TRUSTONIC

Trusted Execution Environment (TEE)
Introduction aux environnements d'exécution de confiance

Johan Amiard

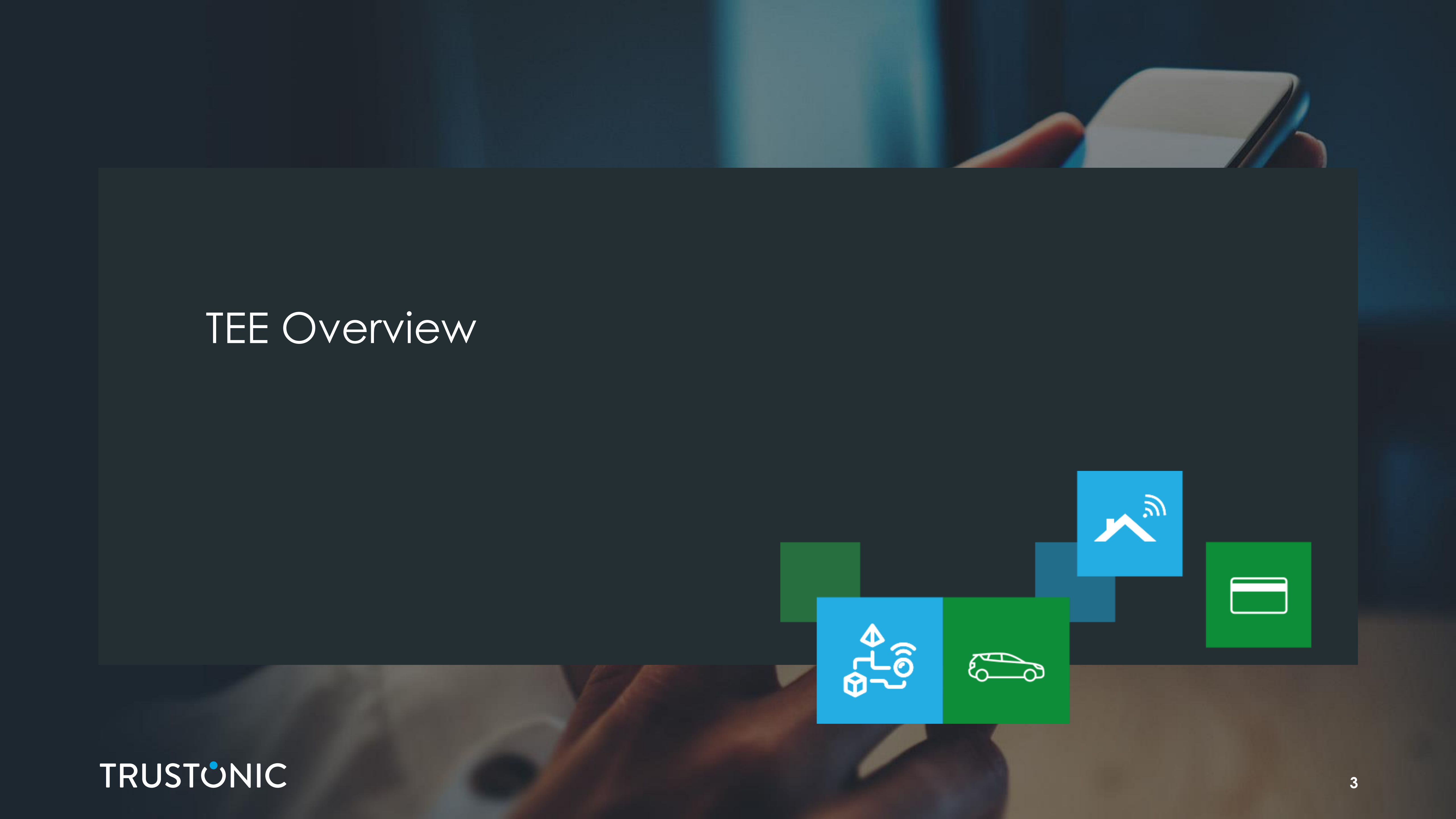
September 20th, 2018

TEEVA - GDR SoC2 – Security (Paris)

Agenda

- TEE Overview
- Use Cases (Examples)
- Introduction to GP API

TEE Overview



All about Trustonic

Governance

- Founded in 2012. Strategic Investors

arm

gemalto
security to be free

Twin Mission

- To embed the best security into the world's smart devices
- To empower app developers to deliver simpler, richer, safer services

GSMA GLOMO AWARDS
THE GLOBAL MOBILE AWARDS

Credentials

- Protect >1 Billion smart devices
- Recognized leader in application security
- GSMA 2016 Award Winner – Best mobile security solution
- Underpins security in services including

SAMSUNG pay

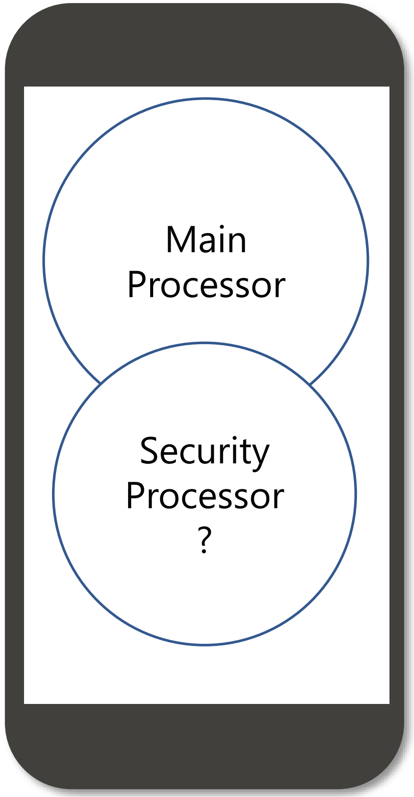
Symantec.

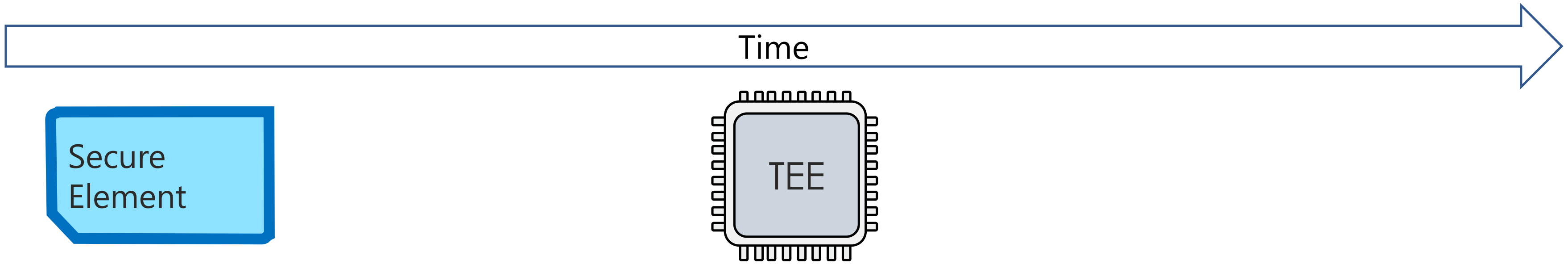
Alipay™

Samsung Knox

LINE

Evolution of Hardware Security





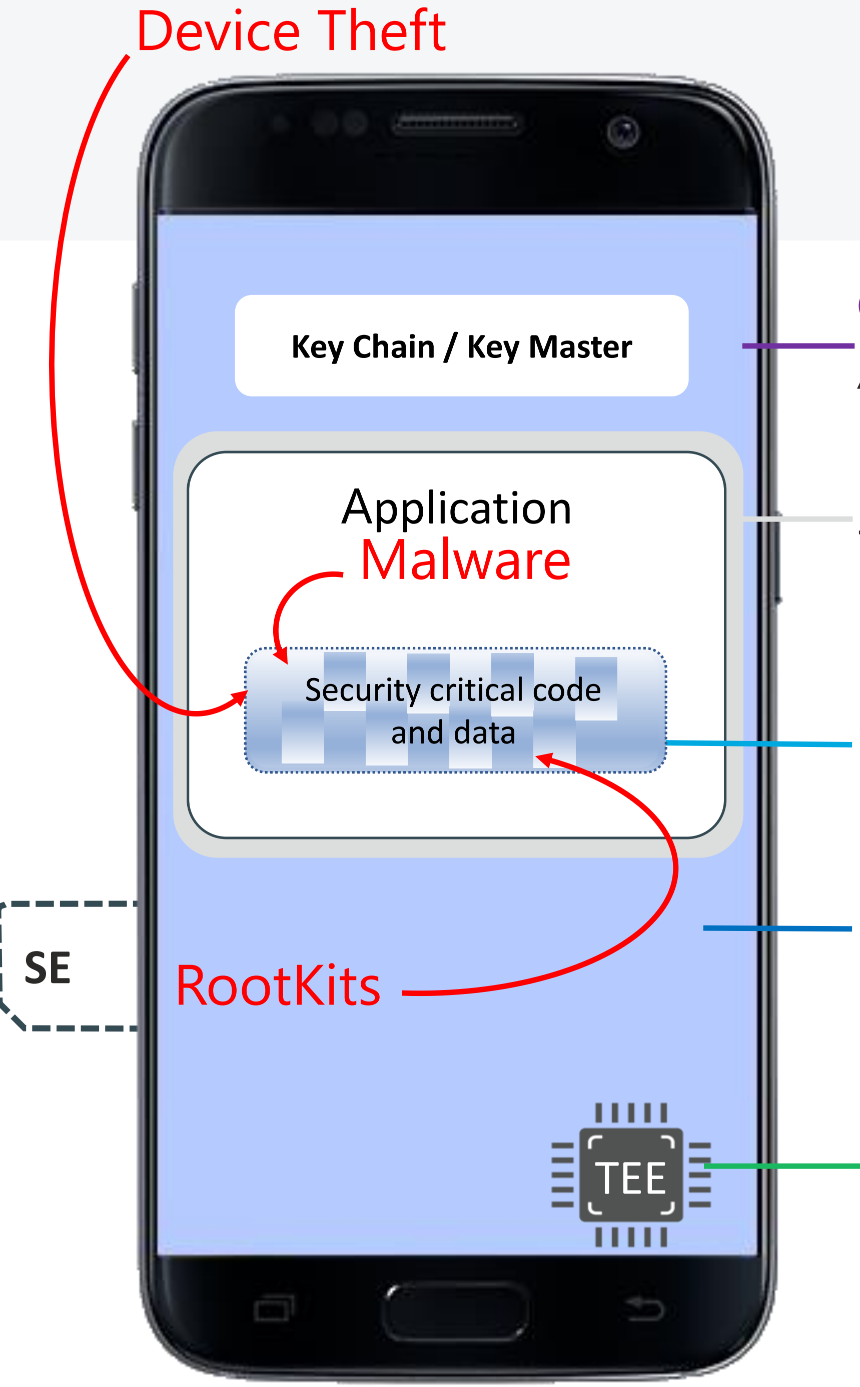
Secure Element

- **Small chip (smartcard)**
- Low Power / Performance
- **Original Focus:**
- Protect GSM network
- **Extended Remit:**
- Apps from Network Operator

TEE

- **'Mode' in ARM Processor**
- Full power / performance
- **Intent: General Purpose Security**
- OS Features
- Preinstalled OEM Apps (e.g. Samsung Pay)
- Third Party Apps (App Store support)

Security Risks and Developer Options



Option 1: **Leverage OS Security Capabilities**
Android/iOS provide basic key storage

Option 2**: App Wrapping**
Targeted at enterprise as post build fix
Focused on data storage / enterprise unlock

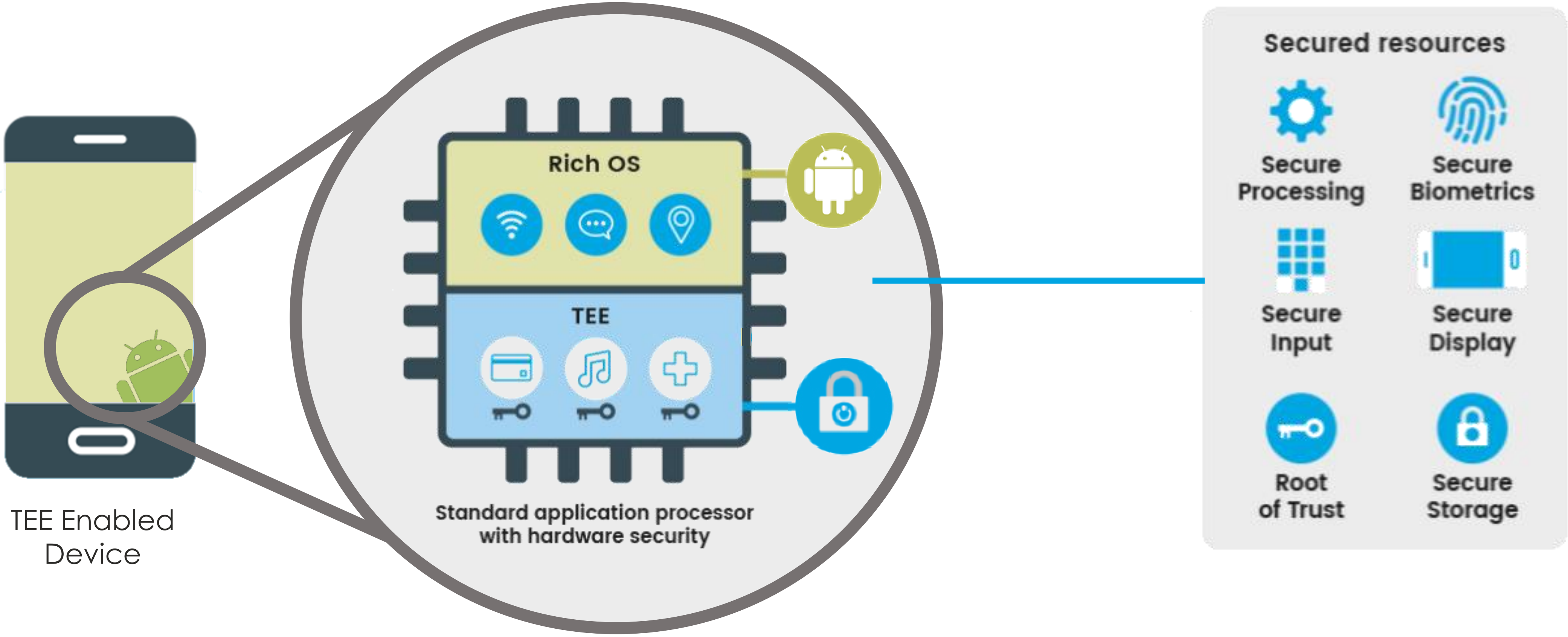
Option 3: **Software protection**
Isolate and obfuscate security critical parts of code.

Option 4: **Run security code in a Secure Element**
Limited processing power / capabilities
Generally only accessible to MNO / OEM

Option 5: **Run code in Trusted Execution Environment**
Hardware isolated 'slice' of main CPU, with secure OS
Only accessible to OEM, *except with Trustonic TEE*

Trustonic TEE

Hardware security for critical applications, on 1 billion devices



TEE Enabled Device

Rich OS

TEE

Standard application processor with hardware security

Secured resources

- Secure Processing
- Secure Biometrics
- Secure Input
- Secure Display
- Root of Trust
- Secure Storage

An Introduction to ARM TrustZone (1/2)

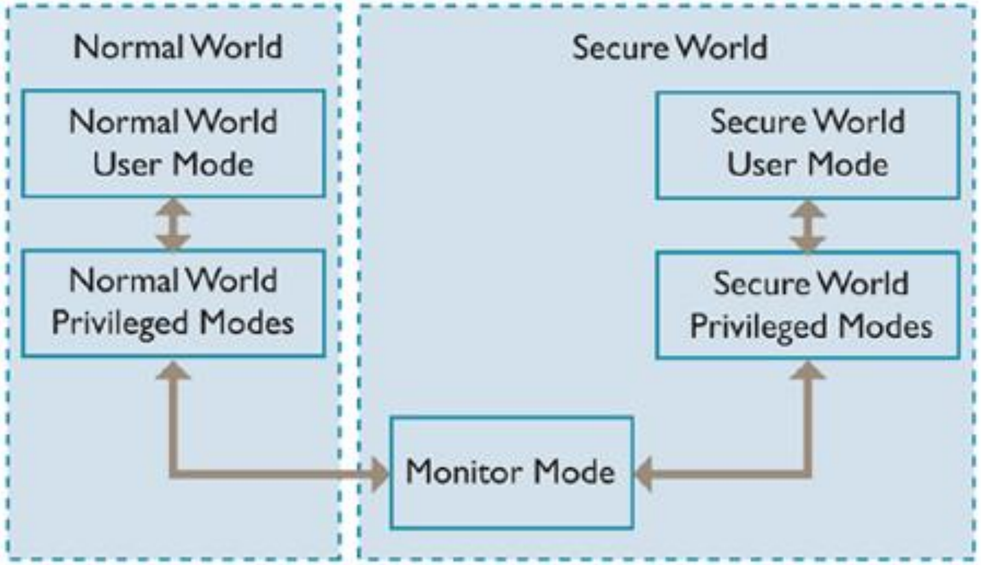
- Feature available from ARM1176, in every Cortex-A processors
- Devices developed with TrustZone technology enable the delivery of platform capable of supporting full Trusted Execution Environment
- This allows splitting the system in 2 states
- TrustZone enables the development of separate Rich Operating System and Trusted Execution Environments by creating additional operating modes to the Normal domain, known as the Secure domain and the Monitor mode

An Introduction to ARM TrustZone (2/2)

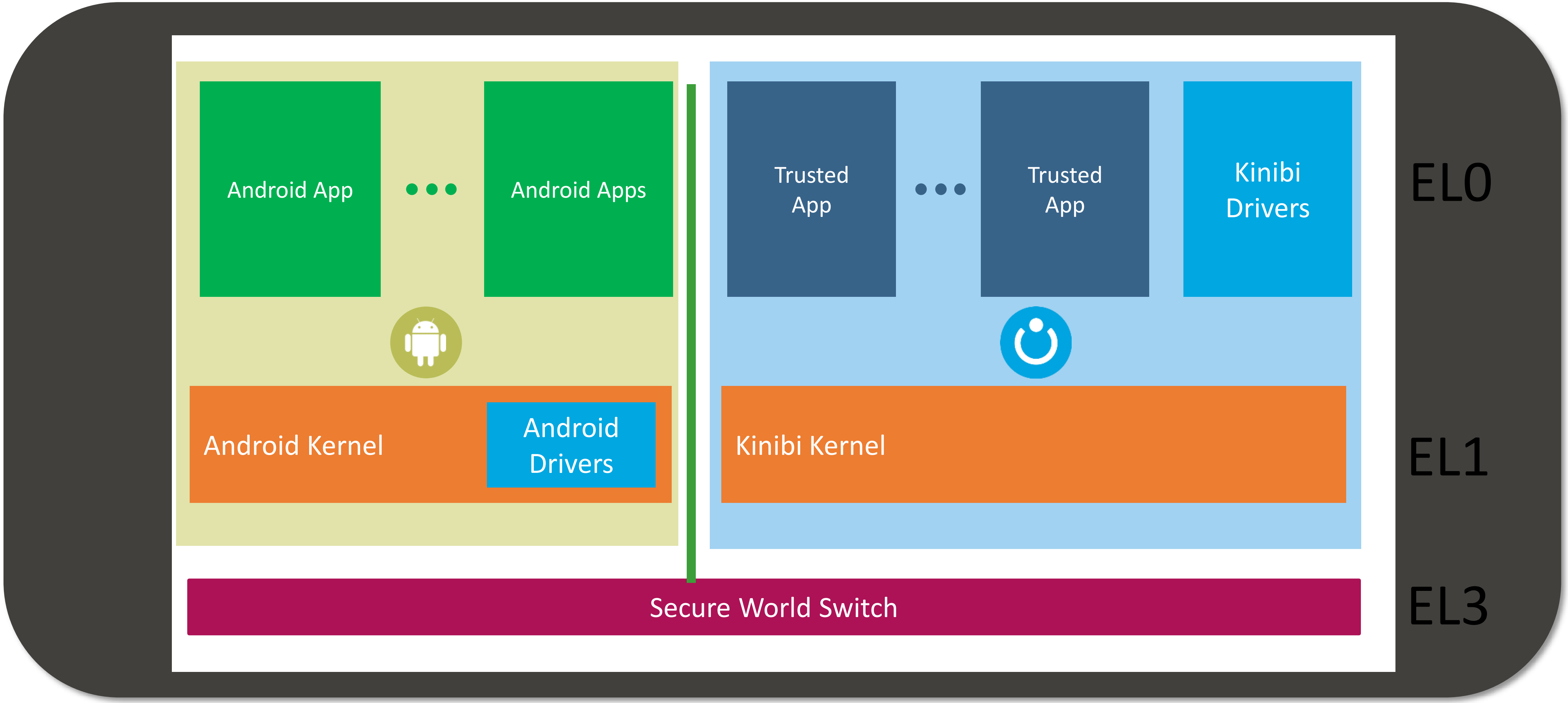
- NS bit added a new state to the processor
  - NS bit = 1 → Non-Secure State
  - NS bit = 0 → Secure State

- NS bit can be propagated to peripherals

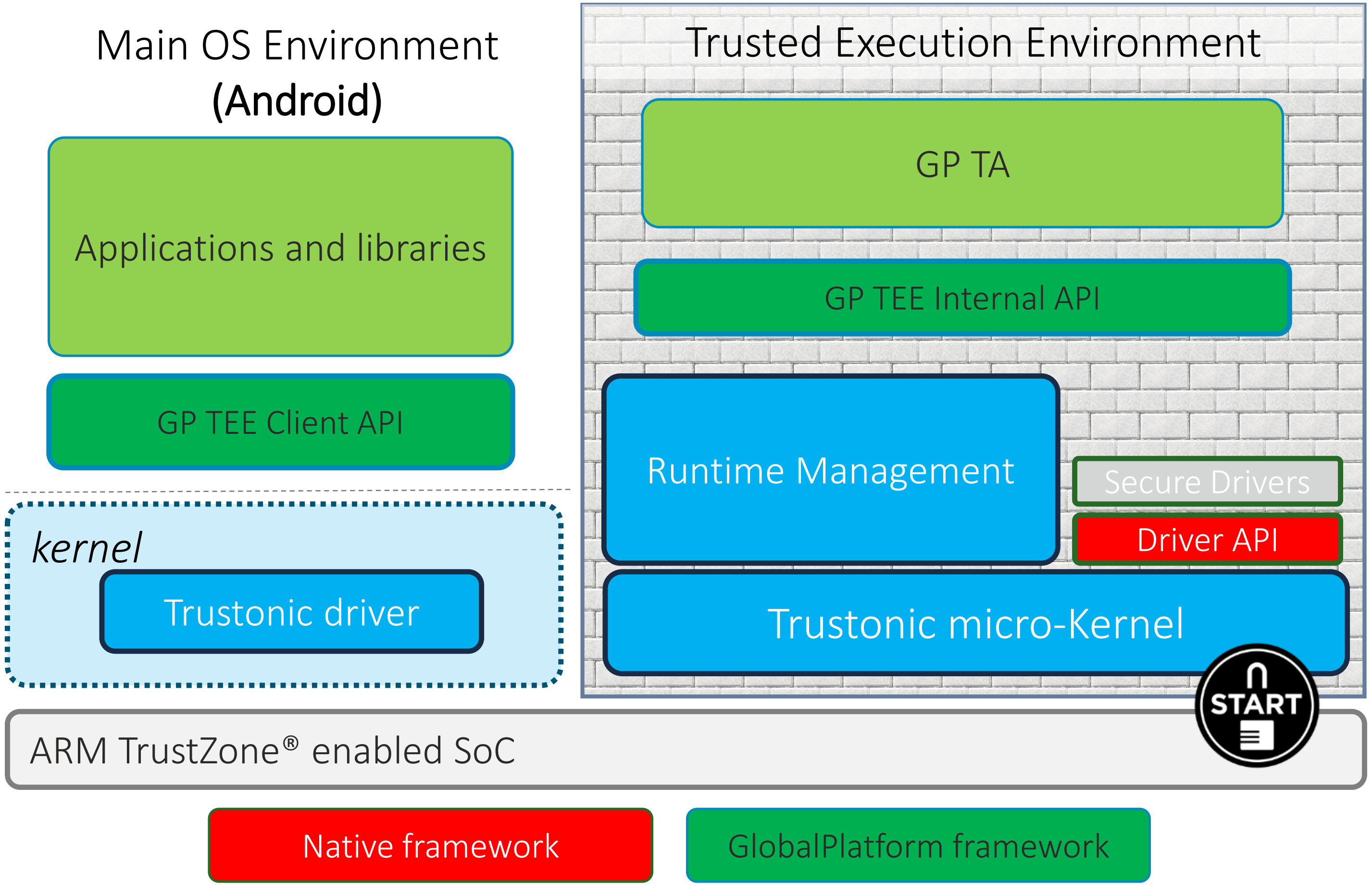
- New Monitor mode
  - To manage transitions between Non-Secure and Secure States
  - Always in Secure State whatever the NS bit



High Level Architecture



Kinibi TEE - High Level Architecture



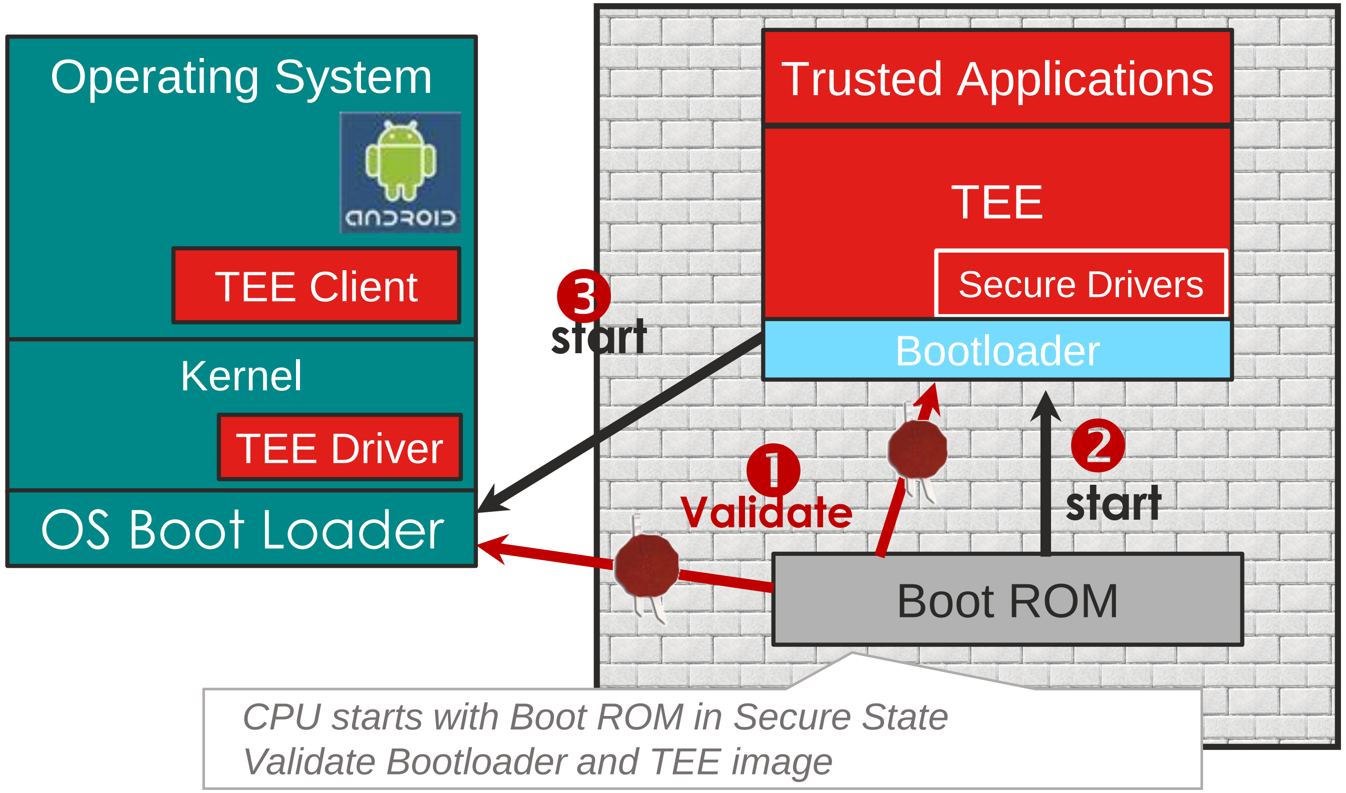
Memory Separation

- Each process and TA has its own virtual address space, enforced by MMU

Power Management compliant

- Supports save and restore of secure memory upon power management events

Typical Secure Boot Sequence



CPU starts with Boot ROM in Secure State
Validate Bootloader and TEE image

- The Boot ROM validates the Secure Boot Loader and OS Boot loader
- The Boot ROM starts the Secure Boot Loader
- The Secure Boot Loader validates & starts the TEE
- The Secure Boot Loader starts the OS Boot Loader
- The OS Boot Loader validates & starts Android

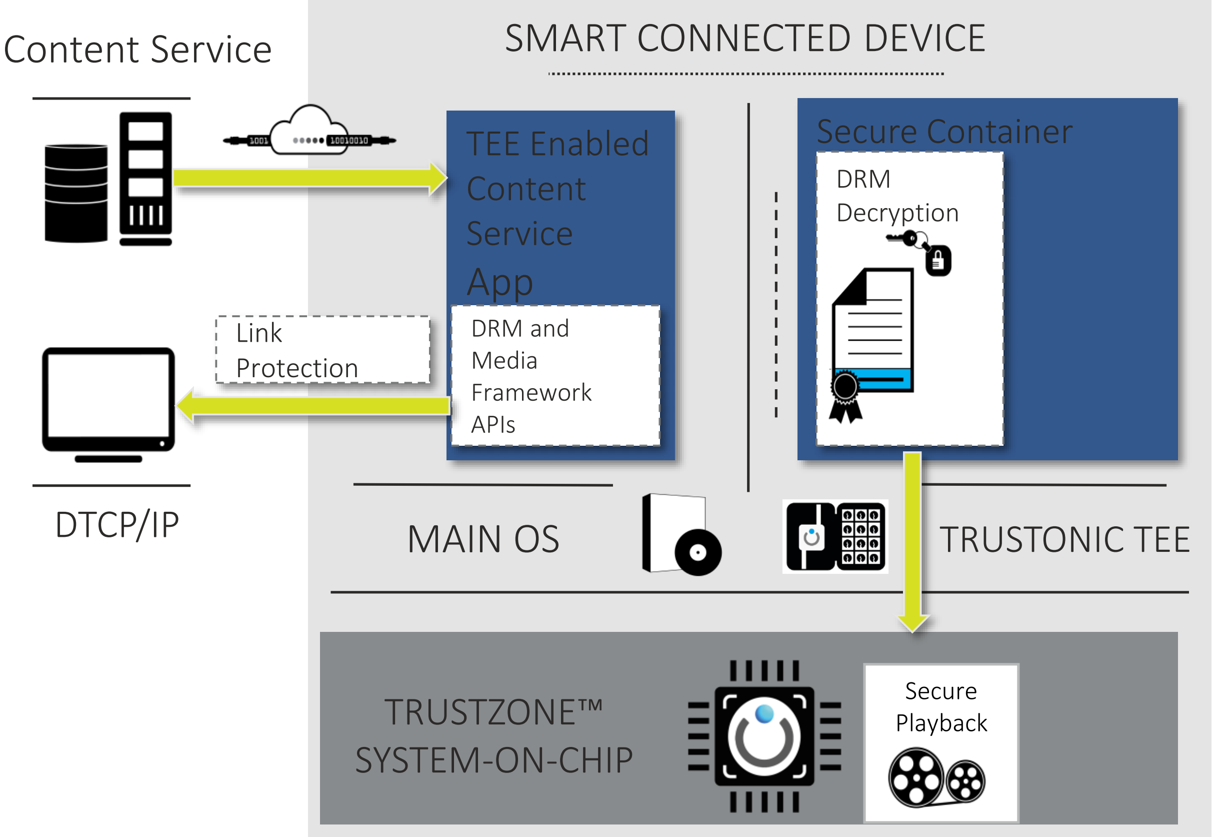
All security is a weakest link problem - A chain is only as strong as its weakest link

TEE – Other features

- Normal World – Secure World communication
  - Based on shared memory mechanisms
  - Optimized for zero-copy data transfer
- Pre-emptive micro-kernel architecture
  - Does not block the Normal World OS
- Custom Secure Drivers
  - OEMs can develop their own Secure Drivers through the DDK
- **Fully GP Compliant** – Client API and Internal API
  - Cryptographic processing with major algorithms support
  - Data wrapping for persistent secure storage
  - Arithmetic API
  - And much more …

TEE Use Cases (Examples)

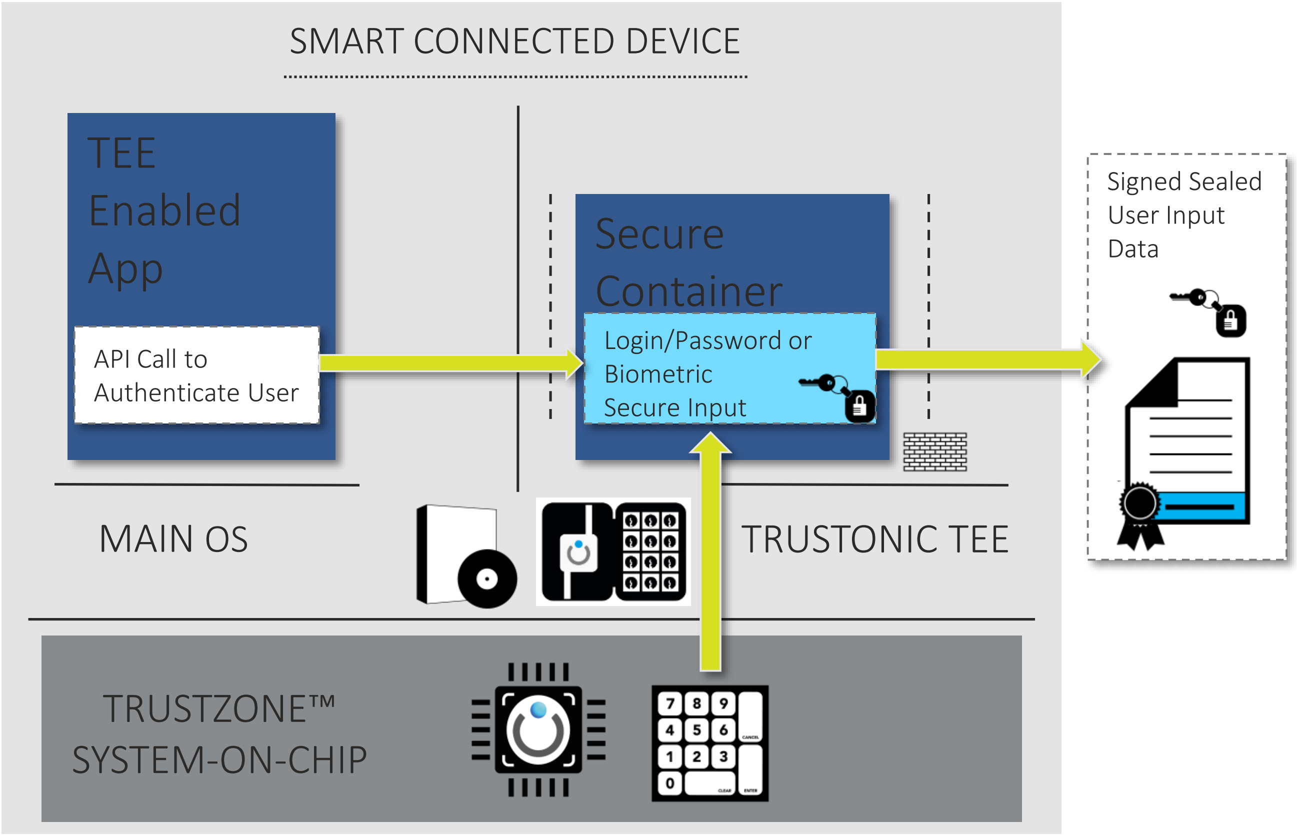
Content Protection



- Secure Boot
- Device Authentication
- User Authentication
- DRM Protection
- Trusted time source
- Secure Playback
- Link Protection DTCP
- Downloadable Scheme

Trustonic protects video path from studio to user

Trusted Display & User Input Capture


SMART CONNECTED DEVICE
TEE Enabled App
API Call to Authenticate User
Secure Container
Login/Password or Biometric Secure Input
Signed Sealed User Input Data
MAIN OS
TRUSTONIC TEE
TRUSTZONE™ SYSTEM-ON-CHIP
7 8 9
4 5 6
1 2 3
0

Trustonic protects PINs and Passwords

Multiple markets, multiple use cases

Identity
Authentication, Identity, Storage

Premium Content
DRM, secure decryption

Financial Services
mPayments, mBanking, mPOS

Internet of Things
Automotive, Industrial…

Enterprise/Gov't
Secure voice & data messaging

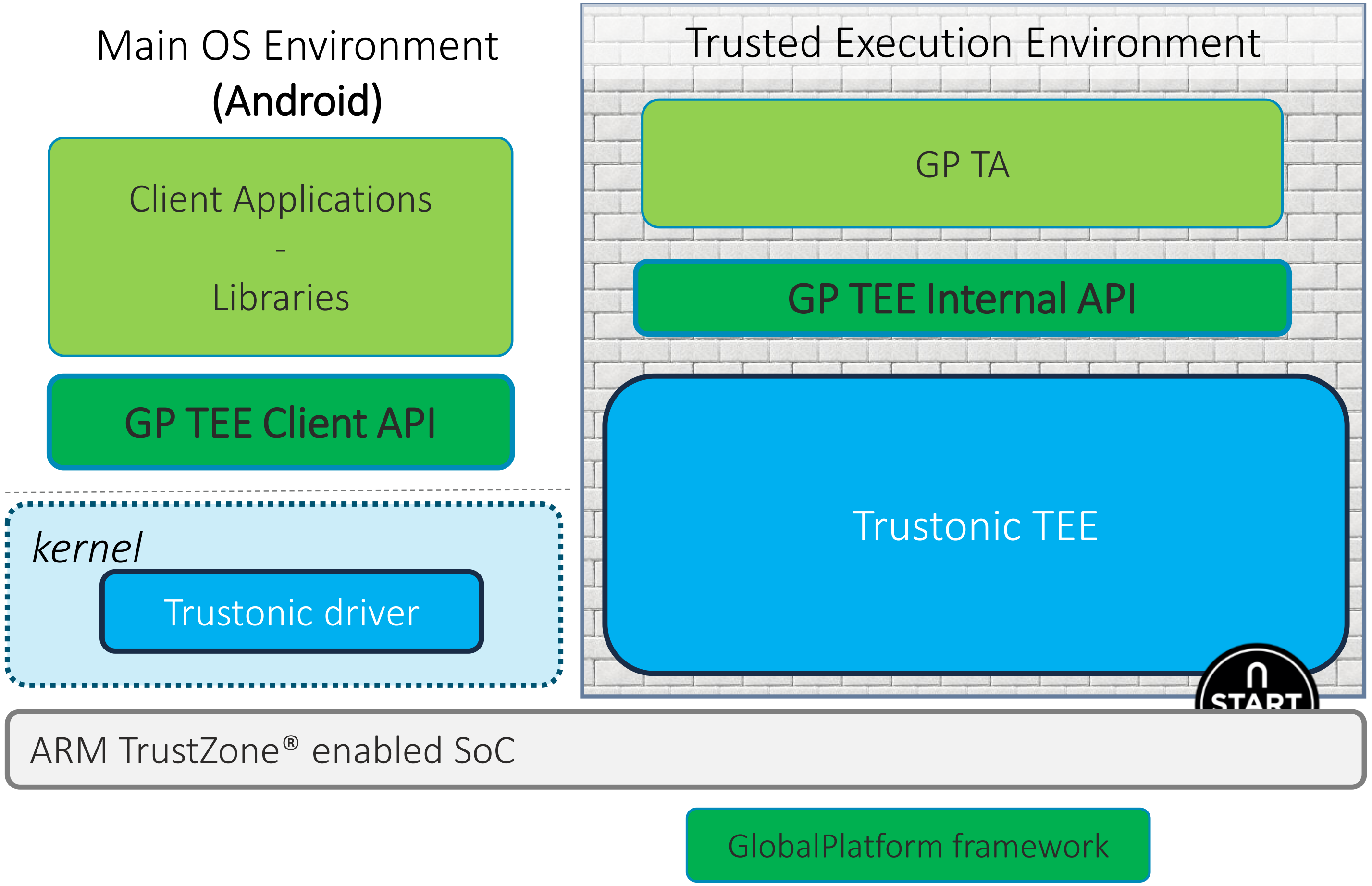
Mobile Network Operators
Device integrity, Subsidy protection
Identity verification …

Introduction to the GP (GlobalPlatform) API

GP (GlobalPlatform) API

- **TEE Client API** for client applications

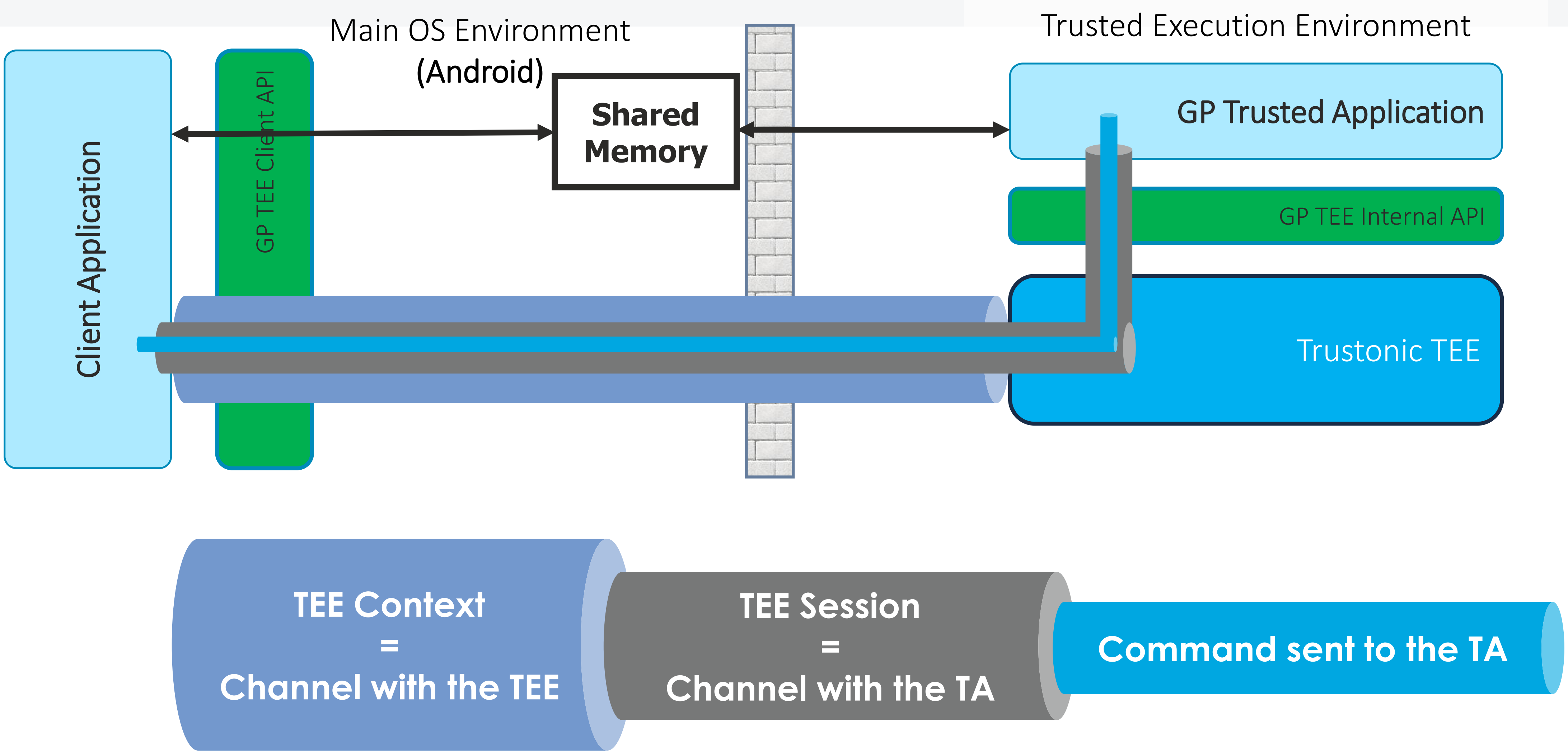
- **TEE Internal API** for TA
  - Memory Management
  - Time Management
  - Properties Management
  - Inter TA Communication
  - Cryptography
  - Trusted Storage
  - Arithmetic



GP (GlobalPlatform) API

- **TEE Context**
  - Logical connection between Applications and TEE
- **Sessions**
  - Logical connection between Applications and Trusted Applications
- **Commands and Responses**
  - With Command ID and payloads
- **Shared Memory**
  - Used for efficient data exchange between applications and secure services

GP Client API Channels



TEE Client API

The API from the Client Side is very simple:

- TEEC_Initialize/FinalizeContext → Link with the TEE
- TEEC_Open/CloseSession → Link with a TA with a **Login** method
- TEEC_InvokeCommand
  - Send a commandID to TA with optional input/output parameters
  - Parameters can be 32 bits or a shared memory reference

- TEEC_Allocate/Register/ReleaseSharedMemory

A protocol must be defined between the Client and the TA:

- ➢ List of Command IDs
- ➢ Input/Output Parameters associated with each command ID

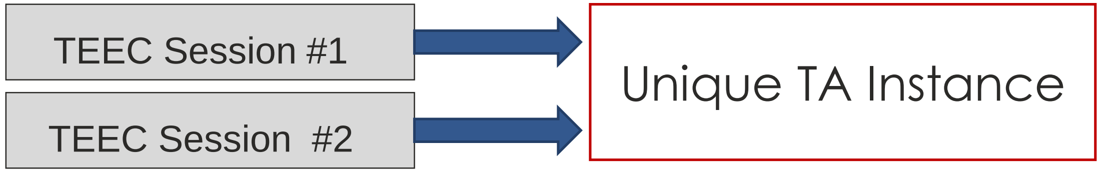
Trusted Application Interface

Those entry points must be implemented by every Trusted Application:

- TA_Create/DestroyEntryPoint
- TA_Open/CloseSessionEntryPoint
- TA_InvokeCommandEntryPoint

Link between TEEC (TEE Client API) and TA – *Example for a Mono-Instance TA*

TEE Client API	*trigger*	TA Entry Point
		TA_CreateEntryPoint (*At TEE start or first OpenSession*)
TEEC_OpenSession	→	TA_OpenSessionEntryPoint
TEEC_InvokeCommand	→	TA_InvokeCommandEntryPoint
TEEC_CloseSession	→	TA_CloseSessionEntryPoint
		TA_DestroyEntryPoint



TEE Internal APIs (Core)

Properties Access Functions

- Access properties of the TA itself, the client, or the TEE implementation

Memory, Panic, Cancellation

- Allocation, MemMove, Compare, Fill, Check Memory Access Rights
- Panic will stop the TA in a proper way

Time management

- Set/Get TA time (reference), Get REE Time, Wait

Internal Client API

- Communication with another TA or with a Secure Driver
- Same mechanism as Client ↔ TA: Open/CloseSession, InvokeCommand

Arithmetical API

- Provides building blocks to implement missing asymmetric algorithms

TEE Internal APIs (Cryptographic and Trusted Storage)

One API to manage Cryptography and Trusted Storage

- Based on object handles – Objects can be transient (memory only) or persistent
- A persistent object can be a Cryptographic Key object, a Cryptographic Key-Pair object, or a Data object (raw data)
- There is a different lifecycle for transient and persistent objects
- **API for Object manipulation**

- There is a lifecycle for Cryptographic operations
- Can manage single or multi-stage operations
- **API for Cryptographic operations**

TRUSTONIC

Thank you.

Johan Amiard
TEEVA Project Manager
Johan.amiard@trustonic.com