

Trusted RV: セキュアコプロセッサを有する64bit
RISC-V TEEとその上のソフトウェア
Trusted RV: 64bit RISC-V TEE with Secure
Coprocessor and software on them

須崎有康^{1,2}

Kuniyasu Suzuki^{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

1) Technology Research Association of Secure IoT Edge application based on RISC-V Open architecture (TRASIO)

2) 産業技術総合研究所

2) National Institute of Advanced Industrial Science and Technology (AIST)

Self Introduction (Kuniyasu Suzuki, 須崎有康)

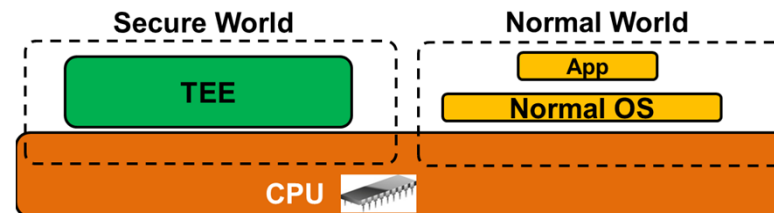
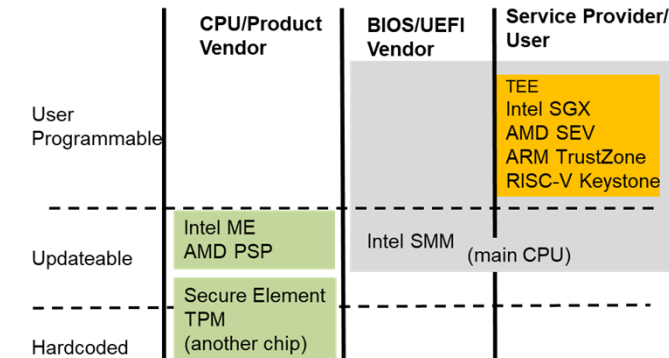
- 2009年よりTrusted ComputingやTEE関連の研究を行う
 - TCG(Trusted Computing Group) Invited Expert from 2019
- TRASIO受託NEDOプロジェクト「セキュアオープンアーキテクチャ基盤技術とそのA I エッジ
応用研究開発 FY2018-2020」でRISC-VベースのTEEの研究
 - 本講演はこちらの成果を中心に話します
- **Reference 参考資料**
 - **Trusted Execution Environmentによるシステムの堅牢化, 情報処理20/06**
 - <https://ci.nii.ac.jp/naid/40022255769>
 - **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 (無償公開)**
 - https://www.jstage.jst.go.jp/article/essfr/14/2/14_107/article/-char/ja/

Contents

- What is TEE?
- 4 Issues of TEE (Trusted Execution Environment) on RISC-V
 1. Root of Trust
 2. Programming Environment
 3. TA(Trusted Application) Management
 4. Remote Attestation
- 4 Security Technologies offered by TRASIO
 - Hardware
 1. Trusted-RV Platform (64-bit RISC-V + 32-bit RISC-V Secure CoProcessor)
 - Software
 2. TEE Programming Environment (GlobalPlatform TEE Internal API)
 3. TA Management Framework: TEEP(Trusted Execution Environment Provisioning)
 4. Remote Attestation
- Future works and Conclusions

What is TEE (Trusted Execution Environment)?

- TEE is one of CPU's execution environments isolated from OS
 - **Caution: TEE is not only-one isolated execution environment**
 - SMM, Intel ME, and TPM are also isolated from OS.
 - TEE is different from others because **it is programmable and opened for normal user.**
- TEE separates the execution environment into 2 worlds
 - Normal World (i.e., REE: Rich Execution Environment) for normal OS and apps
 - Secure World (i.e., TEE: Trusted Execution Environment) for critical apps.



- Available Hardware: ARM TrustZone, Intel SGX, AMD SEV, and RISC-V

RISC-V TEE

- RISC-V TEE implementations

- Academia {
 - Sanctum [MIT, USENIX Sec'16]
 - TIMBER-V [Graz University of Technology, NDSS'19]
 - MI6 [MIT, MICRO'19]
 - Keystone [UC Berkeley, EuroSys'20]
 - HECTOR-V [Graz University of Technology, arXiv'21]
 - CURE [Technische Universität Darmstadt, USENIX Sec'21]
- Industry {
 - MultiZone [HexFive]



**Keystone is an active open-source project.
This talk is based on Keystone.**

Problem of TEE (Root of Trust) 1/4

- **TEE** is just an isolated execution environment and cannot be a Root of Trust.
 - Root of Trust keeps keys and certificates and must be Secure CoProcessor.
 - **Remote Attestation** must be based on Root of Trust.
- Example of Root of Trust
 - Intel SGX has Intel ME(Management Engine). Intel Quark x86-based (32bit)
 - AMD SEV has PSP(Platform Security Processor). Arm Cortex-A5 (32bit)
 - Arm TrustZone needs an extra IP
 - CryptCell(Discretix -> Arm) ● CryptoManager (Rambus) ● Secure Element
 - Apple M2
 - RISC-V needs an extra IP
 - Rambus RISC-V CryptoManager ● Silex Insight Secure Root of Trust
 - OpenTitan (Open Source)

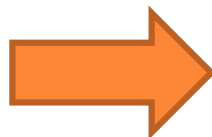


Most of them, the detail is not open.

We cannot verify them and need to trust them.

Problem of TEE (Programming) 2/4

- Each TEE has each SDK for programming
 - Intel SGX
 - Intel SGX SDK
 - Open Enclave (Microsoft)
 - Asylo (Google)
 - AMD SEV
 - Asylo (Google)
 - Enarx (Redhat)
 - Arm Cortex-A TrustZone
 - Open Enclave (Microsoft)
 - GlobalPlatform (GP) TEE Internal API
 - RISC-V Keystone
 - Keystone SKD



**No compatibility and
No portability
for different CPU architecture.**

Problem of TEE (TA Management) 3/4

- Is a TA **installed, updated, and deleted safely**?
 - A TA is developed by a third party (e.g., video supplier, bank), but the supplier and client want to confirm the safety each other.
 - From the view of platform (TEE Edge device)
 - Is the TA trustable? Is the download server trustable?
 - From the view of TA
 - Is the platform genuine (no tempered)?



Management of TA must be safe.

In addition, the management must follow each CPU security procedure.

Problem of TEE (Remote Attestation) 4/4

- Does a genuine TA run on a genuine platform (no tempered)?
 - Remote Attestation is a mechanism to certificate platform and TA.
 - Basement of install/update/delete (problem 3)



Device keys and certificates must be managed by *Root of Trust*.

Security Technologies offered by TRASIO

- Hardware

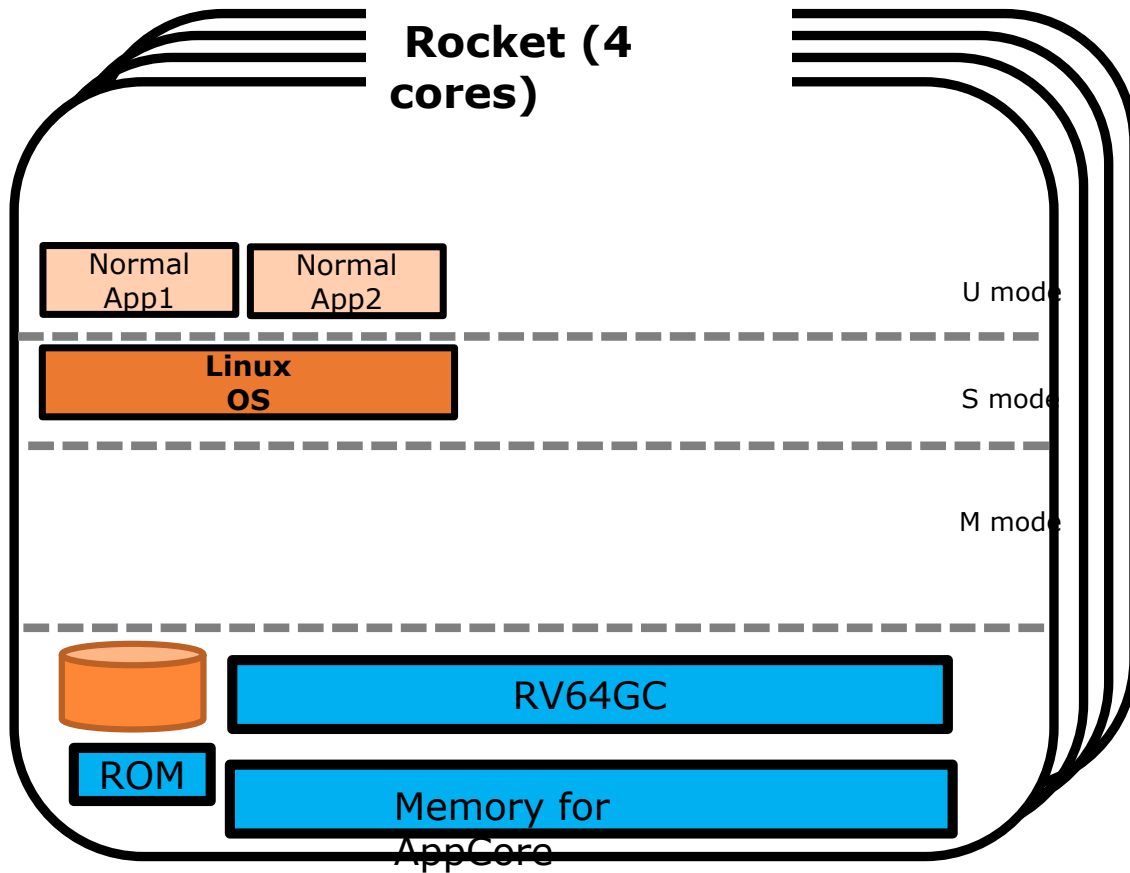
1. Trusted-RV Platform (64-bit RISC-V + 32-bit RISC-V Secure CoProcessor)

- Software

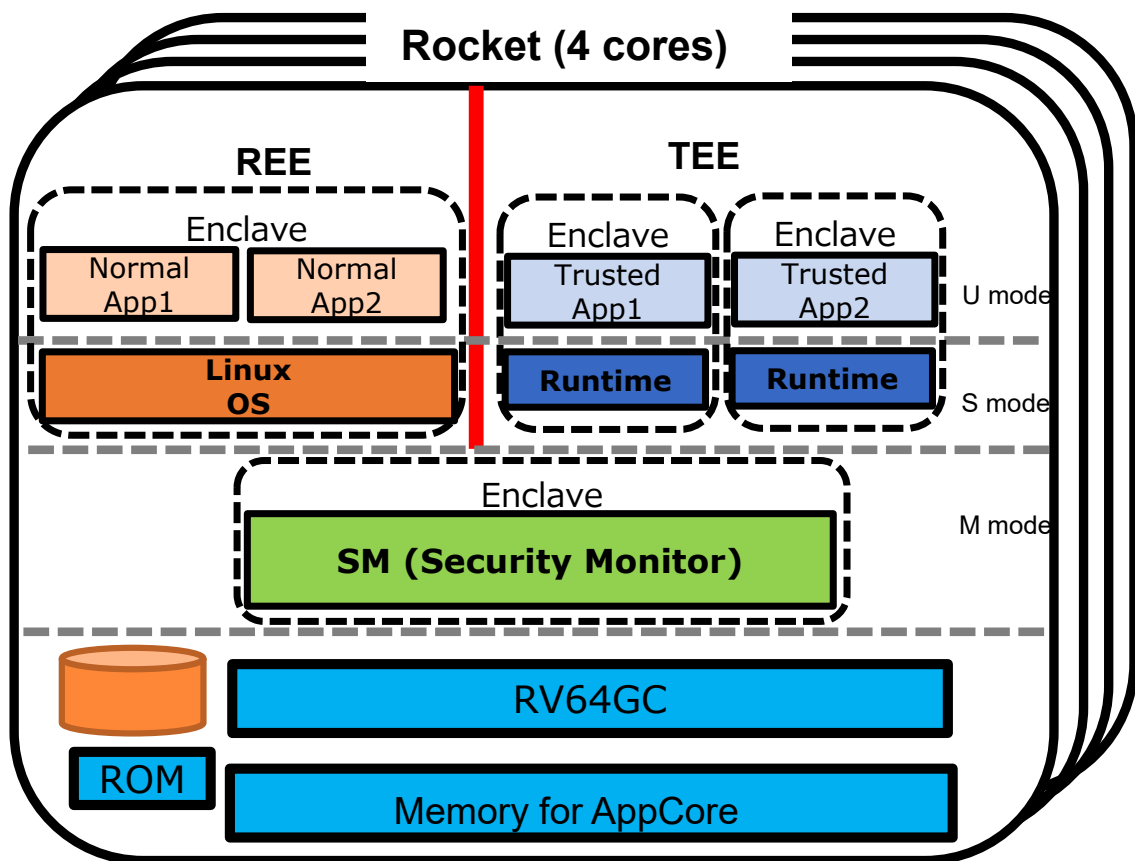
2. TEE's Programming Environment: GlobalPlatform TEE Internal API
3. TA Management Framework: TEEP(Trusted Execution Environment Provisioning)
4. Remote Attestation

Normal RISC-V

- Rocket cores are assumed.

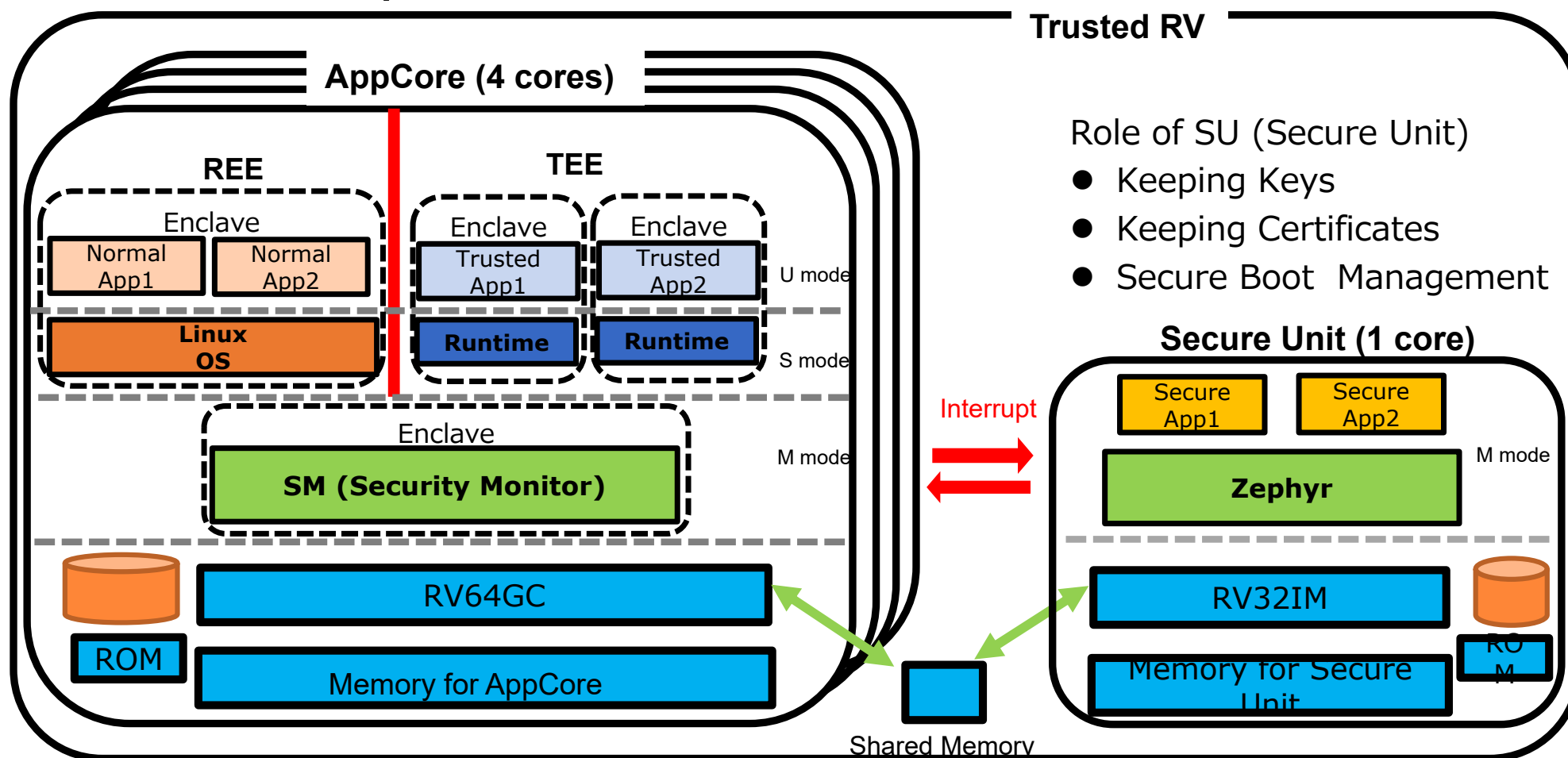


Keystone enabled RISC-V



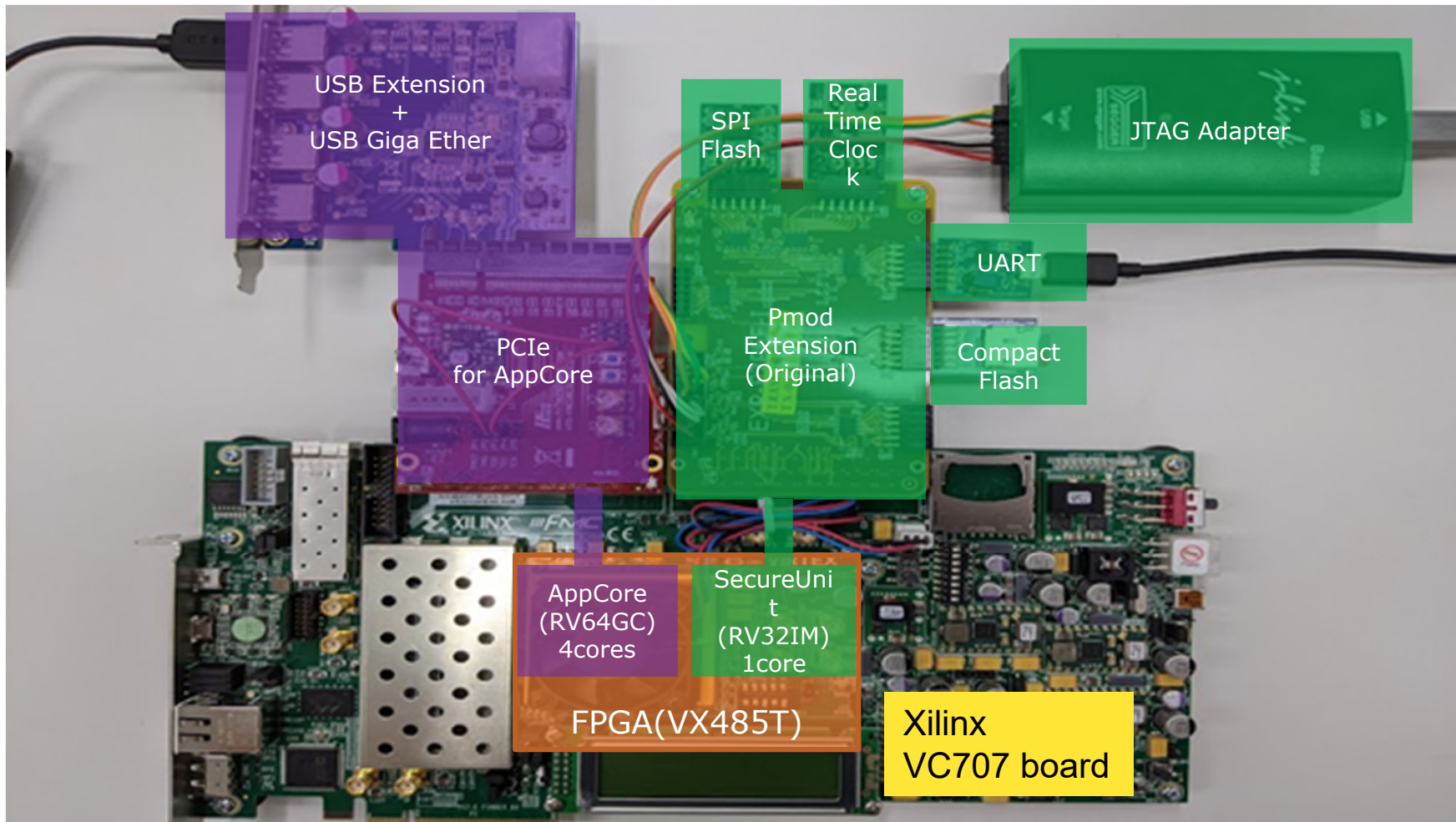
- PMP (Physical Memory Protection) isolates memory. Dotted line indicates an Enclave.
 - One Secure Monitor in M mode
 - One OS (Linux) in REE
 - Two Enclaves in TEE

Keystone with Secure CoProcessor



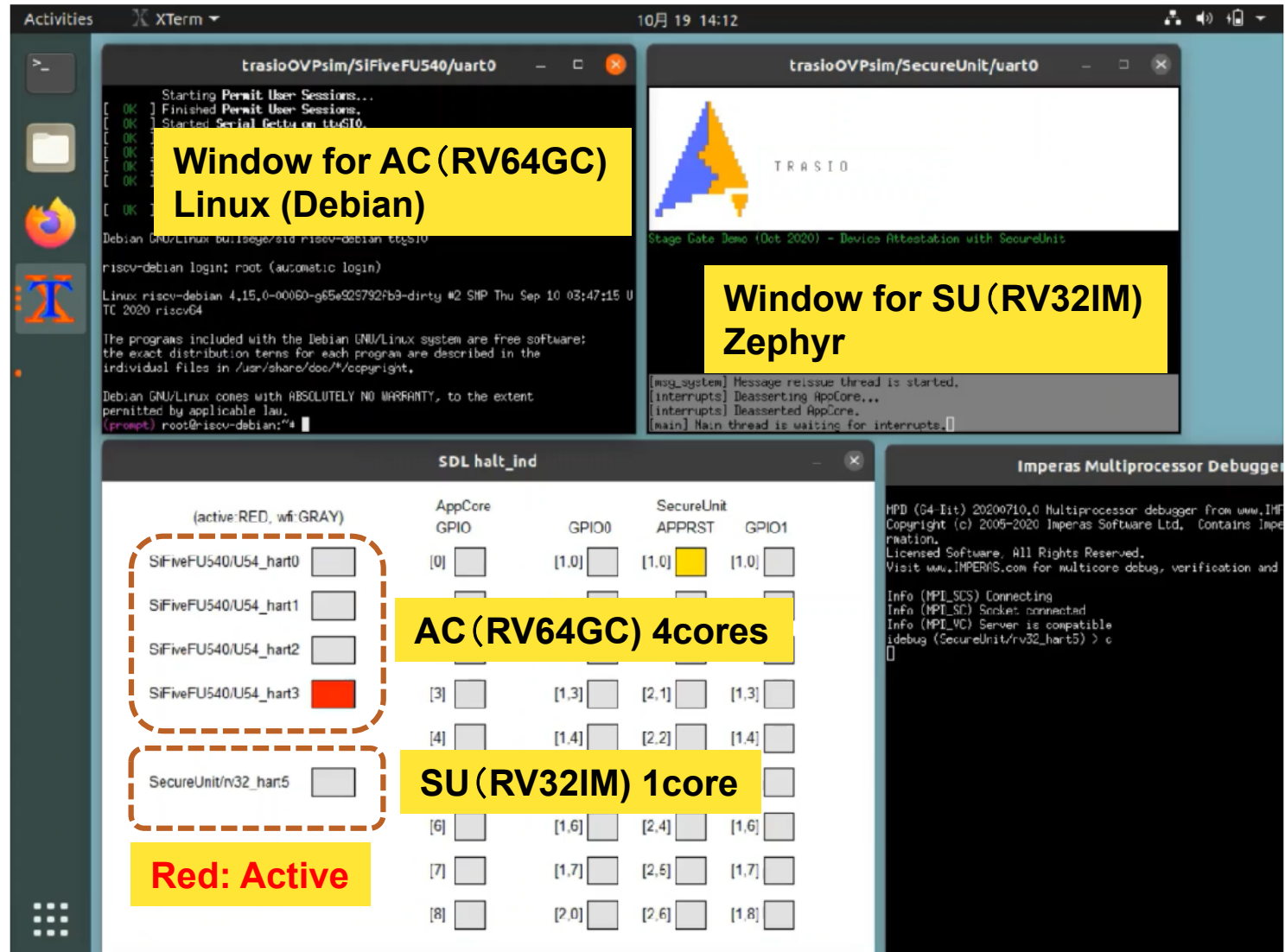
- Role of SU (Secure Unit)
- Keeping Keys
 - Keeping Certificates
 - Secure Boot Management

FPGA Implementation



Simulator

- Based on Imperas RISC-V simulator
- Used for system software development



The screenshot shows the simulator interface with several windows:

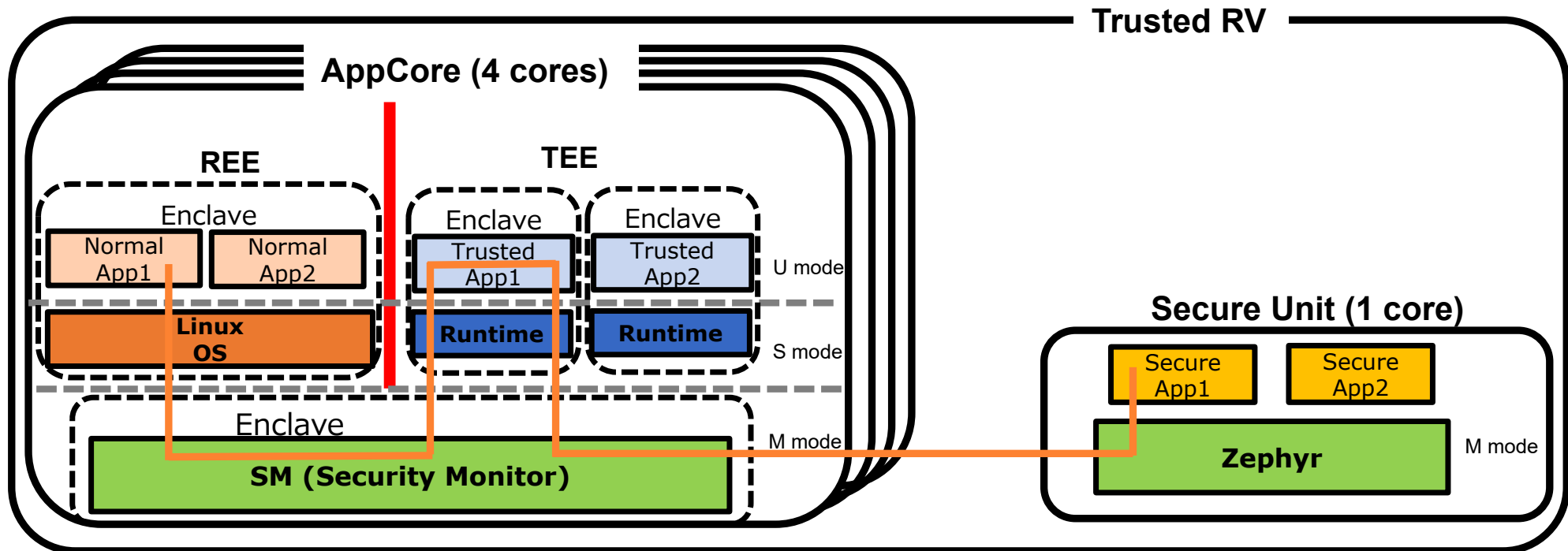
- Window for AC (RV64GC) Linux (Debian):** A terminal window showing the boot process of a Debian Linux system on a RISC-V architecture. It includes messages like "Starting Permit User Sessions...", "Finished Permit User Sessions.", and "Started Serial Getty on ttyS10". The prompt is "root@riscv-debian:~#".
- Window for SU (RV32IM) Zephyr:** A terminal window showing the boot process of a Zephyr RTOS on a RISC-V architecture. It includes messages like "[msg_system] Message reissue thread is started.", "[interrupts] Deasserting AppCore...", and "[main] Main thread is waiting for interrupts.".
- SDL halt_ind:** A window displaying a table of core status. The table has columns for AppCore, GPIO, GPIO0, SecureUnit, APPRST, and GPIO1. The first four rows (cores 0-3) are grouped as "AC (RV64GC) 4cores". The fifth row (core 5) is grouped as "SU (RV32IM) 1core". A red box highlights the "SIFiveFU540/U54_hart3" core, and a yellow box below it says "Red: Active".
- Imperas Multiprocessor Debugger:** A window showing the debugger's status and logs, including "Info (MPI_SCS) Connecting" and "Info (MPI_VC) Server is compatible".

Core	AppCore	GPIO	GPIO0	SecureUnit	APPRST	GPIO1
0	[0]	[0]	[1,0]	[1,0]	[1,0]	[1,0]
1	[1]	[1]	[1,1]	[1,1]	[1,1]	[1,1]
2	[2]	[2]	[1,2]	[1,2]	[1,2]	[1,2]
3	[3]	[3]	[1,3]	[2,1]	[1,3]	[1,3]
4	[4]	[4]	[1,4]	[2,2]	[1,4]	[1,4]
5	[5]	[5]	[1,5]	[2,3]	[1,5]	[1,5]
6	[6]	[6]	[1,6]	[2,4]	[1,6]	[1,6]
7	[7]	[7]	[1,7]	[2,5]	[1,7]	[1,7]
8	[8]	[8]	[2,0]	[2,6]	[1,8]	[1,8]

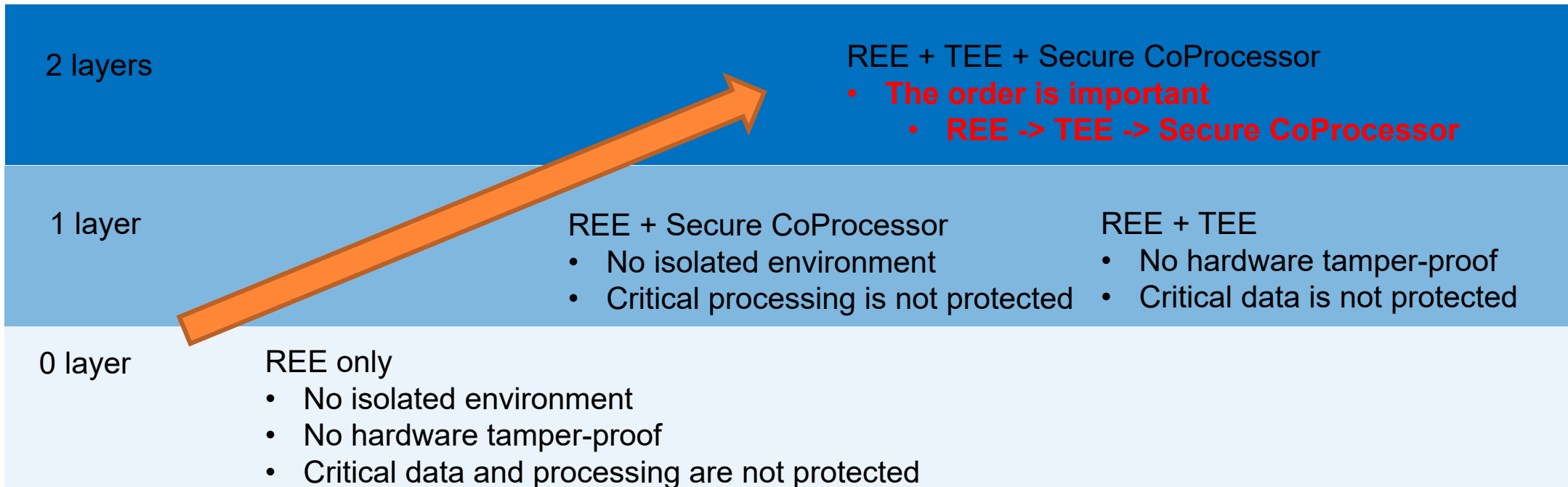
Software structure for Secure CoProcessor

- The communication is limited

- (Normal App) -> (Secure Monitor) -> (Trusted App) -> (Secure Monitor) -> (Zephyr) -> (Secure App)
- No direct access from Normal App / Linux is not allowed.
- **Trusted App and Secure App must not leak critical information.**
 - design depends on use case because TEE is powerful 64RV and SU is low power RV32.



Layer of TEE and Secure CoProcessor



Comparison of RISC-V Secure CoProcessor *this table is not complete.

	Google OpenTitan	Rambus RISC-V CryptoManager (RT-6*0, RT-7*0)	Silex Insight eSecure (BA470)	Trusted RV Secure Unit
Core	Ibex (RV32IMC/EMC) M/U Mode	Custom (RV32IMC) M/S/U Mode	Andes N22 (RV32IMAC/EMAC) M or M/U mode	Custom (RV32IM) M mode
OS	Tock OS	Zephyr	---	Zephyr
Comm to Main	SPI	GPIO/SPI	---	GPIO Shared Memory
Accelerator	AES,SHA	AES, SHA	AES, SHA	Not yet ***
Peripherals	Timer, RNG	Timer, RNG	RNG	Timer, RNG, Flash
Anti-tampering	Yes?	Yes	Yes	Not yet
Target	Key Management, Secure Boot, OTA	Key Management, Secure Boot, OTA, User App	Key Management, Secure Boot, OTA	Key Management, Secure Boot, OTA
Misc.	QEMU support	FIPS 140-2 Level 2	FIPS 140 2 level 3 PUF for Unique Key	Design with TEE (Different part from OpenTitan)

*** We have developed the accelerator for SHA-3 and Ed25519 for quick boot.

“Quick Boot of Trusted Execution Environment With Hardware Accelerator”, IEEE Access 2020 <https://ieeexplore.ieee.org/document/9064723>

Security Technologies offered by TRASIO

- Hardware

1. Trusted-RV Platform (64-bit RISC-V + 32-bit RISC-V Secure CoProcessor)

- Software

2. TEE's Programing Environment: GlobalPlatform TEE Internal API
3. TA Management Framework: TEEP(Trusted Execution Environment Provisioning)
4. Remote Attestation

TEE's Programming: GlobalPlatform TEE Internal API

- GP TEE Internal API does not depend on CPU architecture and is used by many Smartphones
 - kinibi (Trustonic)
 - Kinibi runs on 1.7 billion devices [USENIX Sec20,PARTEMU]
 - QSEE (Qualcomm)
 - 60% Android phones uses SQEE as of 2019 [USENIX Sec20,PARTEMU]
 - OP-TEE(Linaro)
 - Open-source implementation.
- We have developed some applications with GP API on OP-TEE and want to port them to Intel SGX and RISC-V Keystone.

What we did for Keystone and SGX

- We designed the GP internal API library to be portable.
 - We utilize SDK to implement a library which offers new abstraction.
 - The library is ported to Intel SGX as well as RISC-V Keystone.



● **Implementation Challenge**

- Some APIs depend on hardware.
 - We separate APIs into hardware dependent / independent.
- Integrate GP TEE Internal APIs to Keystone SDK
 - Keystone SDK includes EDL (Enclave Definition Language) named "keedger".
 - EDL creates the code for communication (request from TEE to REE) to check the pointer and boundary.



The specification of GP TEE internal API



Functions by Category

Asymmetric	TEE_FreeOperation, 184	Other Arithmetic	TEE_ResetPropertyEnumerator, 80
TEE_AsymmetricDecrypt, 211	TEE_GetOperationInfo, 185	TEE_BigIntComputeExtendedGcd, 278	TEE_StartPropertyEnumerator, 78
TEE_AsymmetricEncrypt, 211	TEE_GetOperationInfoMultiple, 186	TEE_BigIntIsProbablePrime, 277	Random Data Generation
TEE_AsymmetricSignDigest, 213	TEE_IsAlgorithmSupported, 194	TEE_BigIntRelativePrime, 275	TEE_GenerateRandom, 222
TEE_AsymmetricVerifyDigest, 218	TEE_ResetOperation, 188		TEE_CipherUpdate, 199
Authenticated Encryption	TEE_SetOperationKey, 189	Panic Function	Symmetric Cipher
TEE_AEDeCryptFinal, 210	TEE_SetOperationKey2, 191	TEE_Panic, 95	TEE_CipherDoFinal, 200
TEE_AEEncryptFinal, 209		Persistent Object	TEE_CipherInit, 197
TEE_AEInit, 205	Initialization	TEE_CloseAndDeletePersistentObject (deprecated), 334	TEE_CipherUpdate, 199
TEE_AEUpdate, 208	TEE_BigIntInit, 246	TEE_CloseAndDeletePersistentObject1, 160	TA Interface
TEE_AEUpdateAAD, 207	TEE_BigIntInitFMM, 249	TEE_CreatePersistentObject, 155	TA_CloseSessionEntryPoint, 62
	TEE_BigIntInitFMMContext, 247, 335	TEE_OpenPersistentObject, 153	TA_CreateEntryPoint, 58
Basic Arithmetic	Internal Client API	TEE_RenamePersistentObject, 181	TA_DestroyEntryPoint, 59
TEE_BigIntAdd, 261	TEE_CloseTASession, 97	Persistent Object Enumeration	TA_InvokeCommandEntryPoint, 63
TEE_BigIntDiv, 266	TEE_InvokeTACommand, 99	TEE_AllocatePersistentObjectEnumerator, 162	TA_OpenSessionEntryPoint, 60
TEE_BigIntMul, 264	TEE_OpenTASession, 98	TEE_FreePersistentObjectEnumerator, 162	Time
TEE_BigIntNeg, 263	Key Derivation	TEE_GetNextPersistentObject, 168	TEE_GetREETime, 239
TEE_BigIntSquare, 265	TEE_DeriveKey, 219	TEE_ResetPersistentObjectEnumerator, 164	TEE_GetSystemTime, 234
TEE_BigIntSub, 262	Logical Operation	TEE_StartPersistentObjectEnumerator, 165	TEE_GetTAPersistentTime, 238
Cancellation	TEE_BigIntCmp, 254	Property Access	TEE_Wait, 236
TEE_GetCancellationFlag, 103	TEE_BigIntCmpS32, 254	TEE_AllocatePropertyEnumerator, 77	Transient Object
TEE_MaskCancellation, 105	TEE_BigIntGetBit, 257	TEE_FreePropertyEnumerator, 78	TEE_AllocateTransientObject, 134
TEE_UnmaskCancellation, 105	TEE_BigIntGetBitCount, 257	TEE_GetNextProperty, 82	TEE_CopyObjectAttributes (deprecated), 333
Converter	TEE_BigIntShiftRight, 256	TEE_GetPropertyAsBinaryBlock, 74	TEE_CopyObjectAttributes1, 147
TEE_BigIntConvertFromOctetString, 250	MAC	TEE_FreeTransientObject, 138	TEE_FreeTransientObject, 138
TEE_BigIntConvertFromS32, 252	TEE_MACCompareFinal, 204	TEE_GetPropertyAsBool, 71	TEE_GenerateKey, 149
TEE_BigIntConvertToOctetString, 251	TEE_MACComputeFinal, 203	TEE_GetPropertyAsIdentity, 76	TEE_InitRefAttribute, 145
TEE_BigIntConvertToS32, 253	TEE_MACInit, 201	TEE_GetPropertyAsString, 70	TEE_InitValueAttribute, 145
Data Stream Access	TEE_MACUpdate, 202	TEE_GetPropertyAsU32, 72	TEE_PopulateTransientObject, 140
TEE_ReadObjectData, 168	Memory Allocation and Size of Objects	TEE_GetPropertyAsU64, 73	TEE_ResetTransientObject, 138
TEE_SeekObjectData, 173	TEE_BigIntFMMContextSizeInU32, 244	TEE_GetPropertyAsUUID, 75	
TEE_TruncateObjectData, 172	TEE_BigIntFMMSizeInU32, 245	TEE_GetPropertyName, 81	
TEE_WriteObjectData, 170	TEE_BigIntSizeInU32 (macro), 243		
Deprecated	Memory Management		
TEE_CloseAndDeletePersistentObject, 334	TEE_CheckMemoryAccessRights, 107		
TEE_CopyObjectAttributes, 333	TEE_Free, 116		
TEE_GetObjectInfo, 330	TEE_GetInstanceData, 111		
TEE_RestrictObjectUsage, 332	TEE_Malloc, 112		
Fast Modular Multiplication	TEE_MemCompare, 118		
TEE_BigIntComputeFMM, 280	TEE_MemFill, 119		
TEE_BigIntConvertFromFMM, 279	TEE_MemMove, 117		
TEE_BigIntConvertToFMM, 278	TEE_Realloc, 114		
Generic Object	TEE_SetInstanceData, 110		
TEE_CloseObject, 133	Message Digest		
TEE_GetObjectBufferAttribute, 130	TEE_DigestDoFinal, 198		
TEE_GetObjectInfo (deprecated), 330	TEE_DigestUpdate, 195		
TEE_GetObjectInfo1, 127	Modular Arithmetic		
TEE_GetObjectValueAttribute, 132	TEE_BigIntAddMod, 269		
TEE_RestrictObjectUsage (deprecated), 332	TEE_BigIntInvMod, 273		
TEE_RestrictObjectUsage1, 129	TEE_BigIntMod, 268		
Generic Operation	TEE_BigIntMulMod, 271		
TEE_AllocateOperation, 180	TEE_BigIntSquareMod, 272		
TEE_CopyOperation, 193	TEE_BigIntSubMod, 270		

GlobalPlatform Technology
 TEE Internal Core API Specification
 Version 1.1.2.50 (Target v1.2)

Public Review
 June 2018
 Document Reference: GPD_SPE_010

Copyright © 2011-2018 GlobalPlatform, Inc. All Rights Reserved.
 Recipients of this document are invited to submit, with their comments, notification of any relevant patents or other intellectual property rights (collectively, "IPR") of which they may be aware which might be necessarily infringed by the implementation of the specification or other work product set forth in this document, and to provide supporting documentation. The technology provided or described herein is subject to updates, revisions, and extensions by GlobalPlatform. This documentation is currently in draft form and is being reviewed and enhanced by the Committees and Working Groups of GlobalPlatform. Use of this information is governed by the GlobalPlatform license agreement and any use inconsistent with that agreement is strictly prohibited.

Copyright © 2011-2018 GlobalPlatform, Inc. All Rights Reserved.
 The technology provided or described herein is subject to updates, revisions, and extensions by GlobalPlatform. Use of this information is governed by the GlobalPlatform license agreement and any use inconsistent with that agreement is strictly prohibited.

Copyright © 2011-2018 GlobalPlatform, Inc. All Rights Reserved.
 The technology provided or described herein is subject to updates, revisions, and extensions by GlobalPlatform. Use of this information is governed by the GlobalPlatform license agreement and any use inconsistent with that agreement is strictly prohibited.

29 categories
 131 functions

Cipher Suite

Table 6-1: Supported Cryptographic Algorithms⁴

Algorithm Type	Supported Algorithm
Digests	MD5 SHA-1 SHA-256 SHA-224 SHA-384 SHA-512 SM3-256
Symmetric ciphers	DES Triple-DES with double-length and triple-length keys AES SM4
Message Authentication Codes (MACs)	DES-MAC AES-MAC AES-CMAC HMAC with one of the supported digests
Authenticated Encryption (AE)	AES-CCM with support for Additional Authenticated Data (AAD) AES-GCM with support for Additional Authenticated Data (AAD)
Asymmetric Encryption Schemes	RSA PKCS1-V1.5 RSA OAEP
Asymmetric Signature Schemes	DSA RSA PKCS1-V1.5 RSA PSS
Key Exchange Algorithms	Diffie-Hellman

Table 6-2: Optional Cryptographic Algorithms

Algorithm Type	Algorithm Name	When Supported
Asymmetric Signature Schemes on generic curve types	ECDSA	Any of the curves in Table 6-14 for which "generic" is Y
Key Exchange Algorithms on generic curve types	ECDH	Any of the curves in Table 6-14 for which "generic" is Y
Asymmetric Signature on Edwards Curves	ED25519	Any Edwards curve is supported
Key Exchange Algorithms on Edwards Curves	X25519	Any Edwards curve is supported
Various asymmetric Elliptic Curve-based cryptographic schemes using the SM2 curve.	SM2	SM2 is supported
Various signature and HMAC schemes based on the SM3 hash function.	SM3	SM2 is supported (SM2 support implies support for SM3. See Table 4-14).
Various symmetric encryption-based schemes based on SM4 symmetric encryption	SM4	SM2 is supported (SM2 support implies support for SM4. See Table 4-14).

Separate GP TEE Internal API

- Hardware dependent
 - Random Generator, Time, Secure Storage, Transient Object(TEE_GenerateKey)
- Hardware independent (Crypto)
 - Transient Object(exclude TEE_GenerateKey), Crypto Common, Authenticated Encryption, Symmetric/Asymmetric Cipher, Message Digest

Category	CPU (In)Dependent	Functions
Random Number	Dependent	TEE_GenerateRandom
Time	Dependent	TEE_GetREETime, TEE_GetSystemTime
Secure Storage	Dependent	TEE_CreatePersistentObject, TEE_OpenPersistentObject, TEE_ReadObjectData, TEE_WriteObjectData, TEE_CloseObject
Transient Object	Dependent Independent	TEE_GenerateKey, TEE_AllocateTransientObject, TEE_FreeTransientObject, TEE_InitRefAttribute, TEE_InitValueAttribute, TEE_SetOperationKey
Crypto Common	Independent	TEE_AllocateOperation, TEE_FreeOperation
Authenticated Encryption	Independent	TEE_AEInit, TEE_AEUpdateAAD, TEE_AEUpdate, TEE_AEEncryptFinal, TEE_AEDecryptFinal
Symmetric Cipher	Independent	TEE_CipherInit, TEE_CipherUpdate, TEE_CipherDoFinal
Asymmetric Cipher	Independent	TEE_AsymmetricSignDigest, TEE_AsymmetricVerifyDigest
Message Digest	Independent	TEE_DigestUpdate, TEE_DigestDoFinal

Reference

1. Library Implementation and Performance Analysis of GlobalPlatform TEE Internal API for Intel SGX and RISC-V Keystone[TrustCom2020] <https://conferences.computer.org/trustcompub/pdfs/TrustCom2020-4sqfK5r538MStgrShyle8b/438000b200/438000b200.pdf>
2. Portable Implementation of GlobalPlatform API for TEE[RISC-V Global Forum 2020] <https://riscvglobalforum2020.sched.com/event/dO37>

Security Technologies offered by TRASIO

- Hardware

1. Trusted-RV Platform (64-bit RISC-V + 32-bit RISC-V Secure CoProcessor)

- Software

2. TEE's Programming Environment: GlobalPlatform TEE Internal API
3. TA Management Framework: TEEP(Trusted Execution Environment Provisioning)
4. Remote Attestation

TEEP(Trusted Execution Environment Provisioning)

- TEEP is a protocol to manage TA(Trusted Application) to Install/Update/Delete.
 - Caution: Execution is out of scope because it depends CPU Architecture.

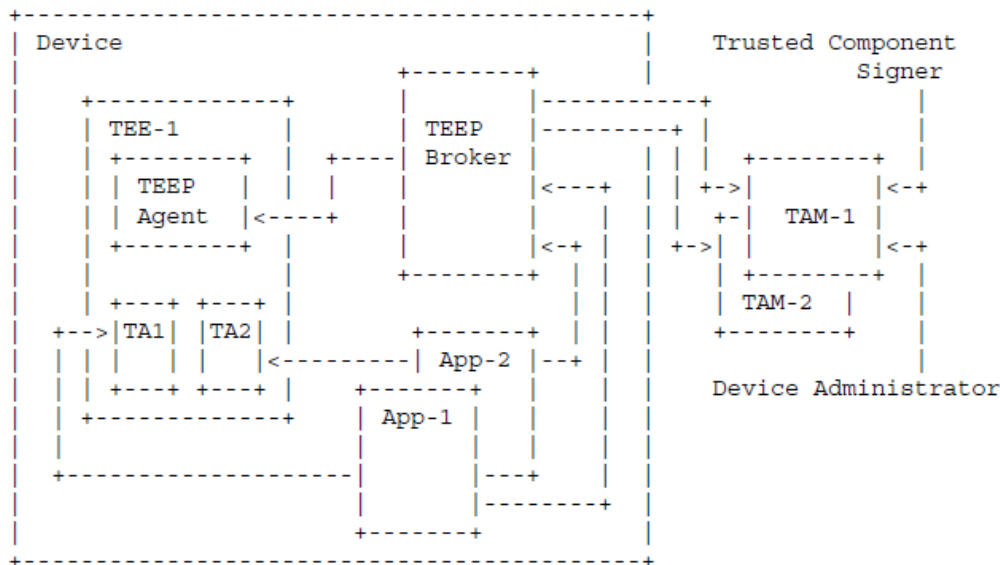


Figure 1: Notional Architecture of TEEP

Purpose	Cardinality & Location of Private Key	Private Key Signs	Location of Trust Anchor Store
Authenticating TEE	1 per TEE	TEEP responses	TAM
Authenticating TAM	1 per TAM	TEEP requests	TEEP Agent
Code Signing	1 per Trusted Component Signer	TA binary	TEE

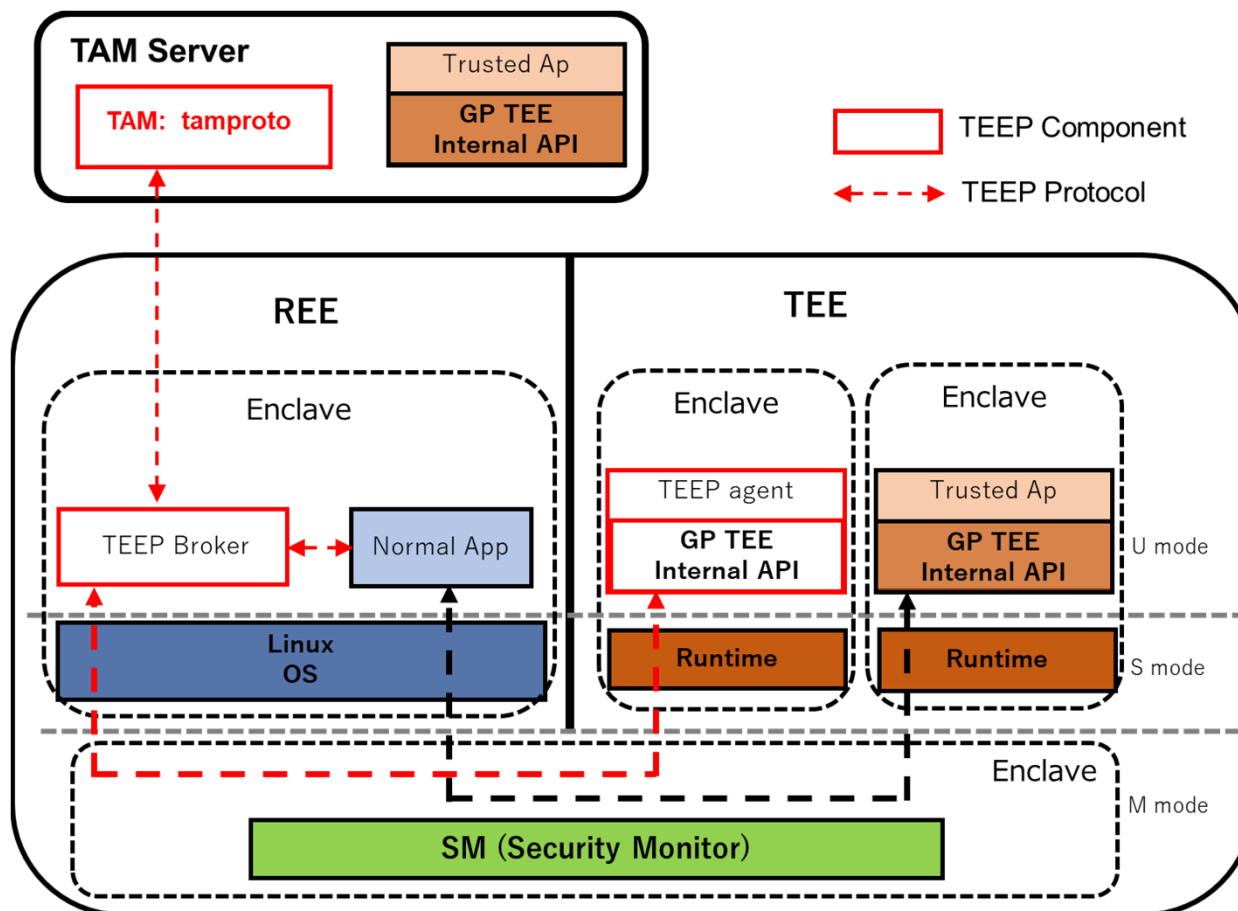
Figure 4: Signature Keys

TEE on RISC-V Keystone

- The implementation uses the GP TEE internal API

- *Reference*

- *“TEEP (Trusted Execution Environment Provisioning) Implementation on RISC-V”, FOSDEM2020*



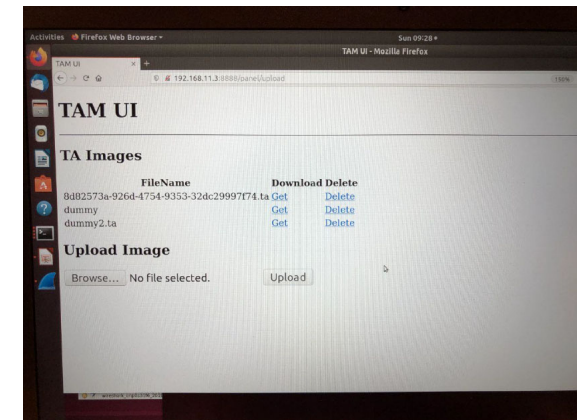
IETF TEEP Hackathon

- IETF 104 (Prague, March/2019) TEEP Hackathon
 - Design of key management
- IETF 105 (Montreal, July,2019) TEEP Hackathon
 - TEEP on Arm Hikey
- IETF 106(Singapore, Nov/2019) TEEP Hackathon
 - Connect to Prototype TAM Sever
- IETF 108(Online, Nov/2020) TEEP Hackathon
 - Adapting revised TEEP and proposing Improving TEEP spec
- IETF 109(Online, Nov/2020) TEEP Hackathon
 - Adapting revised TEEP and proposing Improving TEEP spec
- IETF 110(Online, March2020) TEEP Hackathon
 - Applying SUIT manifest



Isobe(TRASIO/Secom) TAM UI

Online hackathon
On gather.town





Security Technologies offered by TRASIO

- Hardware

1. Trusted-RV Platform (64-bit RISC-V + 32-bit RISC-V Secure CoProcessor)

- Software

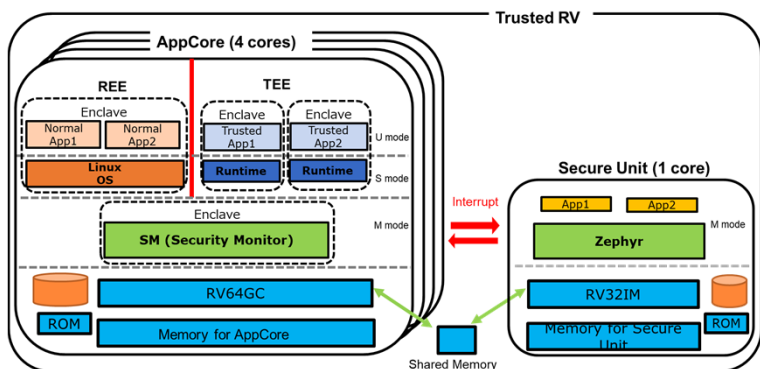
2. TEE's Programming Environment: GlobalPlatform TEE Internal API
3. TA Management Framework: TEEP(Trusted Execution Environment Provisioning)
4. Remote Attestation

Remote Attestation

- Remote attestation offers
 - Platform authentication
 - Platform integrity
 - Binary integrity
- Remote attestation is achieved before the execution of TA and keeps the safe execution of TA on the TEE.
- Remote attestation assumes
 - On Edge (platform)
 - Keys or certificates protected by hardware, i.e., **Root of Trust**.
 - On Sever (verifier)
 - Data base for hash of TA, Device Pub-Key

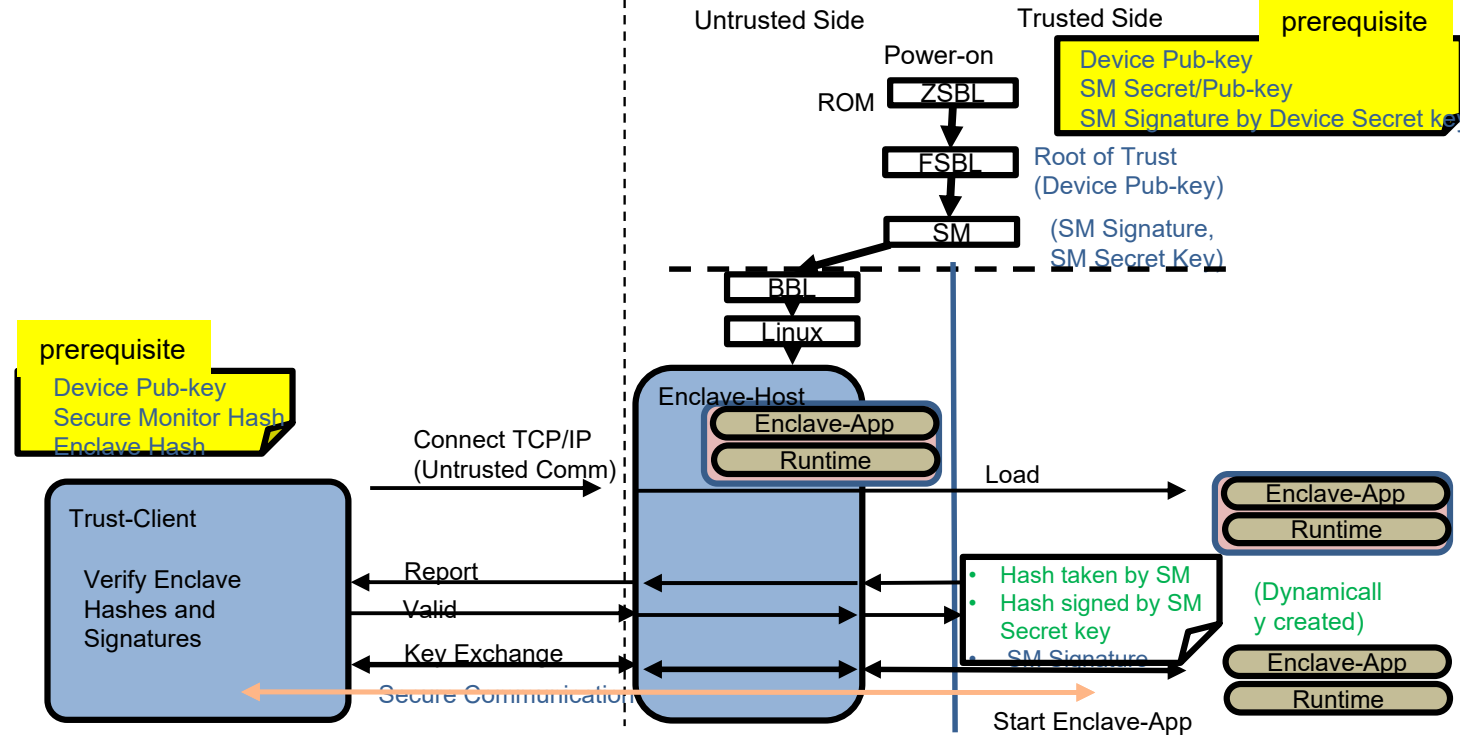
Customized Remote Attestation

- We reuse the Keystone's Remote Attestation.
 - Trusted RV keeps a device key at secure unit.



Remote Attestation Server

Trusted-RV Client (AppCore)



Future Work

- We have developed the infrastructure of RISC-V TEE hardware/software.
- Next step is creation of PoC(Proof of Concept) for real usage.
 - Server
 - Code and Data hiding for Machine Learning
 - Edge
 - Smart city

Conclusions

- Current TEE has some issues and mitigated by Security Technologies offered by TRASIO
 - Hardware
 1. Trusted-RV Platform (64-bit RISC-V + 32-bit RISC-V Secure CoProcessor)
 - Software
 2. TEE's Programming Environment: GlobalPlatform TEE Internal API
 3. TA Management Framework: TEEP(Trusted Execution Environment Provisioning)
 4. Remote Attestation

Acknowledgement: This presentation is based on results obtained from a project (JPNP16007) commissioned by the New Energy and Industrial Technology Development Organization (NEDO).