Trusted Execution Environment: what it is and what it is not

Mohamed Sabt [1,2] Mohammed Achemlal [1,3]
Abdelmadjid Bouabdallah [2]

[1]Orange Labs, France

[2]Sorbonne Universités, UTC, France

[3]Ensicaen, France

Trustcom 2015, August 2015

Outline

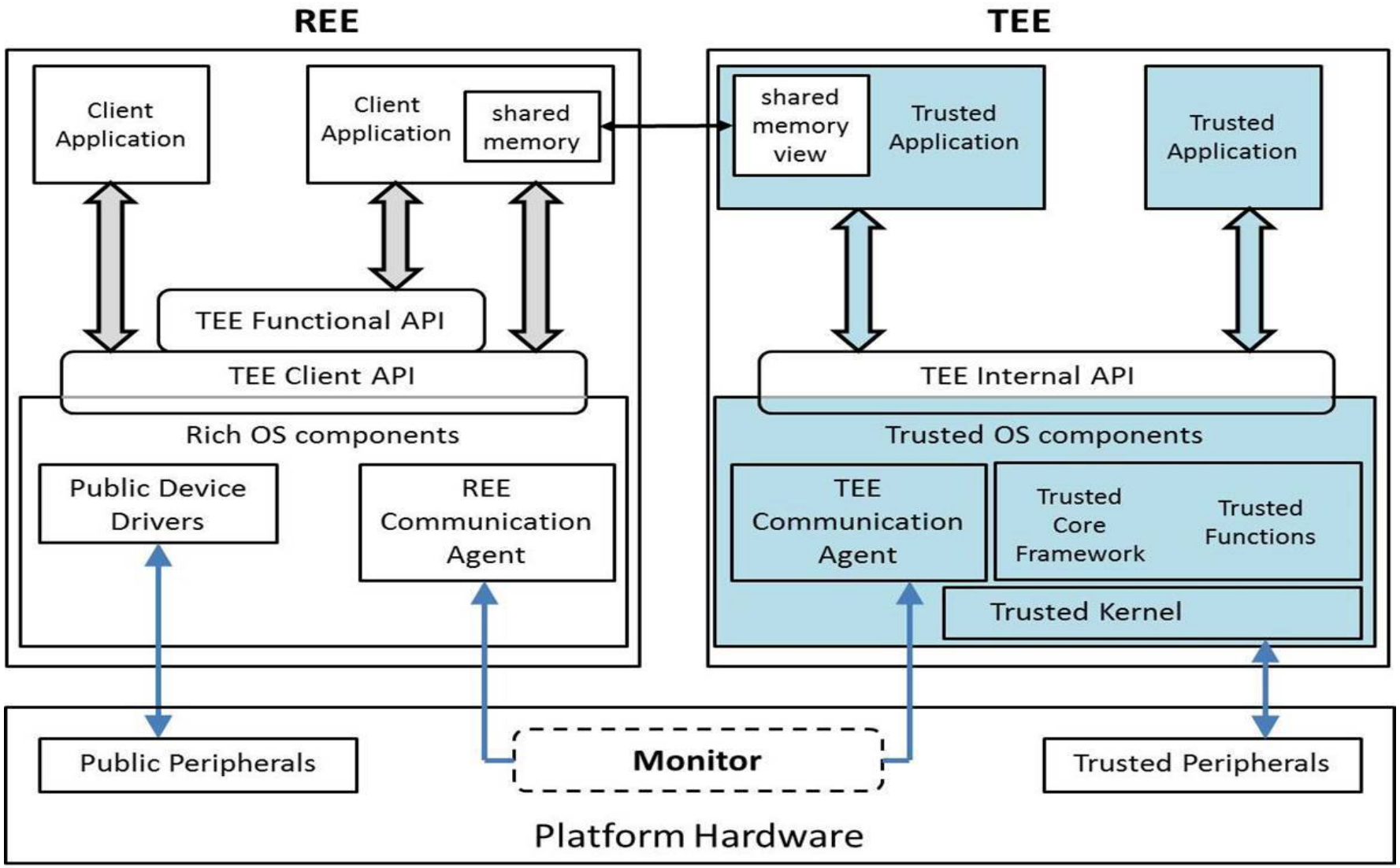
"The time has come," the Walrus said,
"To talk of many things;
Of shoes–and ships–and sealing wax–
Of cabbages—and kings–
And why the sea is boiling hot–
And whether pigs have wings."

— Lewis Carroll, Through the Looking-Glass

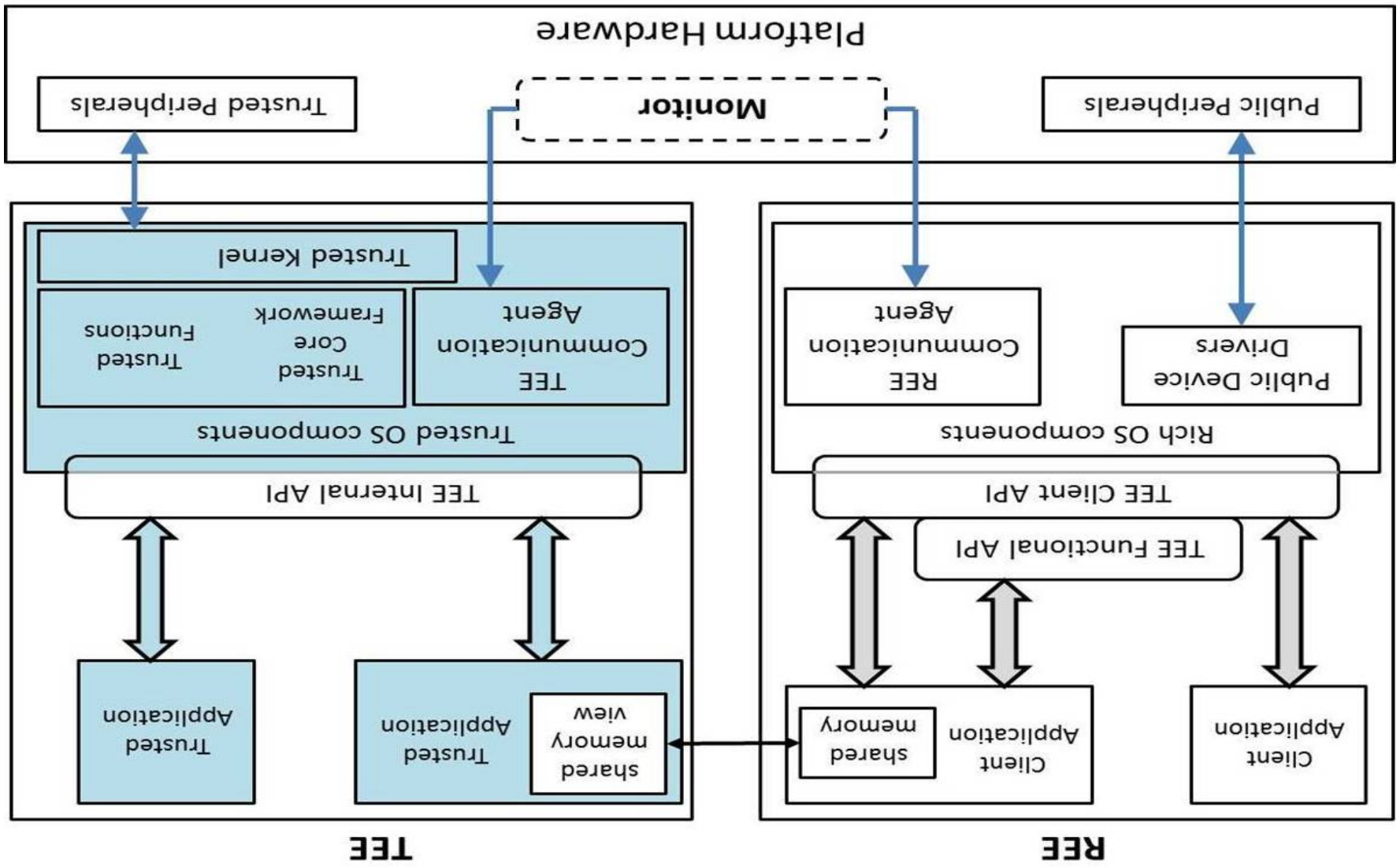
Ubiquitous TEE

- Virtual TPM [Trust 2009];
- Privacy-Preserving Mobile Payment [TrustCom 2012];
- Sensors [MobiSys 2012];
- Two Factor Authentication [NDSS 2014];
- Introspection [CCS 2014];
- Software Obfuscation [ARES 2014];
- Mobile Advertisement [MobiSys 2015];
- Autonomic Systems [ICAC 2015].

Once upon a time



From different directions



Come, listen, my men, while I tell you again
The five unmistakable marks
By which you may know, wheresoever you go,
The warranted genuine Snarks.

— Lewis Carroll, The Hunting of the Snark

Let's start at the very beginning...

Practical Need

Execute a highly-sensitive app on an off the shelf smartphone.

Let's start at the very beginning...

Practical Need

Execute a highly-sensitive app on an off the shelf smartphone.

Problem to solve

Secure an application in the presence of a malicious OS.

Let's start at the very beginning...

Practical Need

Execute a highly-sensitive app on an off the shelf smartphone.

Problem to solve

Secure an application in the presence of a malicious OS.

Existing Solutions

- special hardware processor: AEGIS and CHERI;
- micro-kernel: SeL4;
- separation kernel: TLR.

Let's start at the very beginning...

Practical Need

Execute a highly-sensitive app on an off the shelf smartphone.

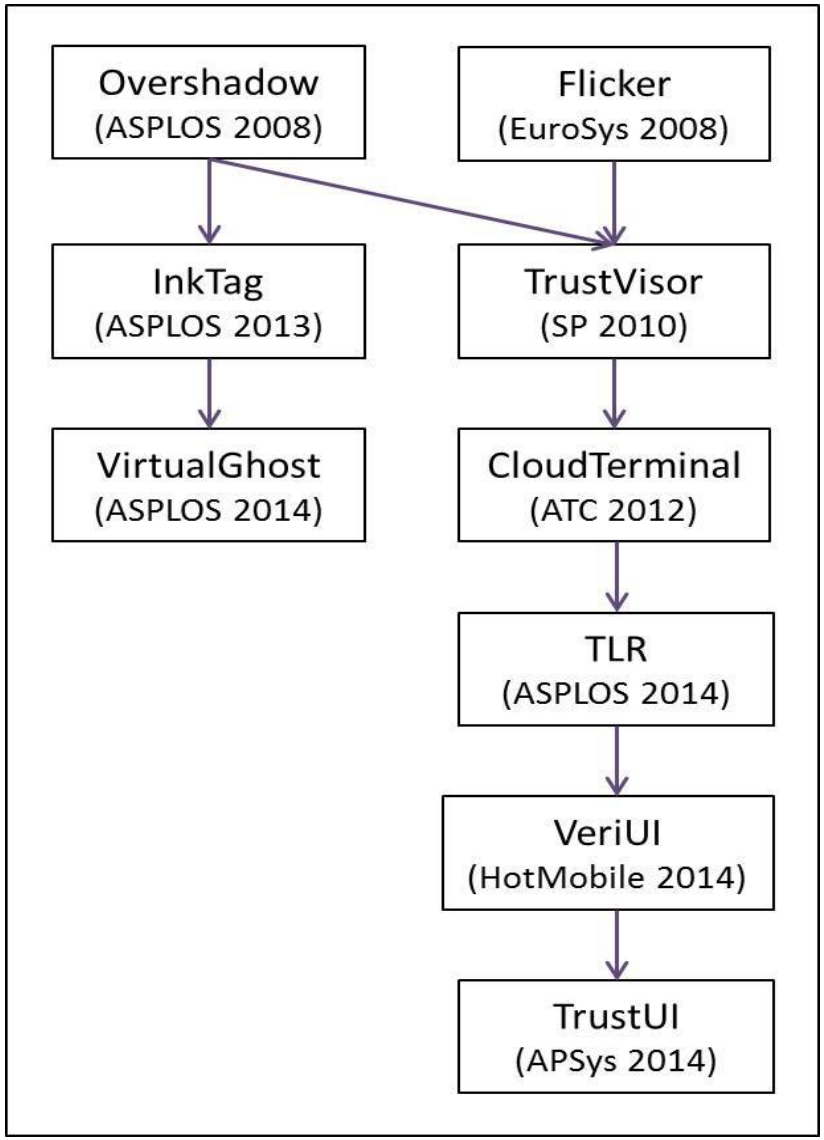
Problem to solve

Secure an application in the presence of a malicious OS.

Existing Solutions

- special hardware processor: AEGIS and CHERI;
- micro-kernel: SeL4;
- separation kernel: TLR.

State of the art



Definitions

Separation Kernel

Security kernel that enables the coexistence of different systems requiring different levels of security on the same platform.

Definitions

Separation Kernel

Security kernel that enables the coexistence of different systems requiring different levels of security on the same platform.

Secure Execution Environment

Processing environment that guarantees the following properties:

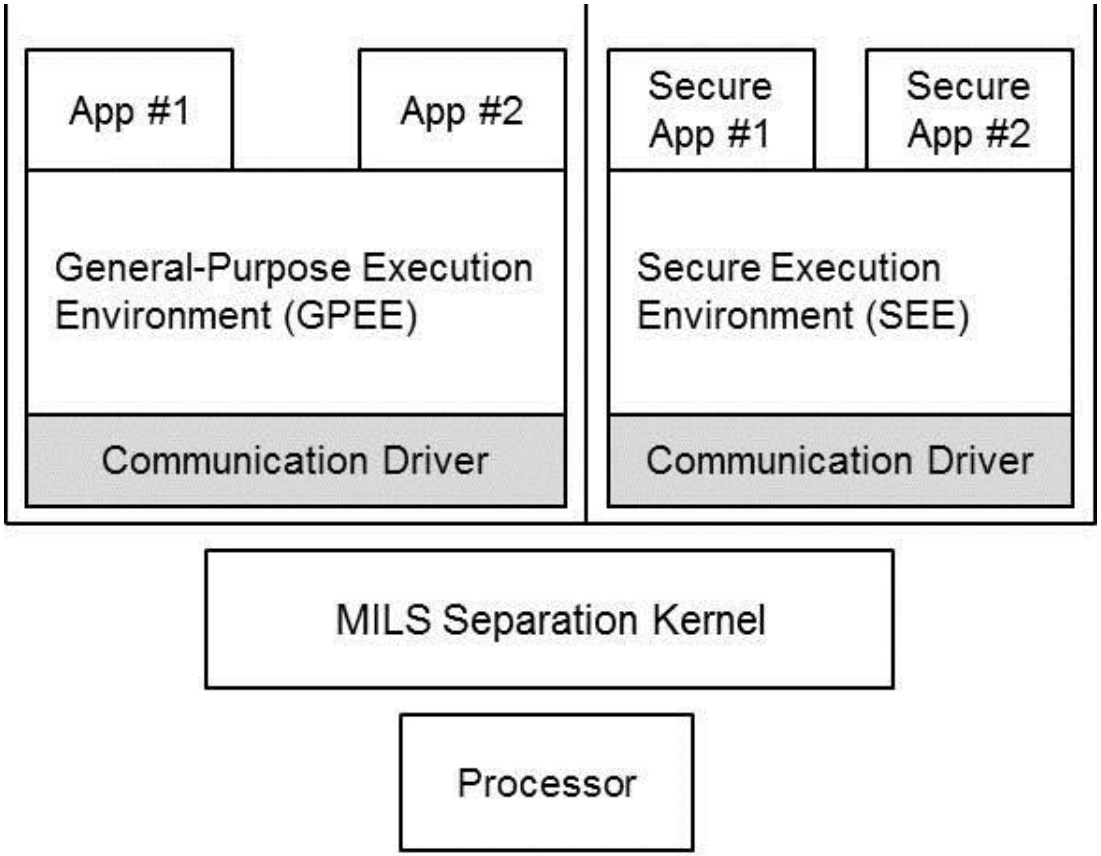
- *authenticity*: the code under execution should not have been changed;
- *integrity*: runtime states should not have been tampered with;
- *confidentiality*: code, data and runtime states should not have been observable by unauthorized applications.

The Dual-Execution-Environment Approach

Definition

The Dual-Execution-Environment is a security architecture where both a Separation Kernel and a Secure Execution Environment play the role of Security Kernel. As its name indicated, only two processing environments run above the defined separation kernel.

Overview


App #1
App #2
General-Purpose Execution Environment (GPEE)
Communication Driver
Secure App #1
Secure App #2
Secure Execution Environment (SEE)
Communication Driver
MILS Separation Kernel
Processor

Isolation

1. **Data separation**: Data within one partition cannot be read or modified by other partitions;
2. **Sanitization**: Shared resources cannot be used to leak information into other partitions;
3. **Control of information flow**: Communication between partitions cannot occur unless explicitly permitted;
4. **Fault isolation**: Security breach in one partition cannot spread to other partitions.

Inter-EE Communication

1. **Reliability**: memory/time isolation;
2. **Minimum overhead**: unnecessary data copies and context switches;
3. **Integrity**: protection of communication structures.

Inter-EE Communication

1. **Reliability**: memory/time isolation;
2. **Minimum overhead**: unnecessary data copies and context switches;
3. **Integrity**: protection of communication structures.

Attacks

- message overload attacks;
- control data corruption attacks;
- memory faults caused by shared pages being removed.

Secure Scheduling

1. **Time-slicing**: balanced sharing of the hardware resources;
2. **Preemptive**: mixing the priority level of the GPOS and SEE activities.

Secure Scheduling

1. **Time-slicing**: balanced sharing of the hardware resources;
2. **Preemptive**: mixing the priority level of the GPOS and SEE activities.

Attacks

- non-responsive event-driven operations;
- unbound waits caused by the non-cooperation of the untrusted part of the system.

"When I use a word," Humpty Dumpty said, in rather a scornful tone, "It means just what I choose it to mean–neither more nor less."

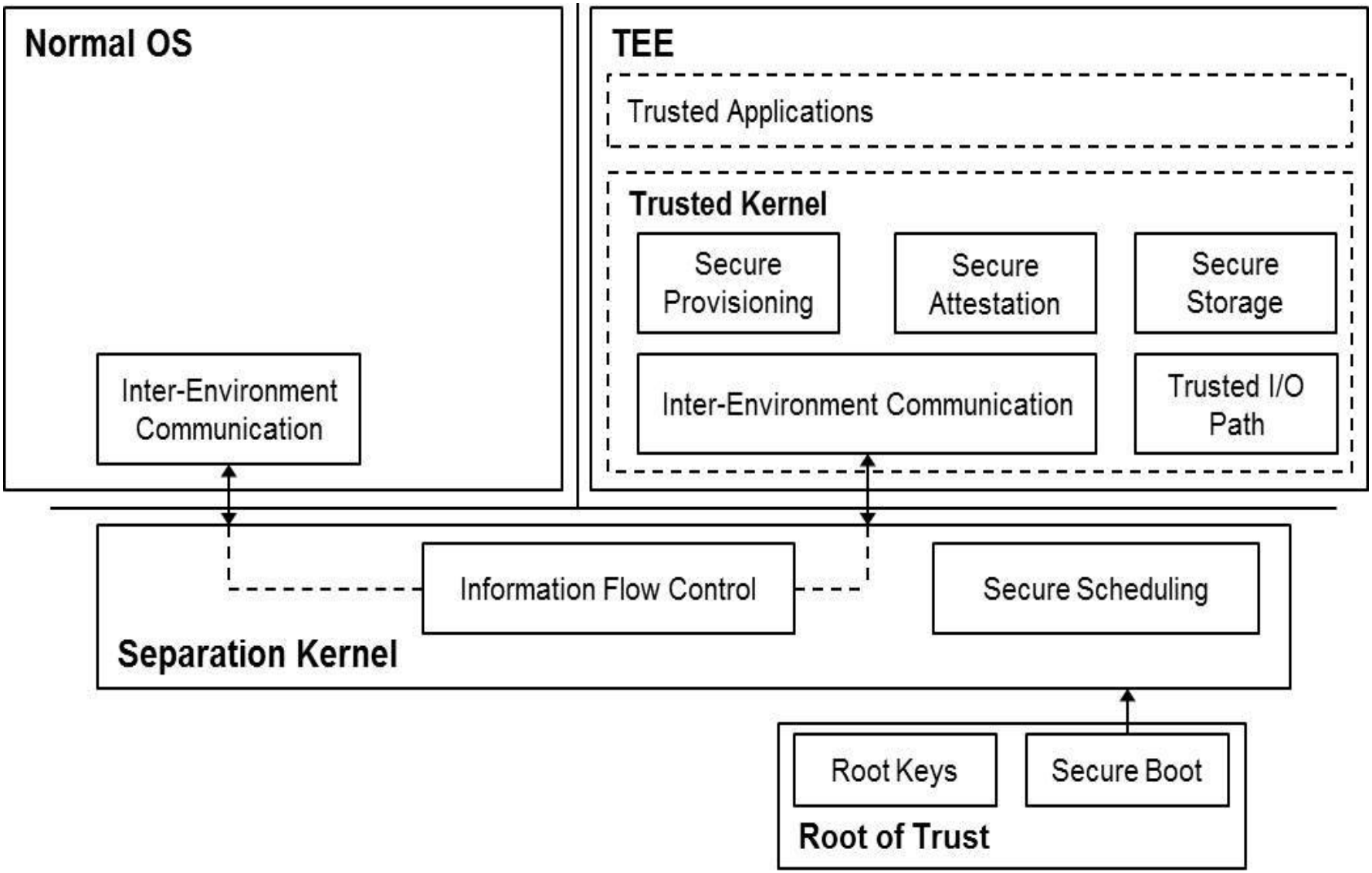
— Lewis Carroll, Through the Looking-Glass

The War of Terminologies

The Existential Question

Does the **TEE** exist before the existence of the *TEE* ?

Building Blocks



Attacks

Adversary model

a powerful attacker who is able to execute an arbitrary code in the kernel privileges.

Attack Classes

- bypassing security features;
- executing arbitrary code in the secure zone;
- overwriting part of the secure region of the memory with certain values;
- Denial-of-service attacks are not included.

Small Survey

TEE	License	Normal World	Hardware Platform
ObC	Close	Symbian OS	300 MHz OMAP 2420
<t-base	Close	Android	Samsung Exynos platforms
Andix OS	Open	Linux	iMX53 QSB
TLK	Open	Android	Tegra SoCs
TLR	Close	.NET CLR	Tegra 250 Dev Kit
SafeG	Open	TOPPERS/ASP	PB 1176 JZF-S board

Summary

- TEE is a promising security technology;
- Formal model can be defined with the Dual-EE approach;
- TEE could be shown as a security architecture, rather than just a technology in order to have enough theoretical basis to answer basic issues related to TEE.

Building Blocks

Thank you for your attention!