Mobile Platform Security:
OS Hardening and Trusted Execution Environment

Onur Zengin
Lead Security Engineer @ TRUSTONIC

Agenda

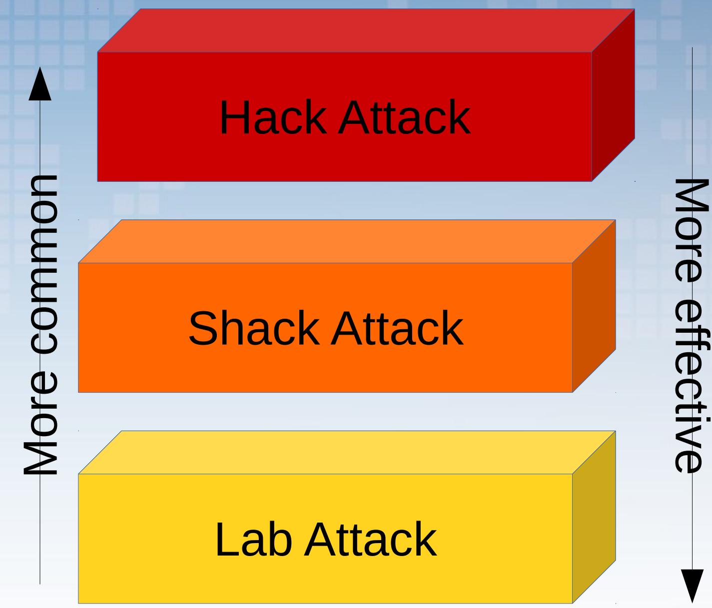
Attack Types on a Platform
Platform Security Components
Operating System Security
OS Security: Linux
Questions on OS security
Motivation for Isolation
HW Components for Secure Code Execution
Trusted Execution Environment
TEE Services and Use Cases (Authentication, RKP, Key Management, DRM and SIM Lock)

Attack Types on a Platform

Hack Attack
- Attacking via software using malwares, spywares and vulnerabilities
- Does not need hands on the device. Remotely done and effective on masses.

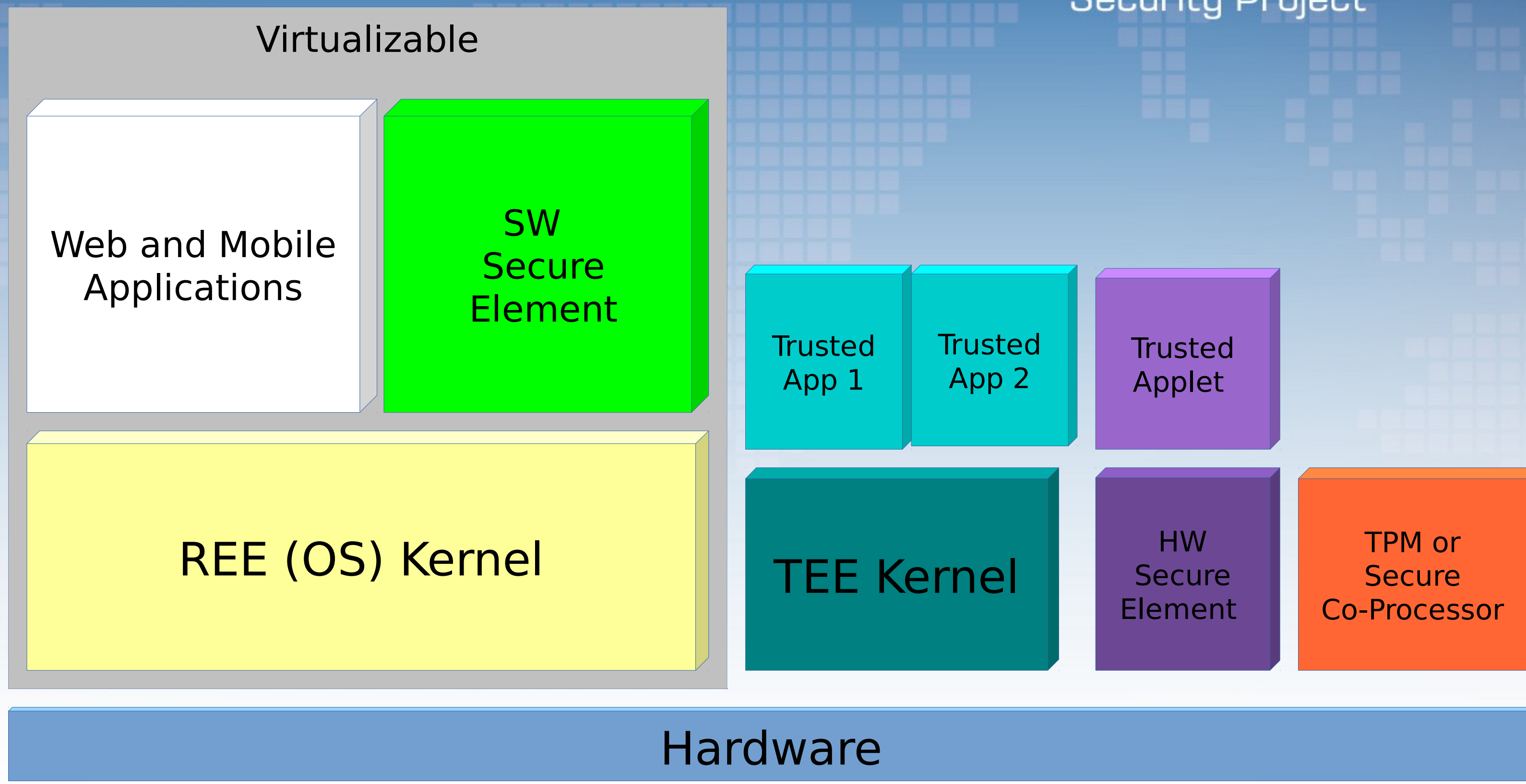
Shack Attack
- Targets certain platforms using vulnerabilities with complex attack techniques. Sometimes requires proximity to the device. Attack cost does not include hardware tampering.
- Some side channel attacks are used as shack attacks.

Lab Attack
- Requires possession of the device and a facility to perform the attack. The costliest and most complex of the attacks. Bus probing etc.



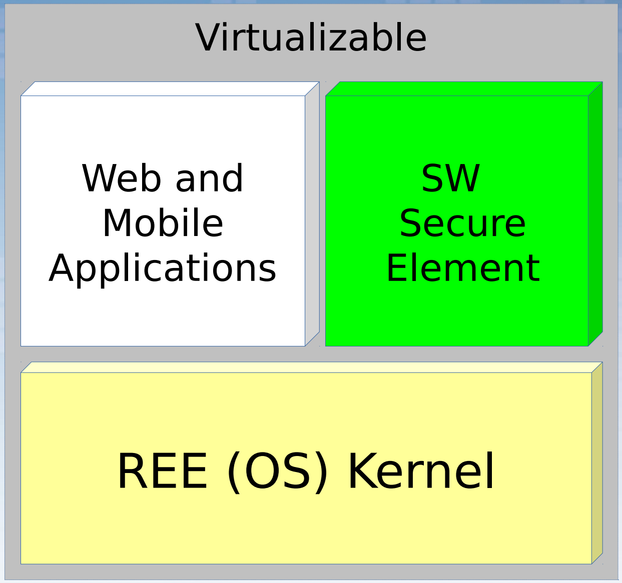
OWASP
Open Web Application
Security Project

Platform Security Components



Operating System Security

- Process Isolation (Virtual Address Spaces, Sand-boxing, Shared Memory, Limited IPC Traffic, Sockets)
- Discretionary Access Control (Object Access Control) – Identity based access restriction for objects. Access permission is transferable between objects.
- Mandatory Access Control (Policy Based Access Control)
- Memory Configuration (Non Executable Stack, Heap)



OS Security: Linux 1/3

Linux Security Extensions & Features

- Address Space Layout Randomization – A Loader feature which loads the parts of the executable in non-contiguous order.
- Support for extended file attributes. (e.g. SELinux, SMACK extended file attributes.)
- Mandatory Access Control systems
- Integrity Measurement - IMA / EVM

OS Security: Linux 2/3

Package and Kernel Module Verification

- Kernel Module White Lists – Limiting allowed modules in kernel.
- Disabling Dynamic Kernel Module Loading – Disabling a kernel module from loading if attacker slips its own binary.
- Signed Application Package Managers (RPM, DPKG ...) - Integrity and authenticity of the package is protected. Even confidentiality if sensitive information is included in the package.

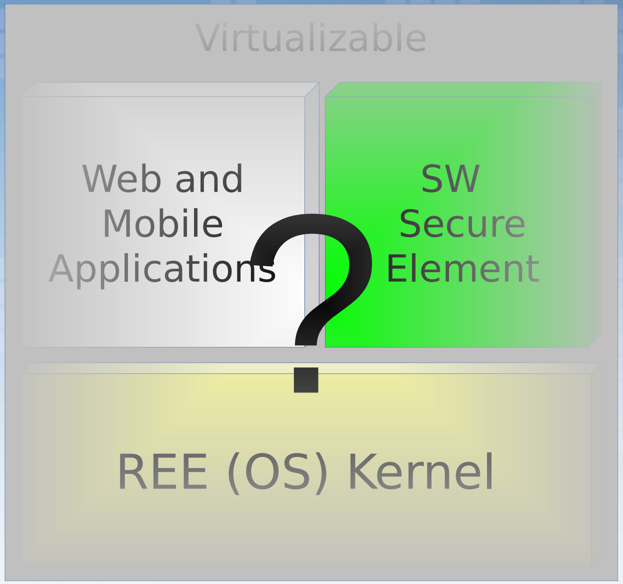
OS Security: Linux 3/3

Compiler Options

- Position Independent Execution – Libraries, heap stack are randomly located using ASLR but randomization of executable layout is done by PIE parameter.
- Stack Smashing Protection – Canary values to pin the return address of the function on the stack.
- Fortify Source – Calls for boundary checking functions instead of standard c functions without boundary check such as “strcpy”.

Food for Thoughts

- Is REE always safe with constant installation of new applications?
- Are attack surfaces on an OS easy and cheap to guard?
- Are both runtime and boot time security provided?
- Are sensitive operations (crypto, decoding, encoding, sign, verify) safe?
- Do all the parties in a mobile platform (i.e. OEM, Operator, OS vendor, Application Providers) provide their services in isolation?
- Is user's sensitive data (Bank id, fingerprint, Login information) safe ?

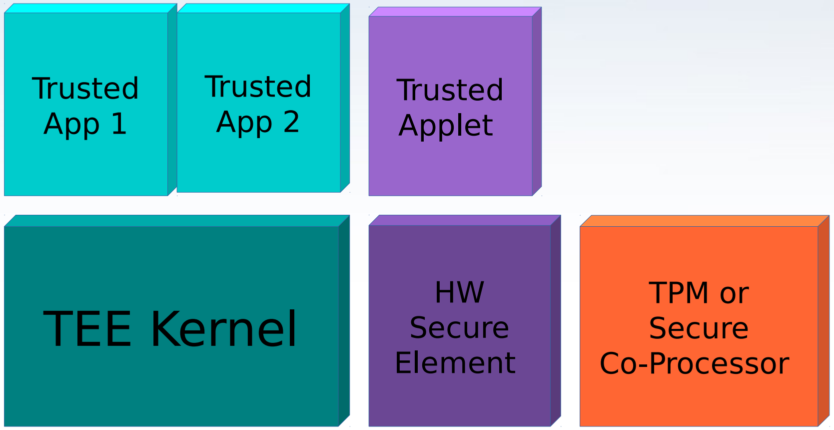


Motivation for Isolation of Sensitive Code Execution

- REE if full of attack surfaces from kernel to Applications.
- Security frameworks have high complexity and dependencies. The weakest link can risk the entire system
- Maintenance of a complex security system is difficult and multiple bugs can complement to a successful attack while they are harmless when alone.
- Security policies are complex to design. Circular permission chains can cause policy breaches
- Sensitive operations need to be maintained independent from the rest of the applications.

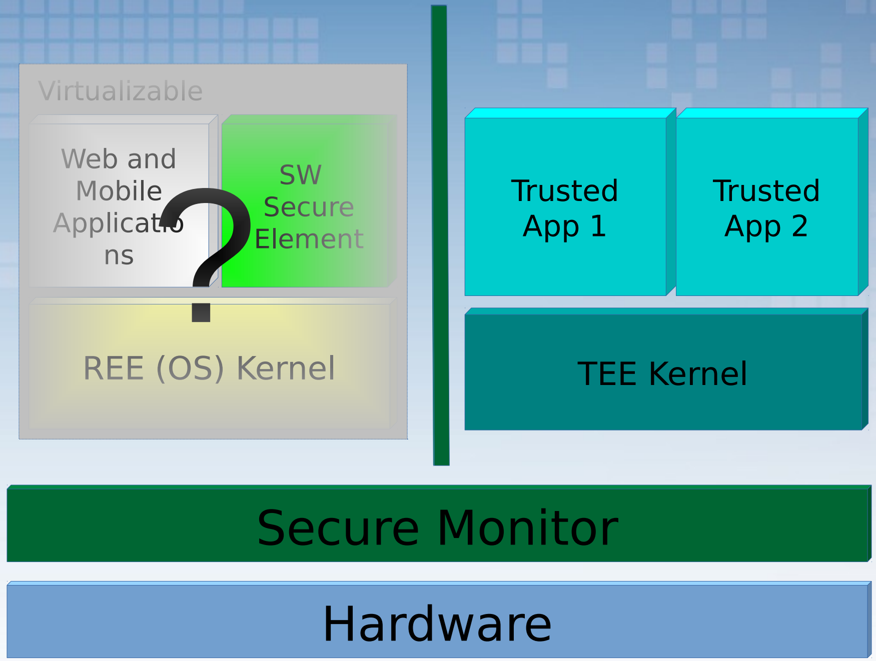
Hardware Components for Secure Code Execution

- Trusted Platform Module – Simple crypto capabilities with limited key storage capability. Used in laptops and desktops. Tamper resistant.
- Secure Element – Subscriber Identity Module. Useful for platform independent identification. Crypto operations with tamper resistance. Low power and performance.
- Secure Co-Processors- External Hardware Entity with relatively high performance. Crpyto operations, limited secure storage capabilities. (Crypto Accelerators, Big Number Operation Accelerators such as NXP ZigBee, NFC solutions and AMD PSP). Tamper resistant.
- Trusted Execution Environment
  Microkernel controlled - ARM® TrustZone®
  Special instruction based - Intel® SGX



Trusted Execution Environment

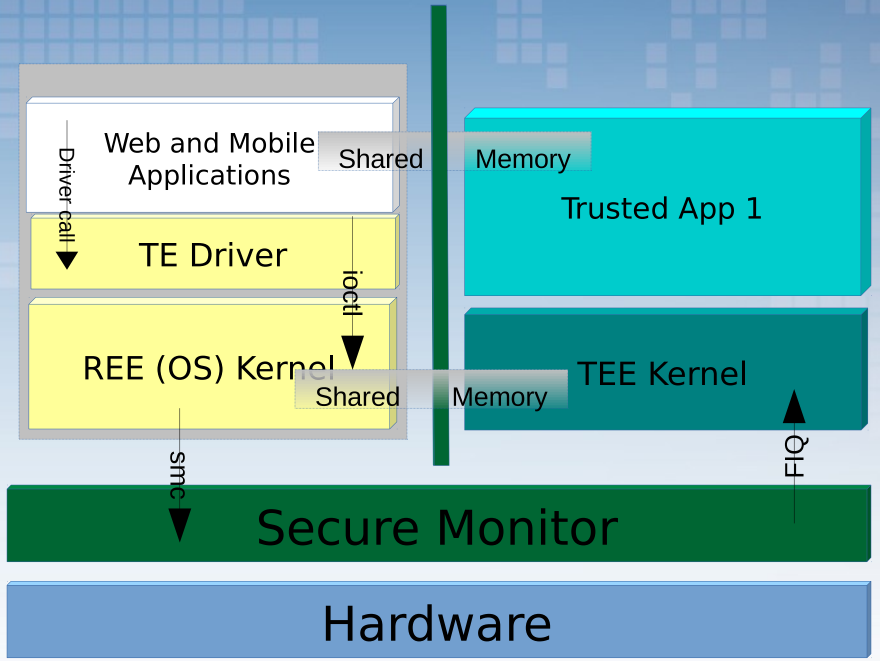
- Corresponds to a secure area of main processor of the system. Two virtual systems in one physical.
- SoC architectures highly benefit because of cost and power management ease. i.e. big.LITTLE, MP models.
- Capable of mastering system peripherals so that a system-wide isolation is facilitated
- Relatively high performance and support for 64 systems.
- Has to share one single core with the REE.
- Hack attacks and Shack attacks are protected in TEE. But timing attacks and side channel attacks (still possible to mitigate some) are not avoided.
- TEE Side can access REE side memory if mapping is possible.



ARM® TrustZone® Design

Trusted Execution Environment: How Does ARM® TrustZone® Work

- User space application establishes a session and world shared memory for data sharing
- Once data is in place, the application notifies the driver via device api.
- Driver translates the notification message and lets related kernel module to issue a Secure Monitor Call.
- Secure Monitor handles the interrupt, configures some critical registers and switches the context to counterparting trusted application.
- The trusted application copies the data into a non-shared memory block, processes and returns the response to the shared memory.
- Trusted application issues a secure interrupt to switch context to normal world.



ARM® TrustZone® Design

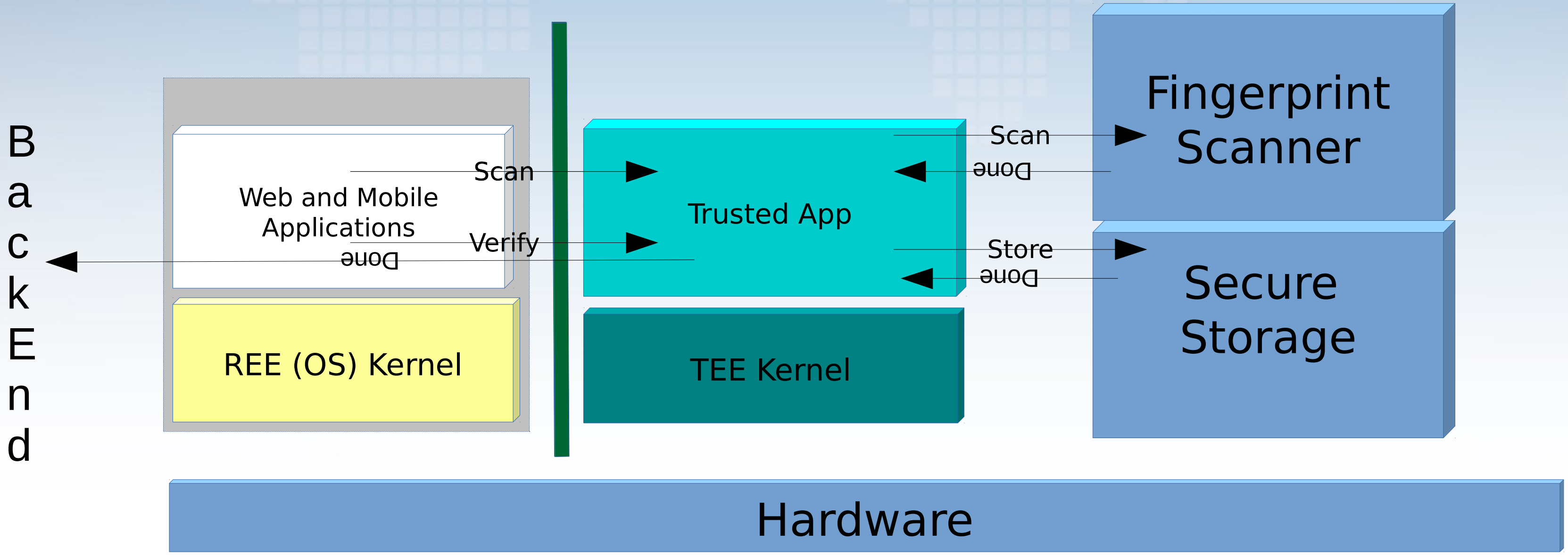
Exisiting TEE Services

- Key management
- Secure crypto operations
- Verification and signing operations
- Biometrics and Simple authentication features.(FIDO like authentication)
- REE runtime memory protection
- Secure Code Execution
- Secure boot chain
- Secure storage
- Trusted User Interface
- Digital Rights Management

Authentication Service

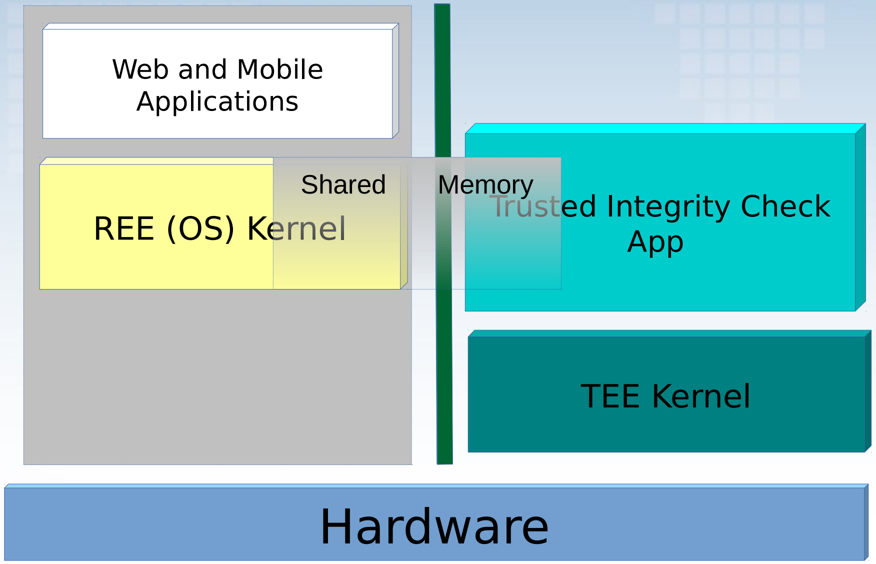
With Hardware Support

- FIDO like password free authentication is possible.
- Device authentication is possible.
- Fingerprint Authentication is possible: Scanning, Storing and Verification



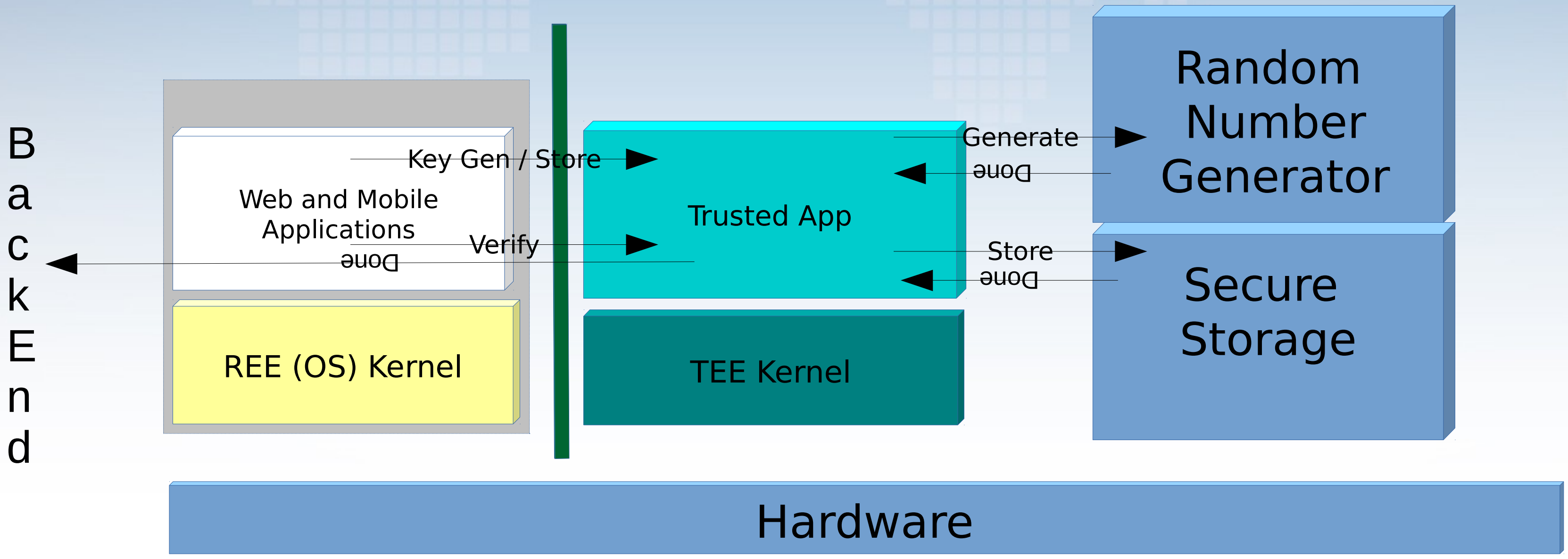
Real-time Kernel Protection

- Runtime integrity check for REE kernel performed by TEE application.
- Code block of the kernel has to remain the same as boot time.
- Checks are performed periodically (watchdog timer) or patagonix (PTE non-exec, when attempted exception triggers to check)



Key Management

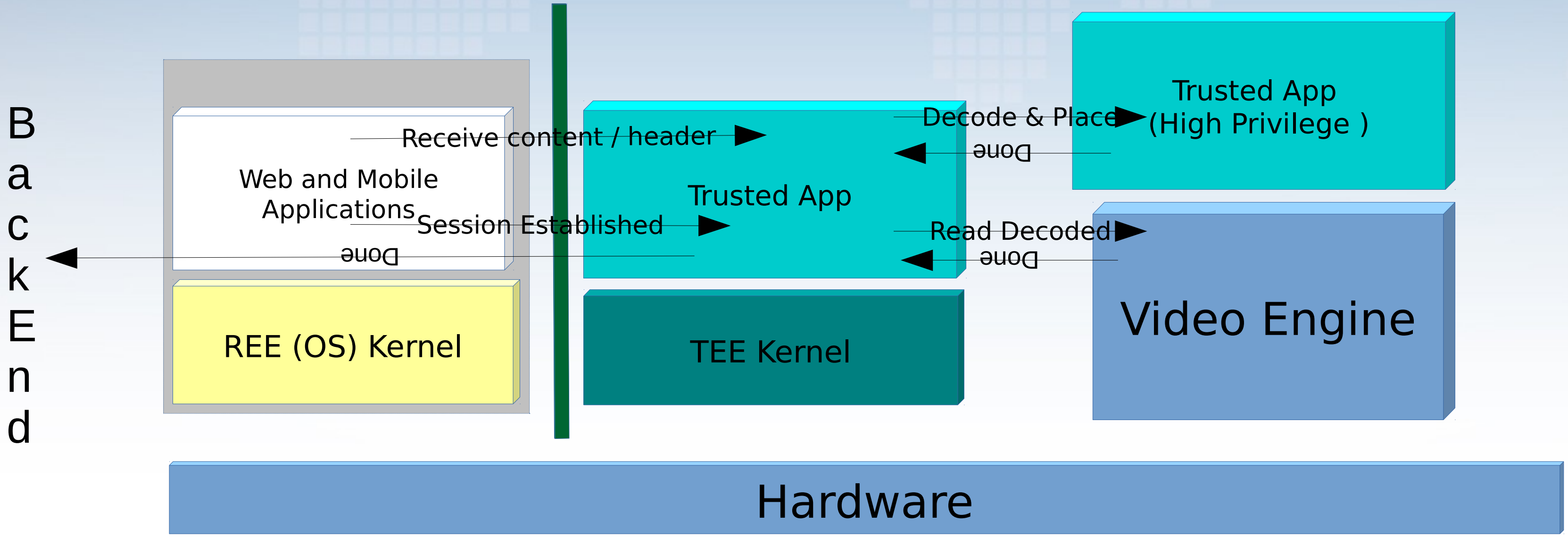
- Keys are stored and used respective to the applications owning them
- All of the keys are derived using RNG
- Keys are never exposed to the REE as well as cypto operations
- Android – Key Master compatible technology(Jbean)
- Heartbleed could be avoided



OWASP
Open Web Application
Security Project

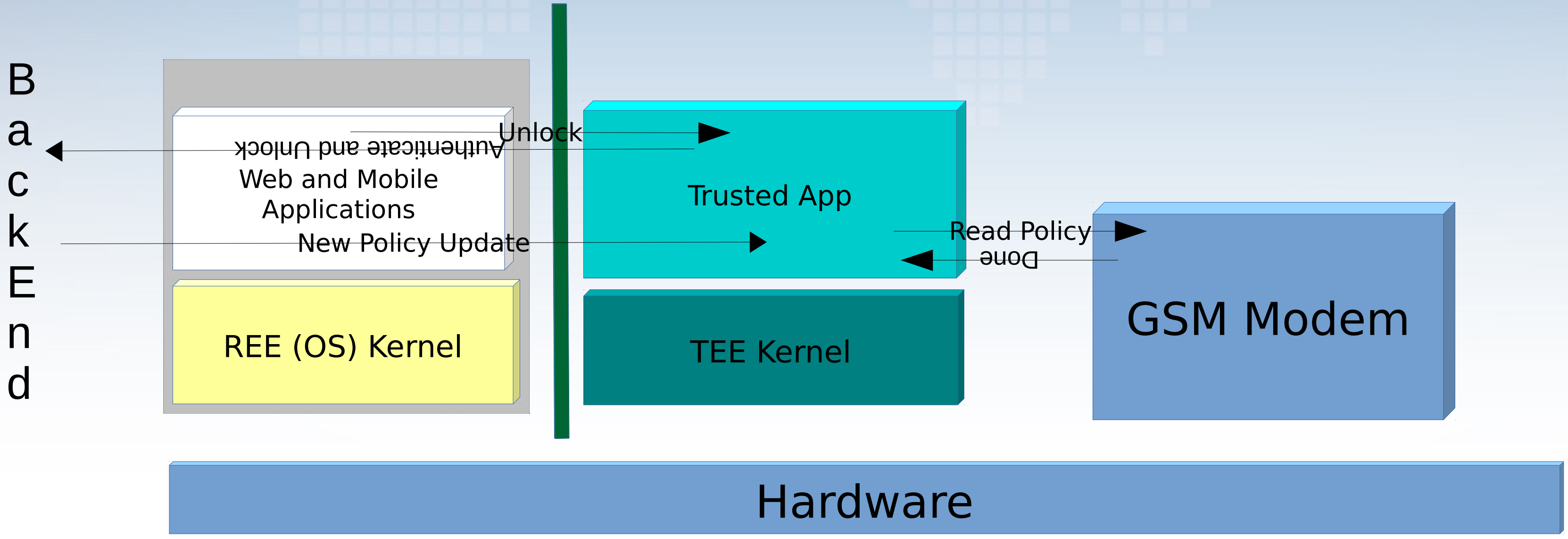
Digital Rights Management

- Content sharing and session key generation
- Encoding and decoding support
- Securing video engine and its bus.
- Storing and reusing session keys.
- Authentication of device



SIM Lock

- Sim lock operations are enabled via TEE
- Proxying new unlock policy
- Device Authentication is done by TEE
- Security is directly related to GSM modem software



Thanks for your time. Any questions ?

Onur Zengin
Lead Security Engineer @ Trustonic

onurzengin (at) gmail (dot) com
http://fi.linkedin.com/in/onurzengin/