arm

OP-TEE & the Trusted Services Project

Firmware components for platform security services

Julian Hall

February 2022

OP-TEE Overview

- A trustedfirmware.org project.
- A Trusted Execution Environment (TEE) running on Arm A-Profile cores using TrustZone-based HW isolation.
- Designed as companion to a Rich Execution Environment (REE) Linux kernel.
- Implementing standard (GlobalPlatform) APIs
  - TEE Internal Core API v1.1.x – towards trusted applications
  - TEE Client API v1.0 – towards userland

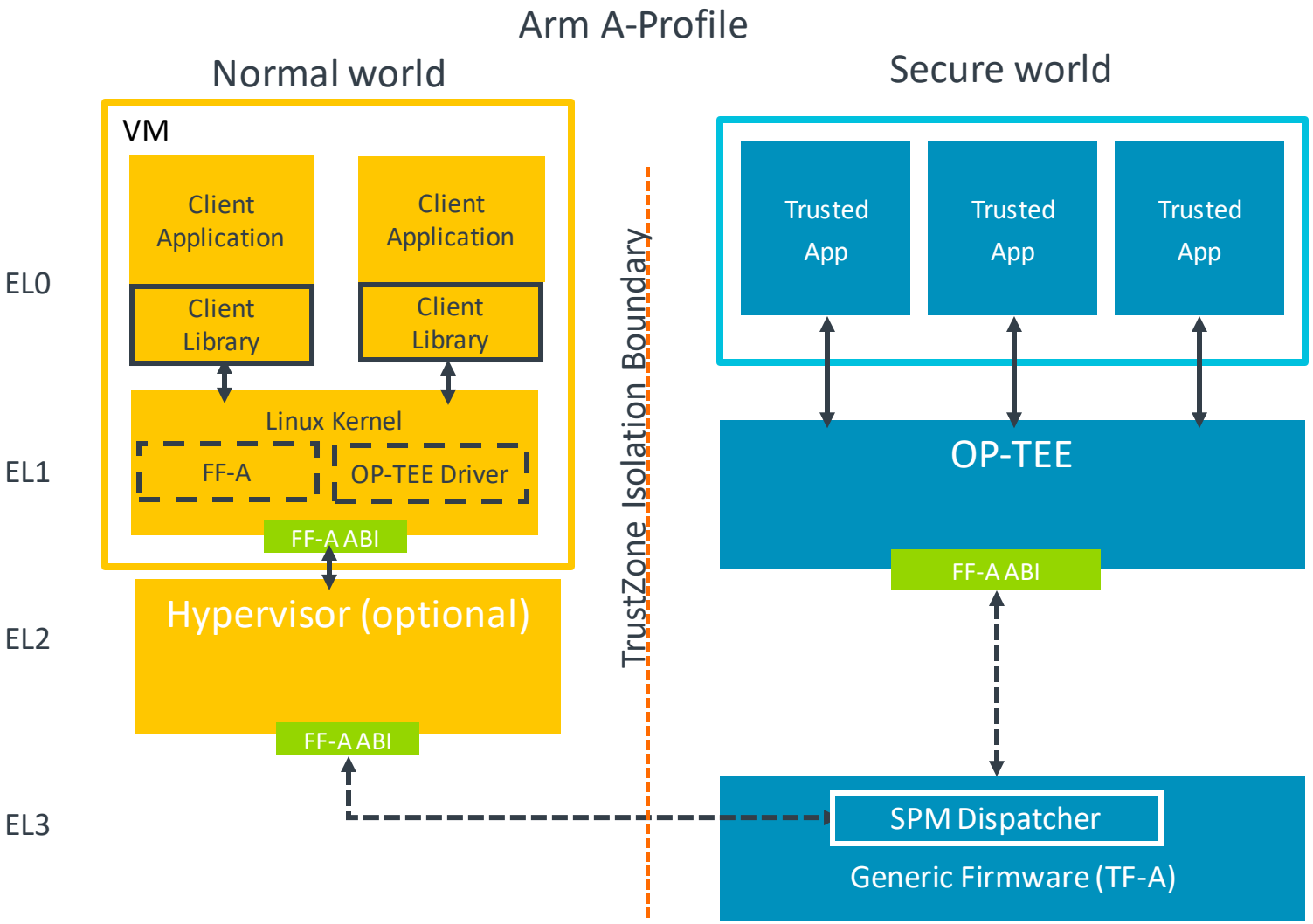
OP-TEE Goals

- Isolation
  - the TEE provides isolation from the non-secure OS and protects the loaded Trusted Applications (TAs) from each other using underlying hardware support
- Small footprint
  - the TEE should remain small enough to reside in a reasonable amount of on-chip memory as found on Arm based systems
- Portability
  - the TEE aims at being easily pluggable to different architectures and available HW and has to support various setups such as multiple client OSes or multiple TEEs.

OP-TEE Components

- A secure privileged layer (optee_os), executing at secure EL-1.
- (Standard) user space client library implementing the GlobalPlatform TEE Client API and supplicant daemon (optee_client)
- A Linux kernel TEE framework and driver (mainlined since v4.12).
- A Trusted Application “devkit” to build and sign the Trusted Applications.
- An extensible test suite (optee_test) for doing regression testing and testing the consistency of the API implementations.
- Build harness for a variety of platforms (including QEMU)

OP-TEE deployment (no S-EL2 virtualisation)



OP-TEE and Trusted Services

- OP-TEE is used by the Trusted Services project to provide isolated processing environments that conform to the Arm FF-A specification.
- FF-A compatibility changes have been contributed to the OP-TEE project.
- Provides the option for deploying Secure Partition images on pre-Armv8.4 silicon.
- New functionality coexists with existing Trusted Applications and GP compliant services.
- The Trusted Services project complements OP-TEE by providing reusable components for building common platform services.

Trusted Services Overview

- A trustedfirmware.org project.
- A home for firmware components for building security related services.
- Supports service deployment in different secure processing environments.
  - FF-A enabled OP-TEE used as reference.
- Creates opportunities for:
  - Adopting a common framework with standard conventions and solutions.
  - Component and test-case reuse.
  - Publishing standard public interfaces.
  - Sharing security enhancements.
  - Having a common solution for build, testing and deployment.
- Enables downstream projects to integrate, configure and build trusted services into firmware to suite product or distro needs.

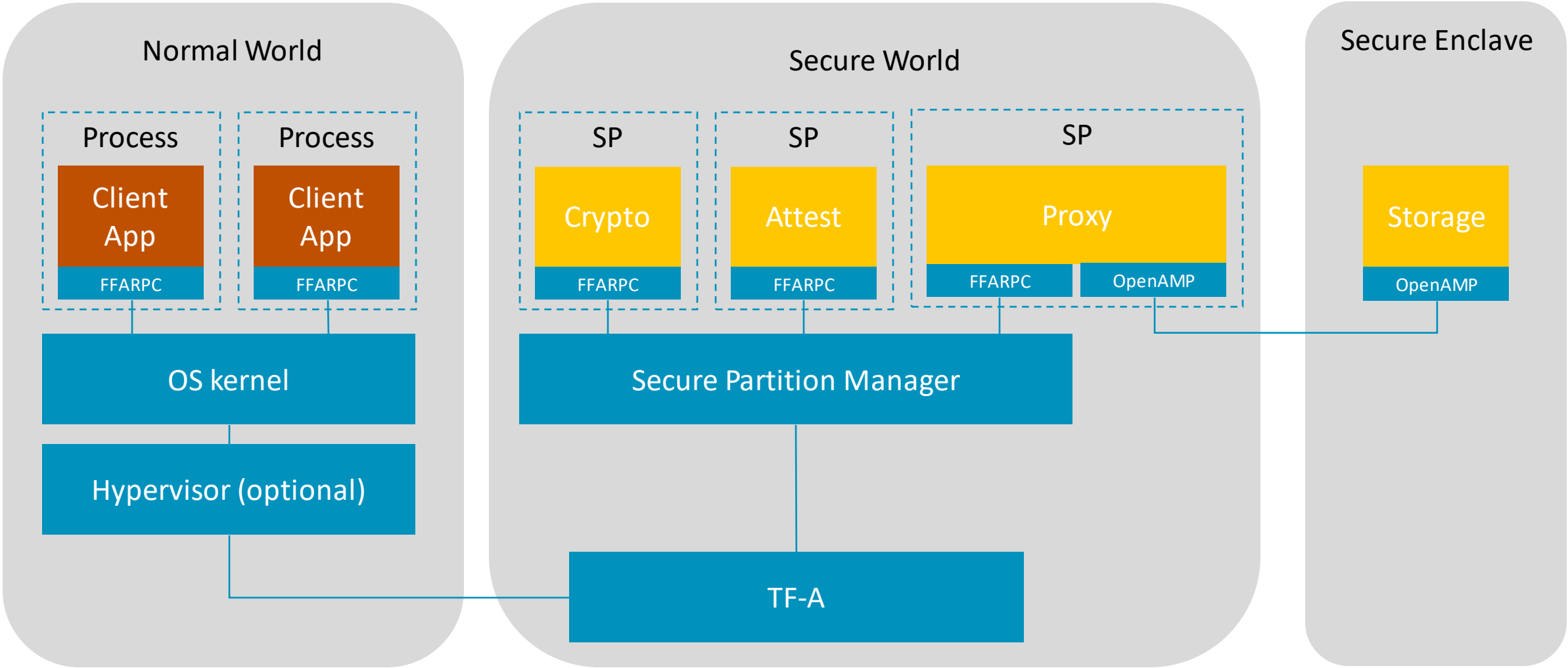
Classes of Service

- General purpose platform security services:
  - Crypto (including TRNG)
  - Secure storage
  - Firmware attestation – reporting on the security state of platform firmware
- Domain specific services that use platform services as backends:
  - UEFI variables
  - Firmware update
- Test and development support:
  - Logging
  - Test runner – test execution from within secure processing environments

Classes of Client

- Bootloader e.g. u-boot UEFI variable access
- Kernel driver
- User-space applications
- System daemons e.g. PARSEC
- Other trusted services

Example Service Deployment



arm

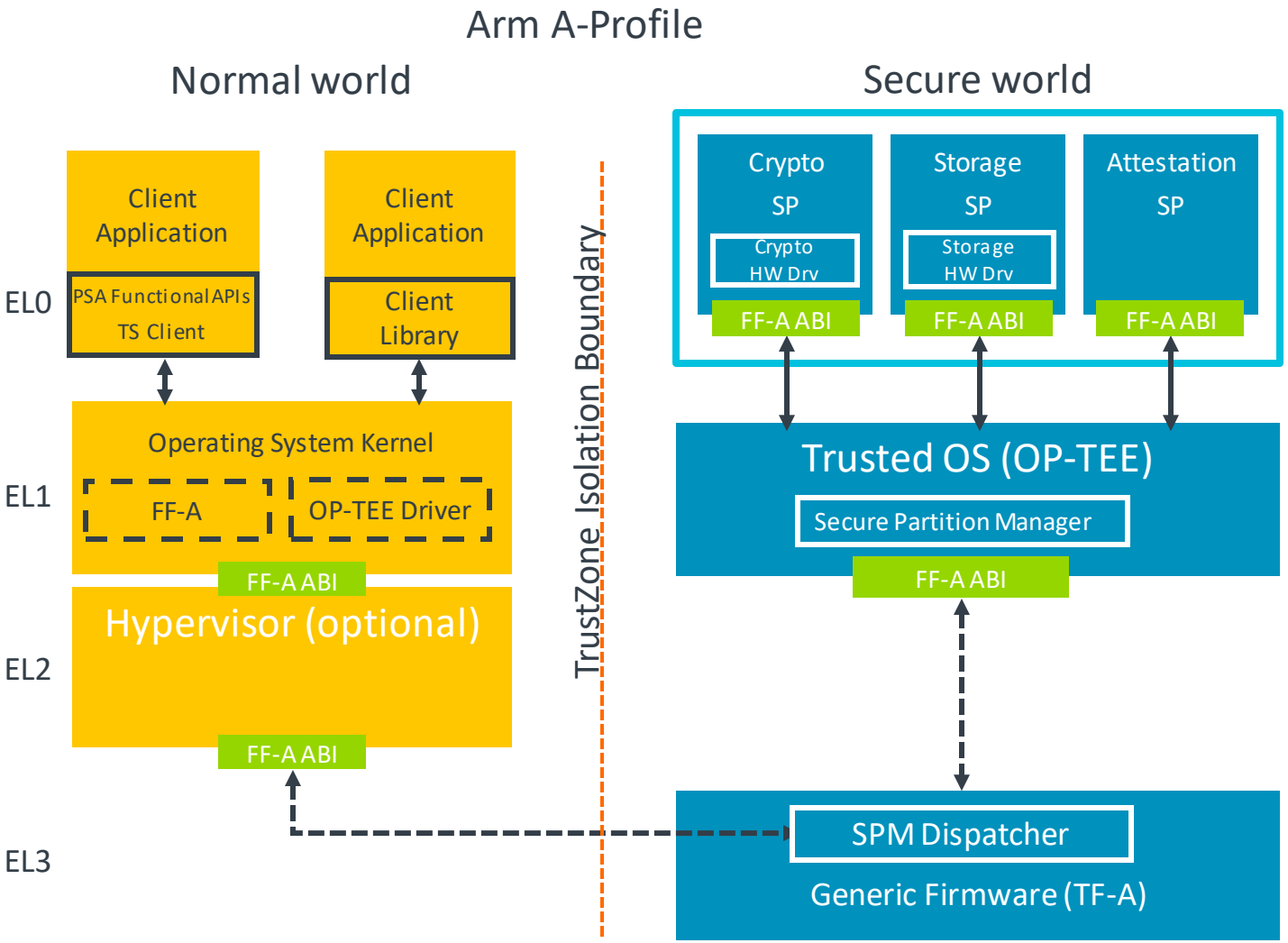
Secure Processing Environments

- Numerous options for realizing secure processing environments:
  - Secure partitions
  - Secure enclave/MCU
  - Dedicated VM
  - Trusted OS (e.g. OP-TEE, Trusty)
- Availability constrained by:
  - Hardware capabilities
  - Silicon vendor BSP capabilities
  - Arm architecture version
  - Segment specific conventions e.g. use of a particular TEE/Trusted OS
- Landscape for trusted service deployment is somewhat fragmented

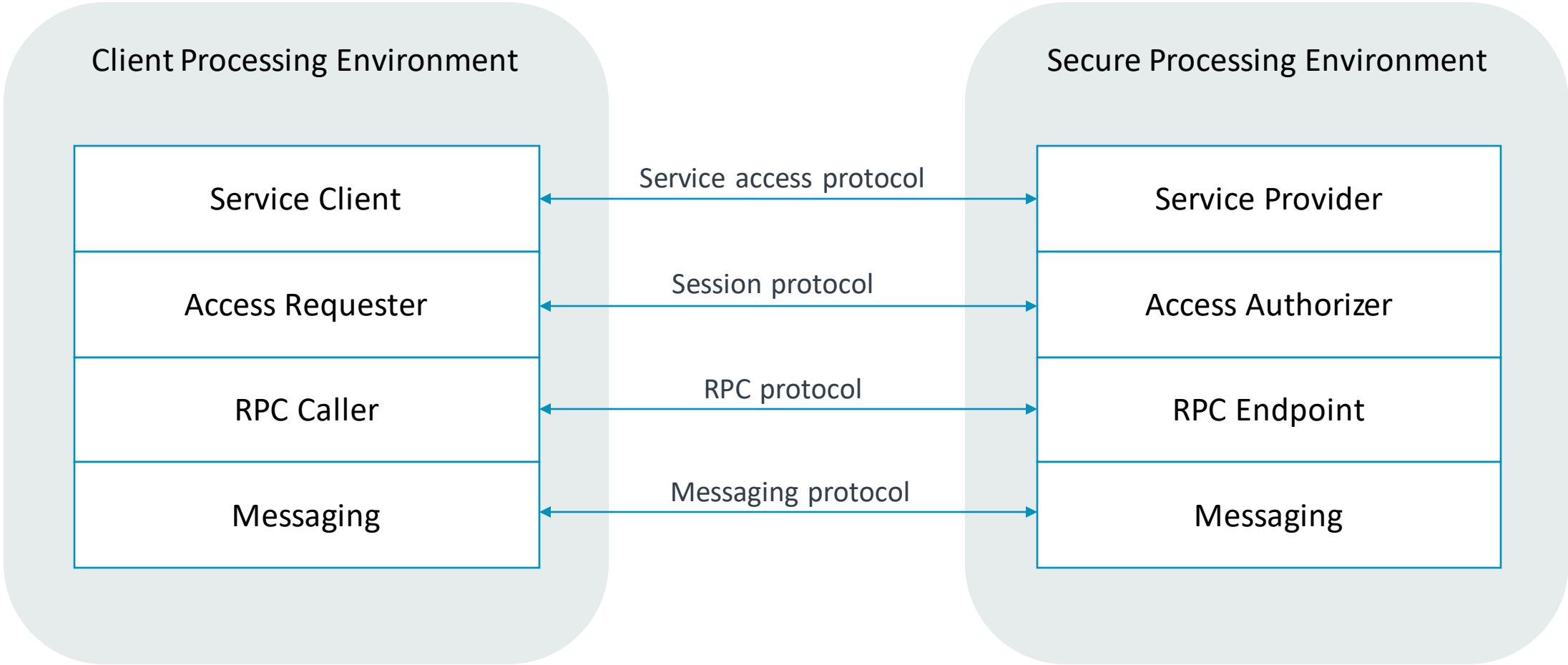
FF-A Enablement in OP-TEE for pre-Armv8.4 Devices

- Armv8.4 architecture introduces virtualization in the Secure state.
- Architectural features enable isolation of mutually distrusting software components running in the secure world.
- An isolated secure world processing environment is referred to as a *secure partition* (SP).
- FF-A (Firmware Framework for Arm A-profile) standardizes interfaces for communication between SPs.
- FF-A reference model defines the role of Secure Partition Manager (SPM).
- For pre-Armv8.4 devices, OP-TEE has been extended to perform role of SPM.
- Enables SP images to run at S-EL0.
- Helps to harmonize software stack between pre and post Armv8.4 devices.
- FF-A enabled OP-TEE used as reference platform for Trusted Services.

OP-TEE based reference deployment



Common Layered Model



Deployments

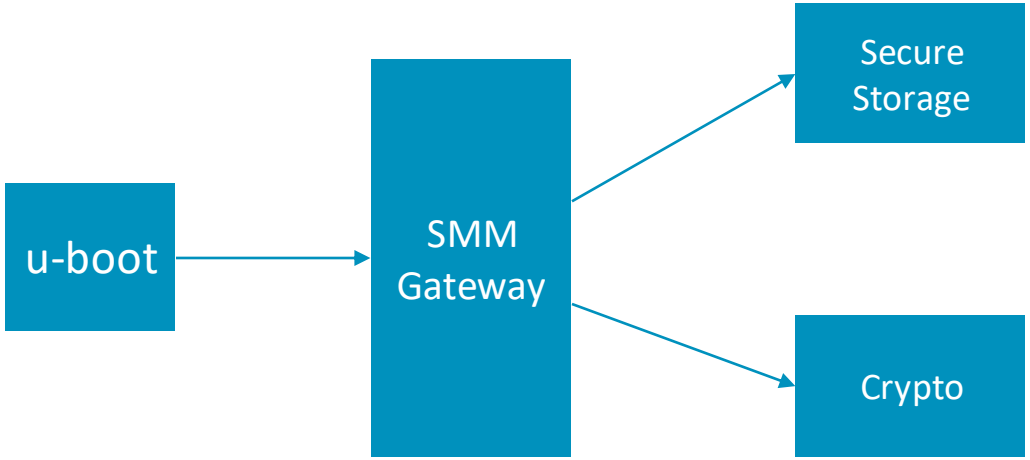
- The TS project adopts a structure that separates:
  - *Components* - reusable units of software (TS or external components from upstream projects)
  - *Environments* - execution environments e.g. an FFA compliant secure partition, a GP TA
  - *Platforms* - platform specific components such as hardware drivers
- The *opteesp* environment corresponds to FF-A enabled OP-TEE.
- A TS deployment combines a set of components and an environment to be built and installed on a particular platform.
- The TS project maintains many deployments such as:
  - **crypto/opteesp** - The crypto service deployed in an SP running under OP-TEE
  - **protected-storage/opteesp** - Secure storage service deployed in an SP under OP-TEE
  - **smm-gateway/opteesp** - UEFI smm services that use backed platform services
  - **ts-service-test/linux-pc** - service level test that test services running within a user-space process.
  - **ts-service-test/arm-linux** - service level tests that run on Arm target from Linux user-space and test services running in real environment.

Downstream Consumers

- The TS project can act as an upstream source for:
  - TS maintained deployments such as secure partitions that host service providers.
  - Service provider components, integrated and deployed by downstream projects.
  - Client API adapters e.g. present PSA functional APIs to client applications.
  - Service access protocol definitions.
  - Service level test suites - includes PSA API compliance test integration.
- For Yocto based distros, TS configuration and build recipes are being added to *meta-arm*.

UEFI Services - SMM Gateway

- UEFI System Management Mode (SMM) services available via SMM Gateway.
  - Intended for non-EDK2 normal world firmware
  - Alternative to StMM.
  - ABI compatible with FFA based StMM interface.
  - Currently only SMM Variable service support.
- Adapts the popular API Gateway pattern used in microservices where an API gateway provides a single entry point for clients. Service requests are delegated to service backends.
- Exploits backend platform services e.g. secure storage, crypto.
- Flexibility to use any backend service deployment.
- SMM Gateway role is limited to presenting SMM service APIs and adapting to backend service protocols.
  - Small footprint



Summary

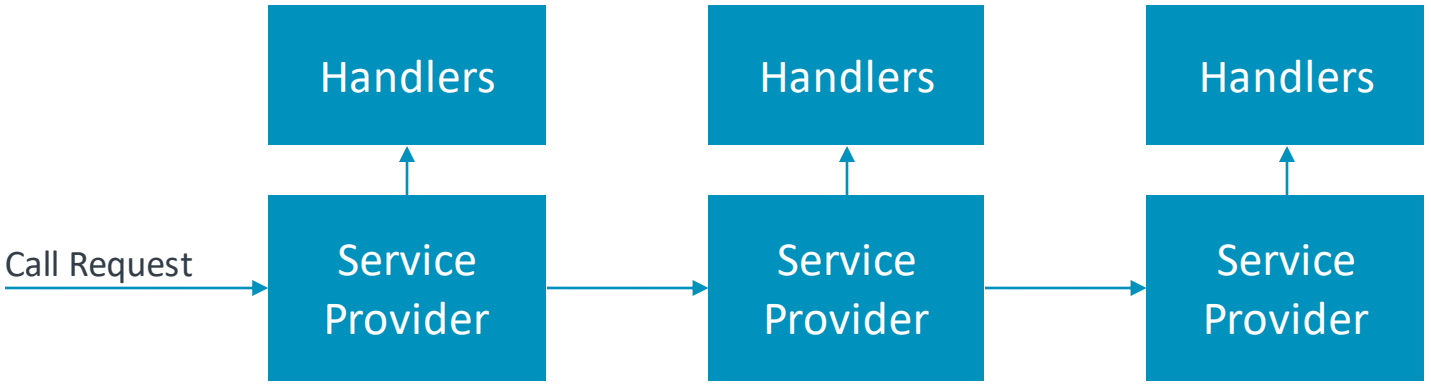
- The TS project is structured to promote reuse.
- Service provider components are decoupled from any particular secure processing environment.
- Project currently uses FF-A enabled OP-TEE as the reference platform for integration and testing.
- Downstream projects may pick-and-mix components and deployments to suite project/distro goals.
- Despite diversity in platform architectures, the TS project can help deliver a uniform set of services.

arm

Thank You
Danke
Gracias
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה

Common Service Provider Architecture

- All service providers conform to a common architecture
  - Base service provider delegates service requests to handlers
  - Service providers may be chained for modular extension of capabilities
  - Common framework for access control
  - Independent of underlying RPC layer
- Service providers allow for alternative backends.
- Alternative parameter serializations supported.



Service Identification and Discovery

- Deployment independent service identifiers may be used to decouple client applications from platform specifics.
- Compatible with Global Platform TPS Client service identifiers.
  - `TFORG-Crypto-0/1.0.1/TFORG-TSSP/TFORG-TS-crypto-opteesp-2.3`
- Any service provider may optionally support discovery operations for discovering:
  - Basic information such as maximum parameter size, supported serializations and service status.
  - Service capabilities - supported operations.
  - Service location for service enumeration.
- Alternatively, well-known identifiers may be used e.g. Service GUID for UEFI SMM service identification.

Security Model

- Trusted services are assumed to be deployed within an isolated execution environment with hardware backed protection of memory and secure peripherals.
- System integrators may use kernel-level access control at device-node level.
- Additionally, access control may be applied by a service provider to control what a client can do.
- The access control model separates the following concerns:
  - Client authorization - authenticating a client, providing a trusted identity and authorizing a level of privilege.
  - Access authorization - checking that a call request is permitted for the originating client.
- Access control features are currently under development.