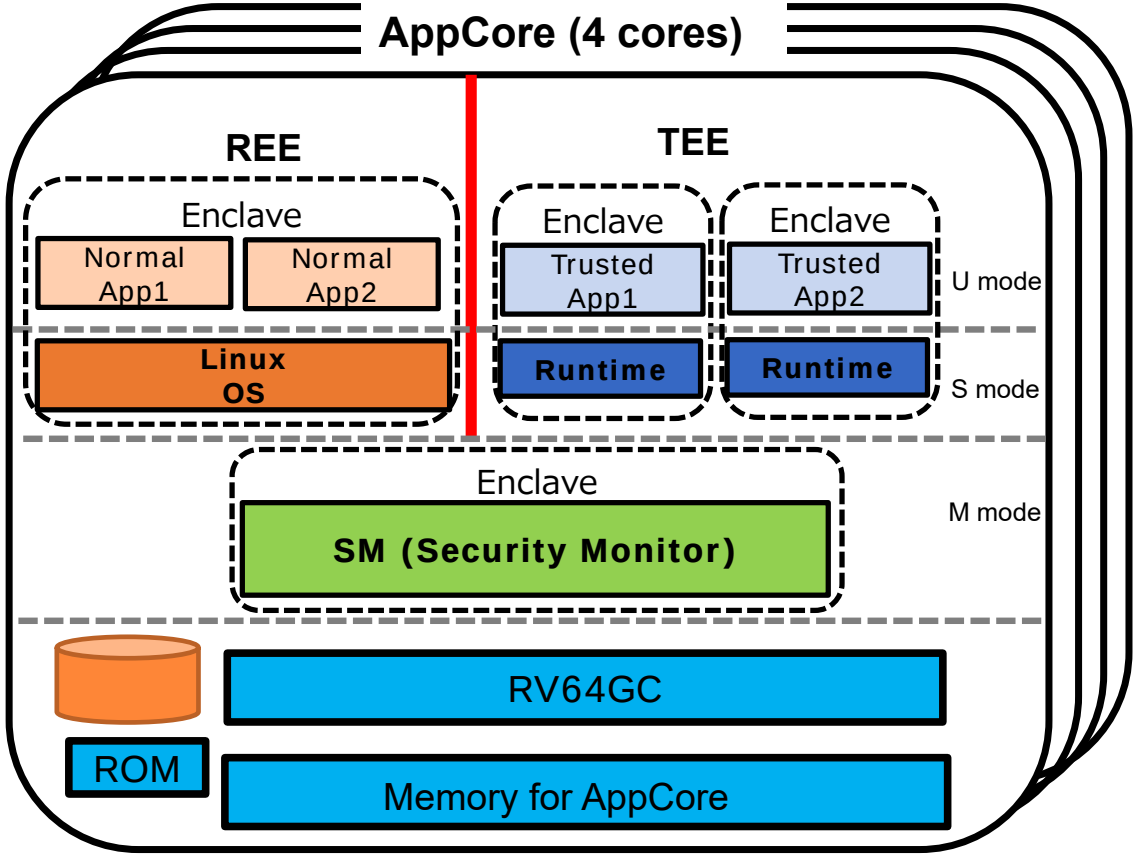
- PRM: Processor Reserved Memory　BIOSで確保するメモリ

通常のRISC-V



- Rocketコアを想定
  - 割込みはM modeに落ちるが図中では省略

Keystoneを有効にしたRISC-V

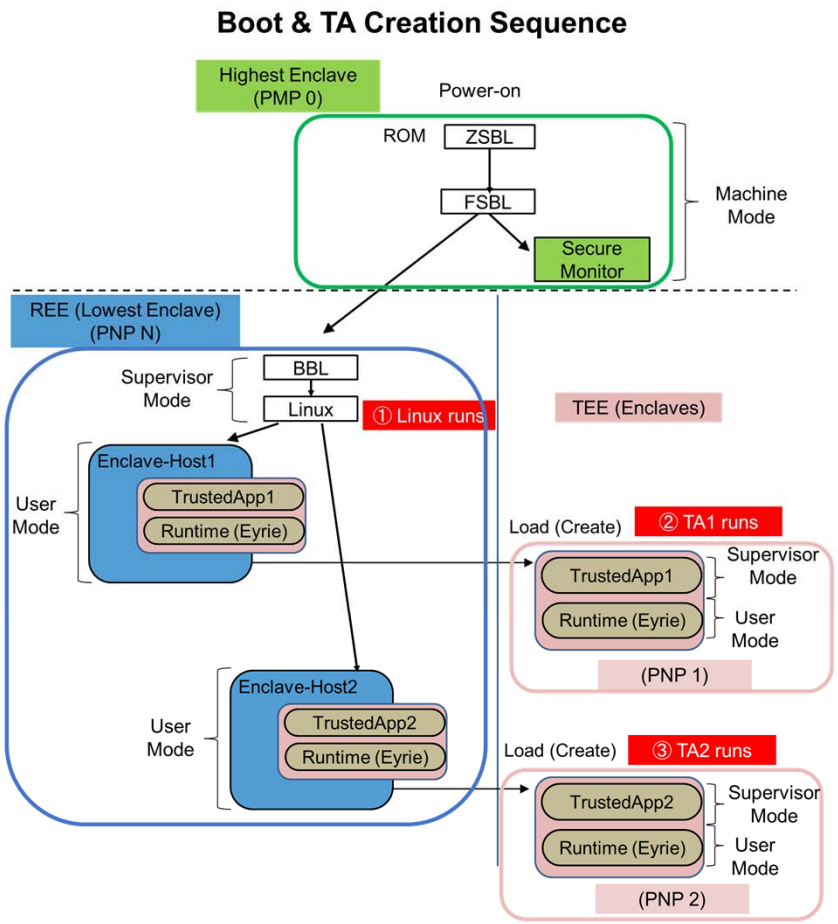


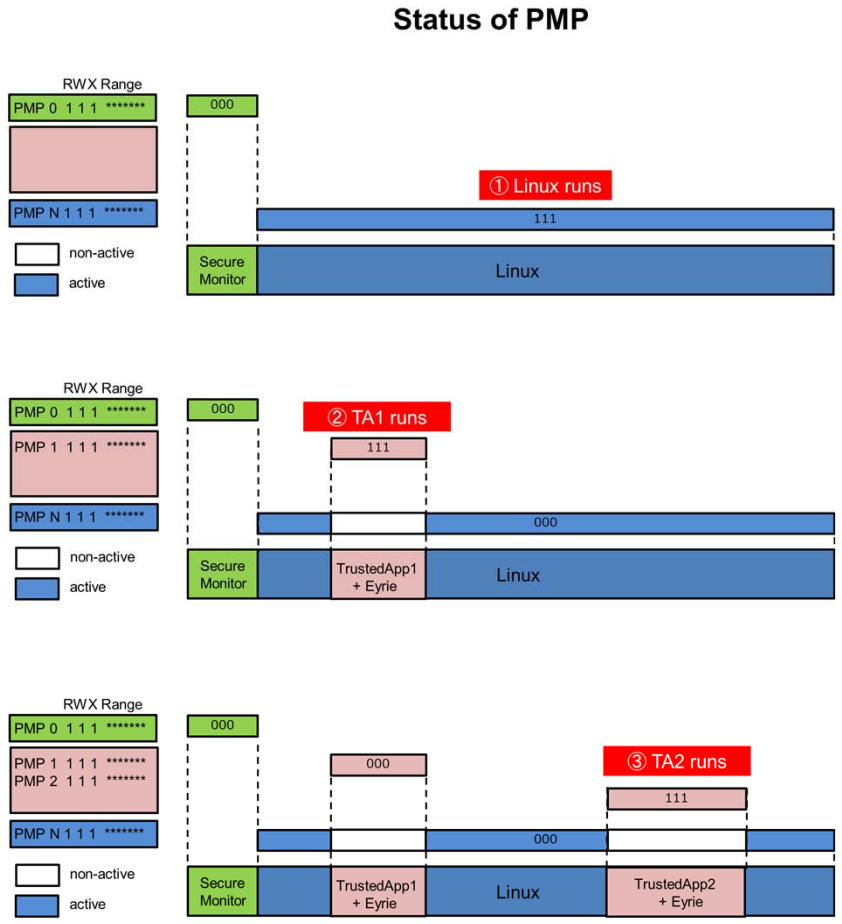
- ハードウェア的には変わりはない。
- PMP: Physical Memory Protectionを活用したメモリ分離
  - 図中の点線で囲われているEnclave単位で分離される
    - M modeのSMで一つ
    - REEのLinuxで一つ
    - TEE内に二つ

REE: Rich Execution Environment

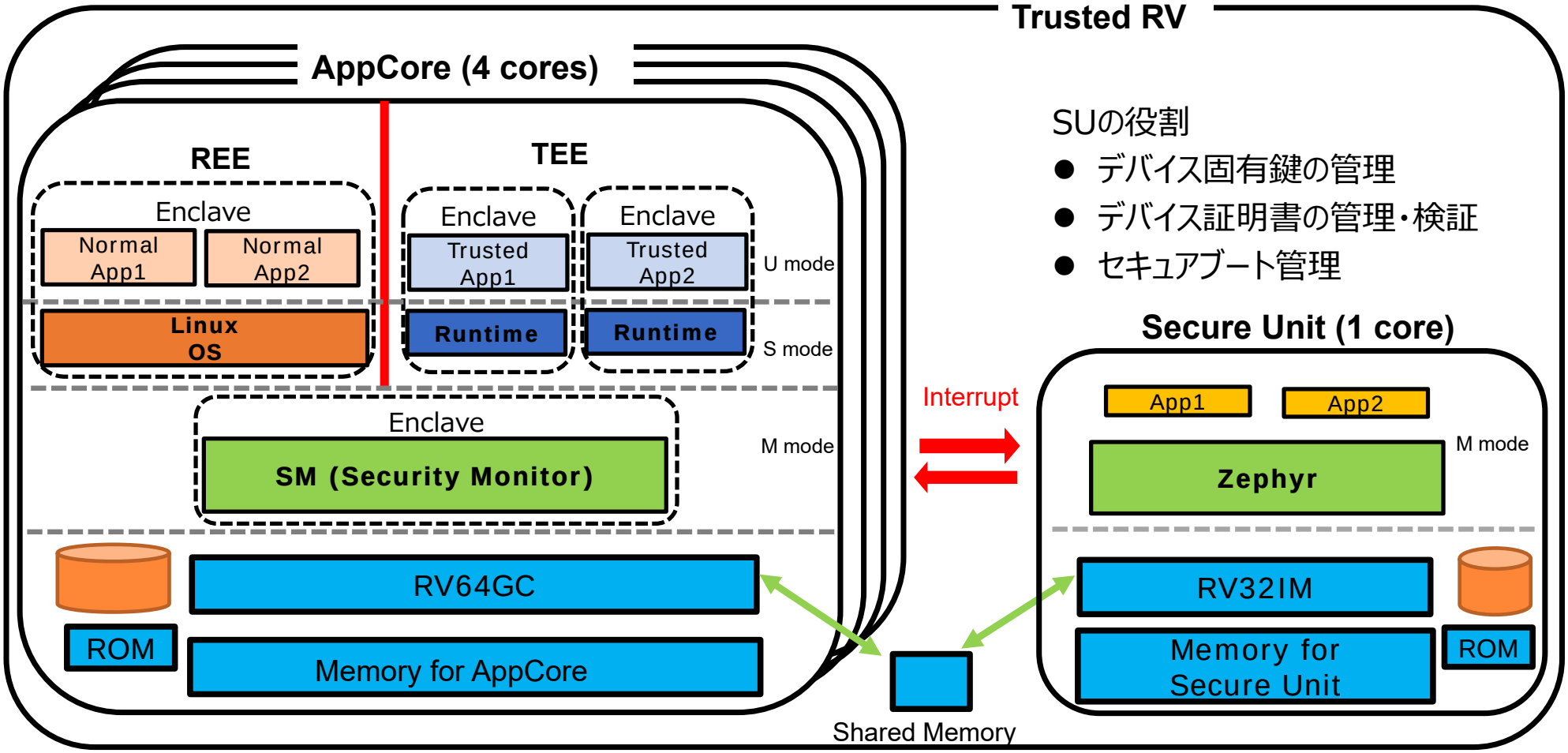
RISC-V Keystone

- Keystoneを有効にした場合の起動とPMPの動作





Trusted-RVプラットフォーム



TRASIOが提供する要素技術

- ハードウェア
  1. Trusted-RV プラットフォーム（64bit RISC-V + 32 bit RISC-V Secure CoProcessor)
- ソフトウェア
  2. TEEのプログラム開発環境　GlobalPlatform TEE Internal API
  3. TAの管理フレームワーク　TEEP(Trusted Execution Environment Provisioning)
  4. Remote Attestation

TEE上でシステムソフト実装

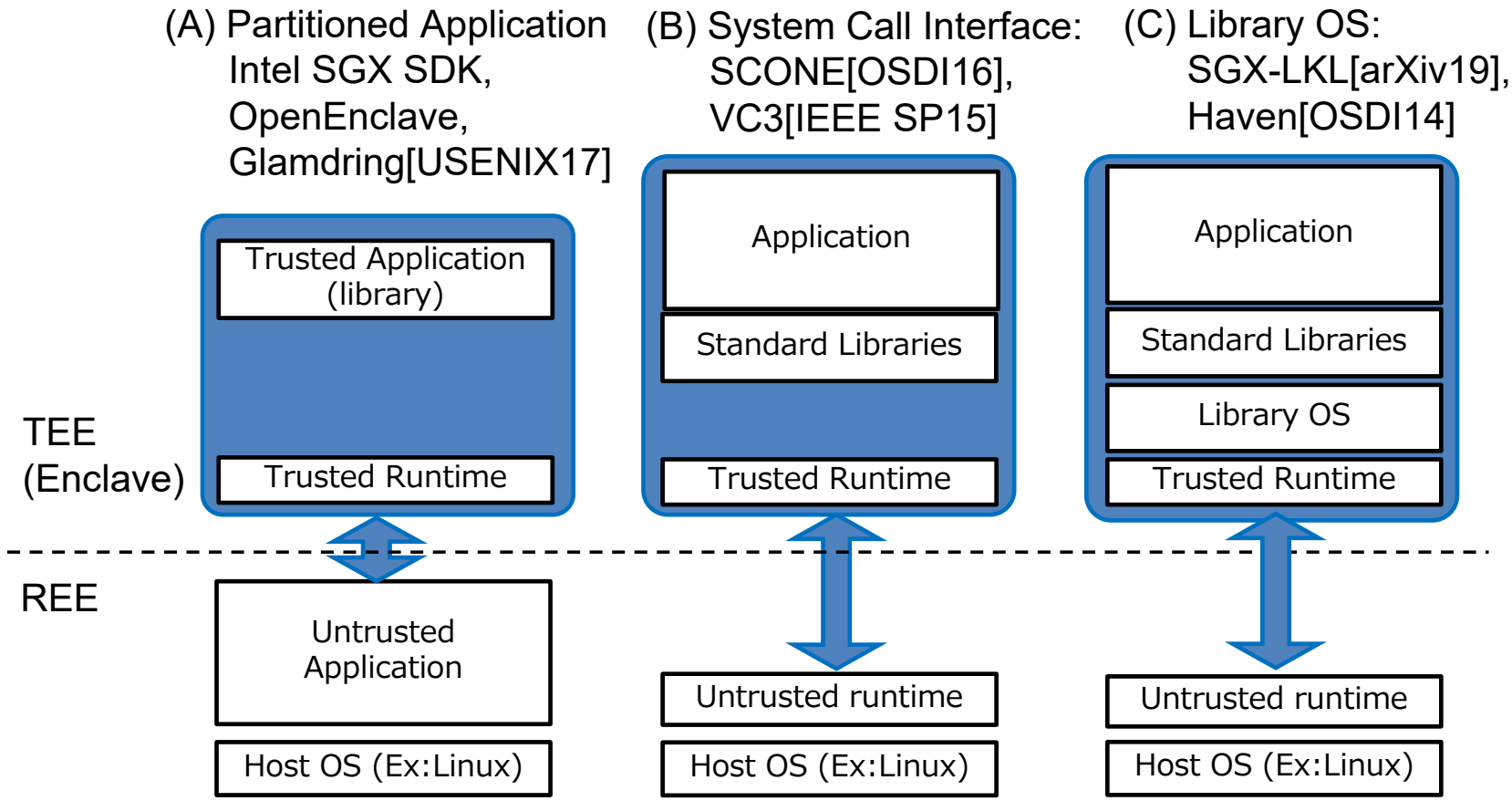
同一TEEハードウェアでもシステムソフト実装は**大きく異なる。**
これが引き起こす問題

Intel SGXのシステムソフト

- 代表的な実装
  - Intel SGX SDK
  - Haven[OSDI14]
  - SCONE[OSDI16]
  - Glamdring[USENIX17]
  - SGX-LKL (Linux Kernel Library) [arXiv19]
  - VC3[IEEE SP15]
  - OpenEnclave [Microsoft]

- 抽象化、インターフェースの切り方が異なる

SGX上での実装


(A) Partitioned Application
Intel SGX SDK,
OpenEnclave,
Glamdring[USENIX17]
(B) System Call Interface:
SCONE[OSDI16],
VC3[IEEE SP15]
(C) Library OS:
SGX-LKL[arXiv19],
Haven[OSDI14]
TEE
(Enclave)
REE
Trusted Application
(library)
Trusted Runtime
Application
Standard Libraries
Trusted Runtime
Application
Standard Libraries
Library OS
Trusted Runtime
Untrusted
Application
Host OS (Ex:Linux)
Untrusted runtime
Host OS (Ex:Linux)
Untrusted runtime
Host OS (Ex:Linux)

TRASIO

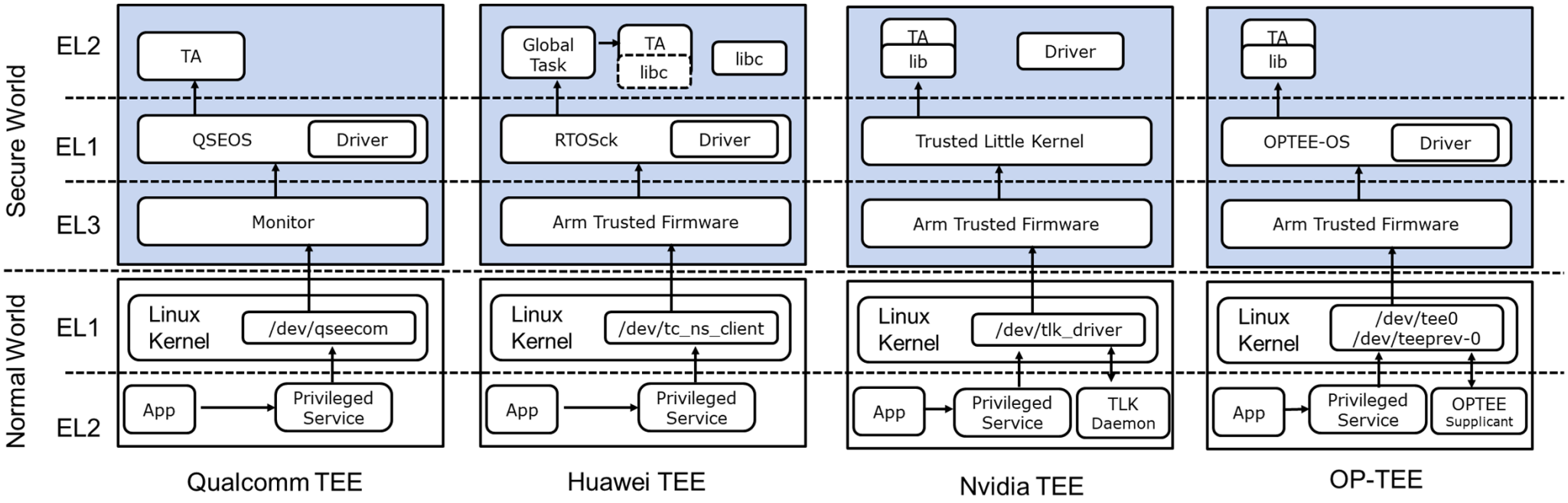
AIST CPSEC | Cyber Physical Security Research Center

Arm TrustZone上の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
- 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
  - Nvidia's TLK (Trusted Little Kernel)

Arm Cortex-A TrustZone の実装

- Secure WorldのTAのライブラリ実装(Dynamic/Static),Driverの実装などが異なる。
- Normal WorldのDaemon実装やdeviceインターフェースが異なる。



出典：SoK: Understanding the Prevailing Security Vulnerabilities in TrustZone-assisted TEE Systems [IEEE SP20]より

多様なTEEシステムソフトの問題点

- TEEを使いたいサービス提供者側に対する制約
  - TEEではユーザアプリやTrusted AppもTEEシステムソフトの合わせて作る必要がある。
    - 私見：完全仮想化の様にVMに乗るOSが既存のものが変更なく使えれば問題はない。Dockerもカーネルとユーザ空間と言う既存のインターフェースを上手に切って成功した。逆にLibrayOSは切り方の制約(すべてのSysCallが提供できないなど)がアプリにまで及び成功していない。

- 【現状】色々なTEE実装があると対応しきれない。
  - LINE Dev Day 2020のCross-platform Mobile Security at LINEの話

    TEEの有用性は理解できるが、デバイスやOSバージョンによってTrusted Applicationの実装方法が異なるのですべてに対応できない。

    このため、LINEではWhitebox Cryptographyを使う。

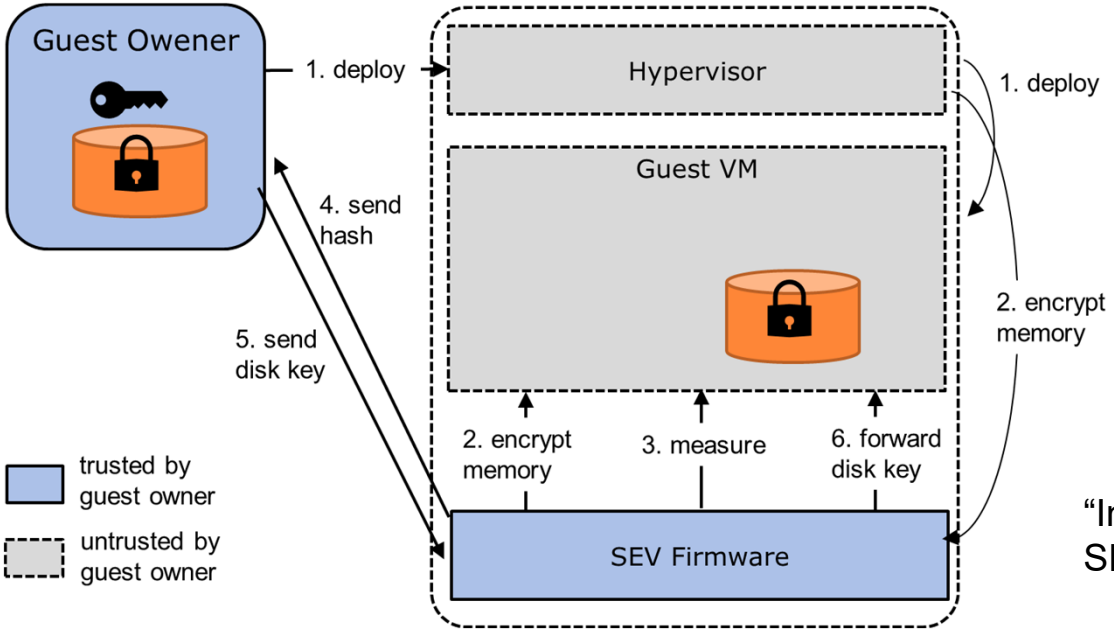
    実際に使われているのはAndroidのKeyMaster程度？

新しい方向：TEEの仮想化対応

- 仮想マシンのTEE化
  - 仮想マシン自体をEnclave/TEEとするもの
- TEE内の仮想化
  - TEE内を仮想化して複数のTrusted OSを実行するもの
- 課題

仮想マシンのTEE化 1/3

- AMD Secure Encrypted Virtualization(SEV)
  - 暗号化されたDisk ImageがSEV Firmwareで検証される。
  - VM用メモリも暗号化されHypervisorはVMの内容に関知できない。



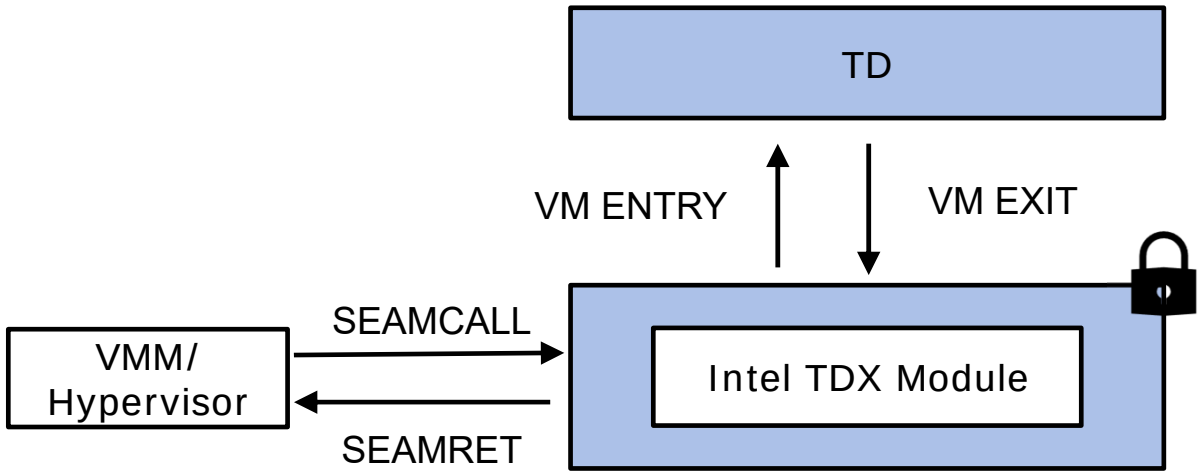
“Insecure Until Proven Updated: Analyzing AMD SEV‘s Remote Attestation”CCS19より

仮想マシンのTEE化 2/3

- Intel Trusted Domain Extensions (TDX)
  - Intel VT, TXT, SGXを組み合わせてTD (Trusted Domain)の保護
  - Secure Arbitration Mode (SEAM)を通して通信

Trusted by TD
- Intel TDC module
- Intel authenticated code (ACM)
- TD Quoting Enclave
- CPU hardware

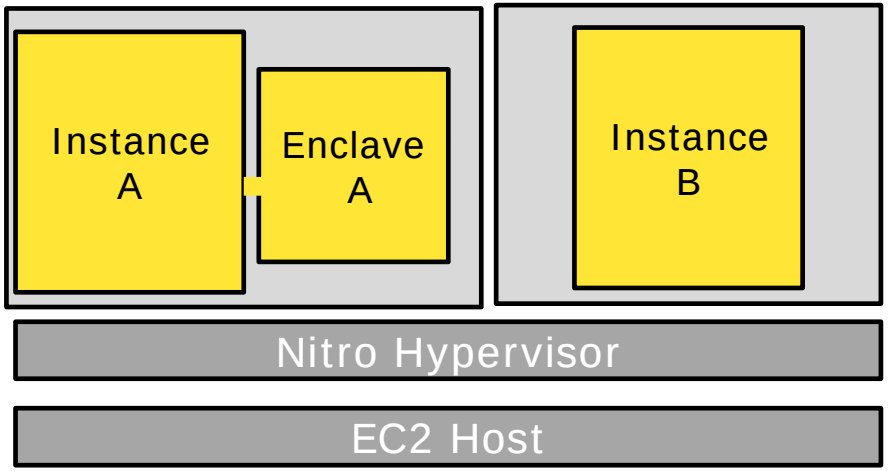
Untrusted by TD
- Platform Admin
- Devices
- All other software
- Platform Firmware
- Host-OS/VMM
- BIOS/SMM



Intel White Paper “Intel Trusted Domain Extensions”より

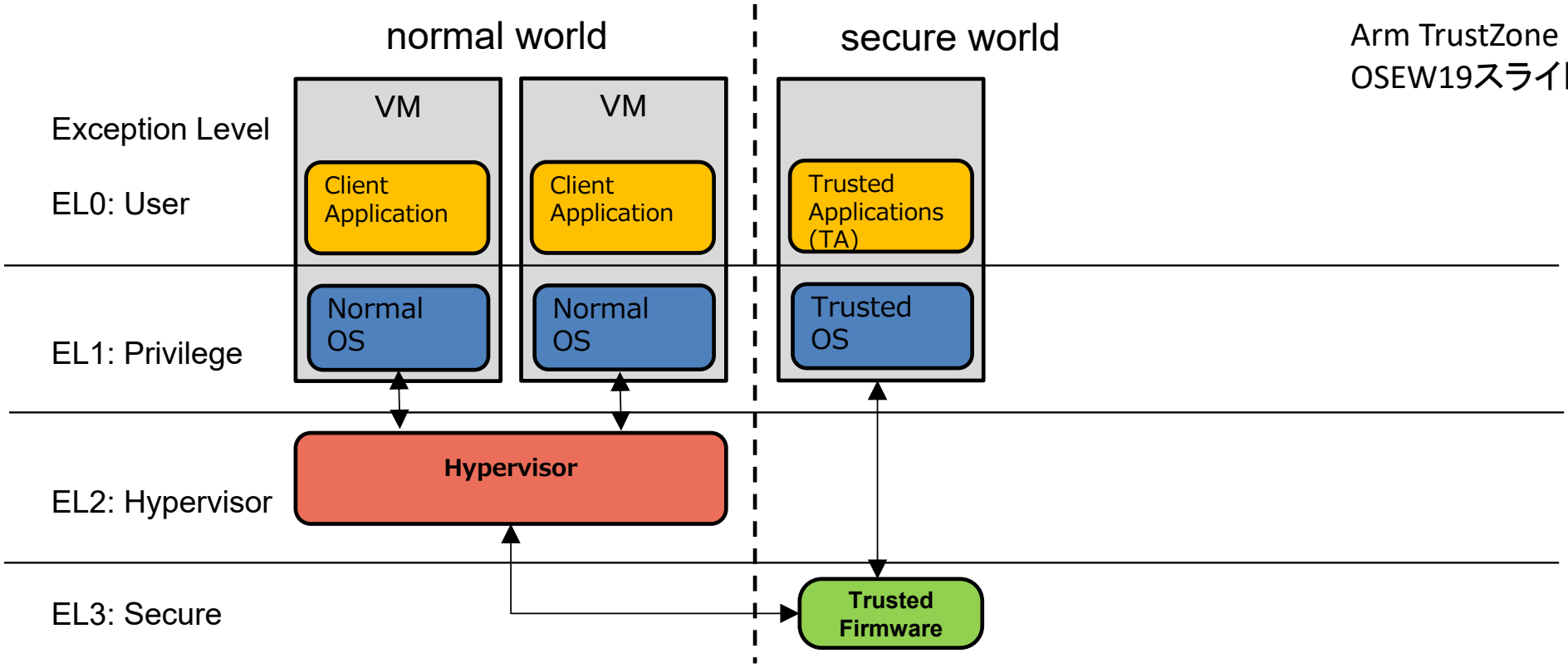
仮想マシンのTEE化 3/3

- Amazon EC2 Nitro
- EC2インスタンスのみに繋がる隔離実行(Enclave)
  - 外部ネットワーク接続も永続的なストレージもない。ローカル仮想ソケット (vsock) のみで通信。
  - Nitro Hypervisor はEnclave を作成する際に構成証明ドキュメント(適切はブート処理の測定値)を作成および署名。これにより改竄検知。
    - https://aws.amazon.com/jp/ec2/nitro/nitro-enclaves/
    - https://aws.amazon.com/jp/blogs/news/aws-nitro-enclaves-isolated-ec2-environments-to-process-confidential-data/



TEE内の仮想化: Arm TurstZoneの進化 (1/3)

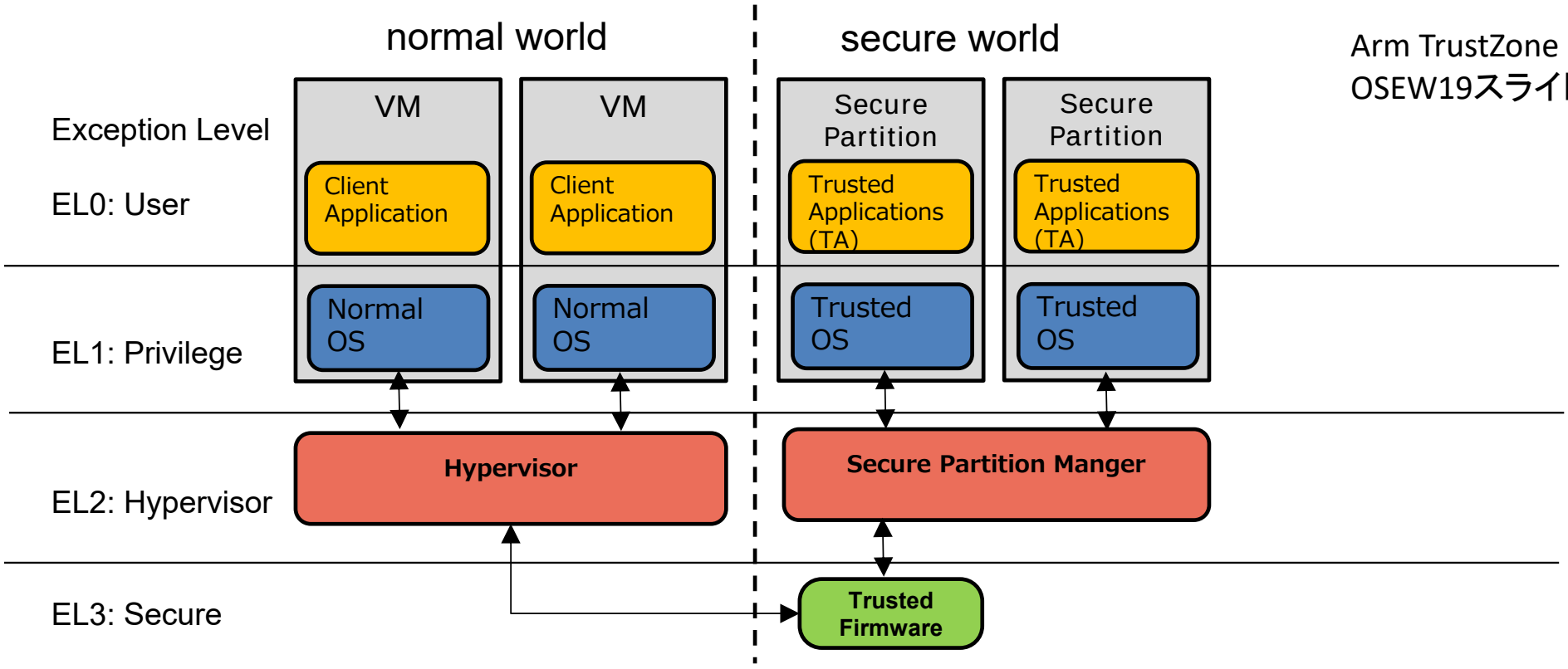
- 現状のArm Cortex A



Arm TrustZone Evolution,
OSEW19スライド資料より

TEE内の仮想化: Arm TurstZoneの進化 (2/3)

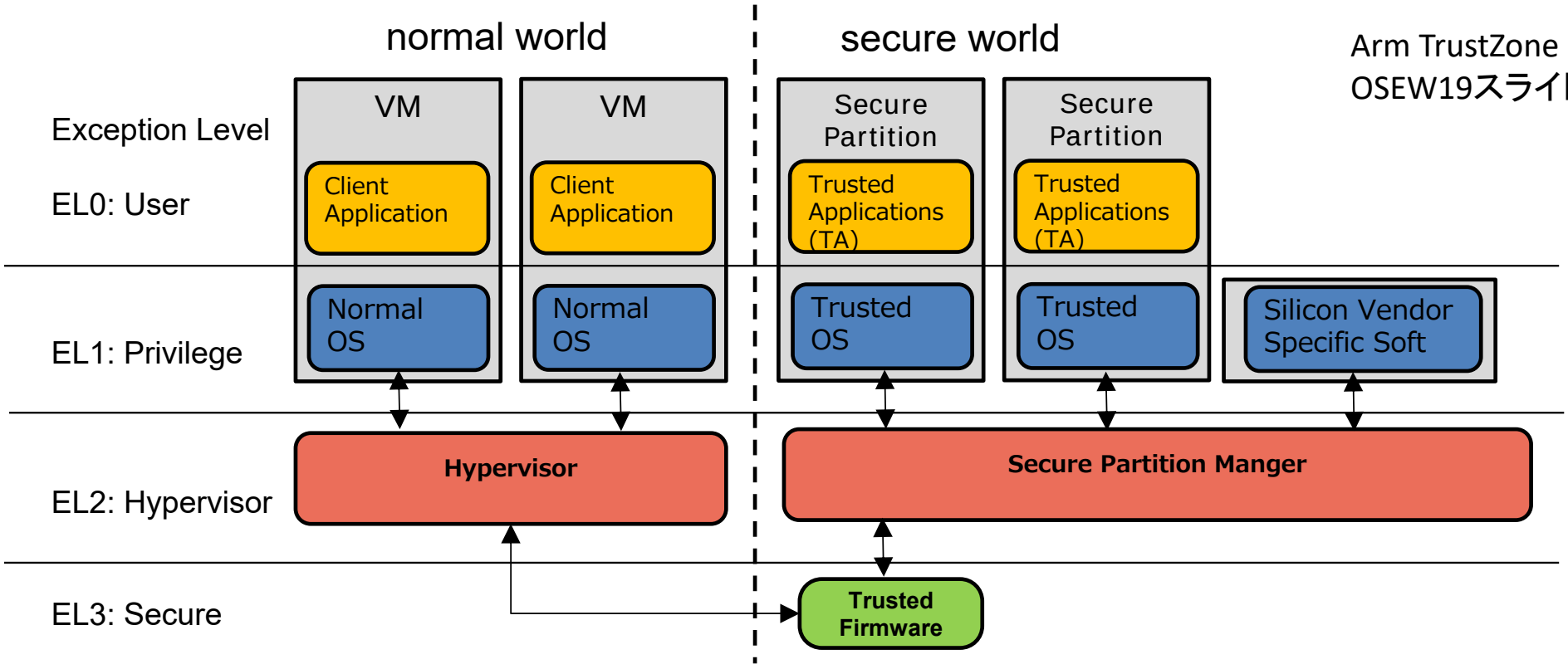
- Arm Cortex v8.4A: Secure Partition Mangerが入り、複数のTrusted OSをサポートできる。



Arm TrustZone Evolution,
OSEW19スライド資料より

TEE内の仮想化: Arm TurstZoneの進化 (3/3)

- Post v8.4A: Silicon Vendor Specific Soft がsecure worldで実行



Arm TrustZone Evolution,
OSEW19スライド資料より

TEEの仮想化のチャンスと課題

- TEEの仮想化はハードが出た段階で実装研究のチャンスがある。
- 但し、先にTEEのシステムソフト実装の多様性を見せたようにインターフェースの混乱が予想される。
- 実際の活用のためには統一規格が必要。
  - ここが規格争いの主戦場？（後述）

その他の話題

- TEEへのアンチテーゼ
- 関連組織・規格

TEEアンチテーゼ

- 隔離実行はインターフェースを増やし、そのための検査不足から脆弱性を起こす。
- むしろ、すべてを**一元的に管理し、同一レベルの形式検証を適用できるがよい**のではないか。
  - Theseus [OSDI'20]
    - Single Address Space (SAS) is good to remove "state spill".
    - The compiler of "Rust" can check the whole memory space.

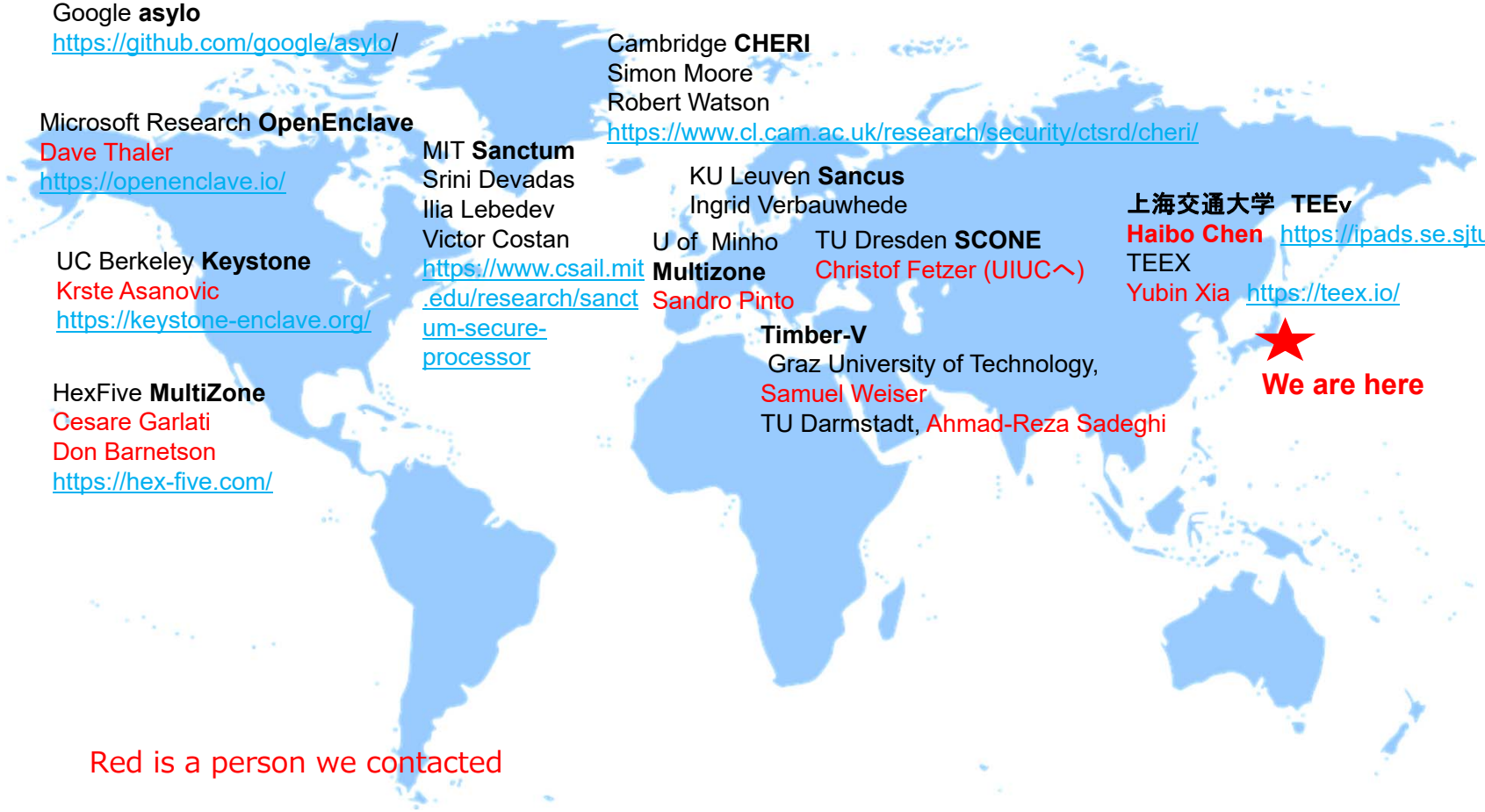
- TEEとREEの通信についてはIntel SGX/RISC-V KeystoneにEnclave Definition Language (EDL)があり、ポインタやバッファのチェックを行っている。

TEE関連組織・規格

- GlobalPlatform
  - TEE関係のAPI規格。スマートフォンで採用が多い。
  - SESIP: Security Evaluation Standard for IoT Platforms
- CCC: Confidential Computing Consortium
  - Linux Foundationプロジェクト
- TCG: Trusted Computing Group
  - TPMの仕様を作成している組織。
- Arm PSA(Platform Security Architecture) Certificate
- IETF Protocol
  - TEEP: Trusted Execution Environment Provisioning
  - RATS: Remote Attestation Procedures
  - SUIT: Software Updates for Internet of Things

- 規格争い
- 主導権争い

私から見たTEE Research Map



今後の方向性予想

- FPGAでCPU実装が簡単になっており、TEE/仮想化の追加機能ハードウェアを含むOS研究が多くなることを予想。
  - FPGAと低レイヤソフトウェア開発の知識の両方が必要
- 形式検証、安全な言語の活用
  - TEE自体の作成、アプリの作成、TEEインターフェースも厳密にする
- 実用面ではTrusted Appが作りやすい共通インターフェース
  - 規格争い

まとめ

- TEE技術の概観
- TEEの次の方向性：仮想化技術との融合、ハードウェアとの一体開発。

産総研で大学院生のRA(Research Assistant,給与あり)を募集しています。

この成果の一部は国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務(JPNP16007)の結果得られたものです。
TEE関連技術の議論を行っているセキュアオープンアーキテクチャ・エッジ基盤技術研究組合(TRASIO)の研究者に感謝します。