arm

Trusted Firmware-A

Secure Partitions

Sandrine Bailleux

Senior Software Engineer

Open Source Software

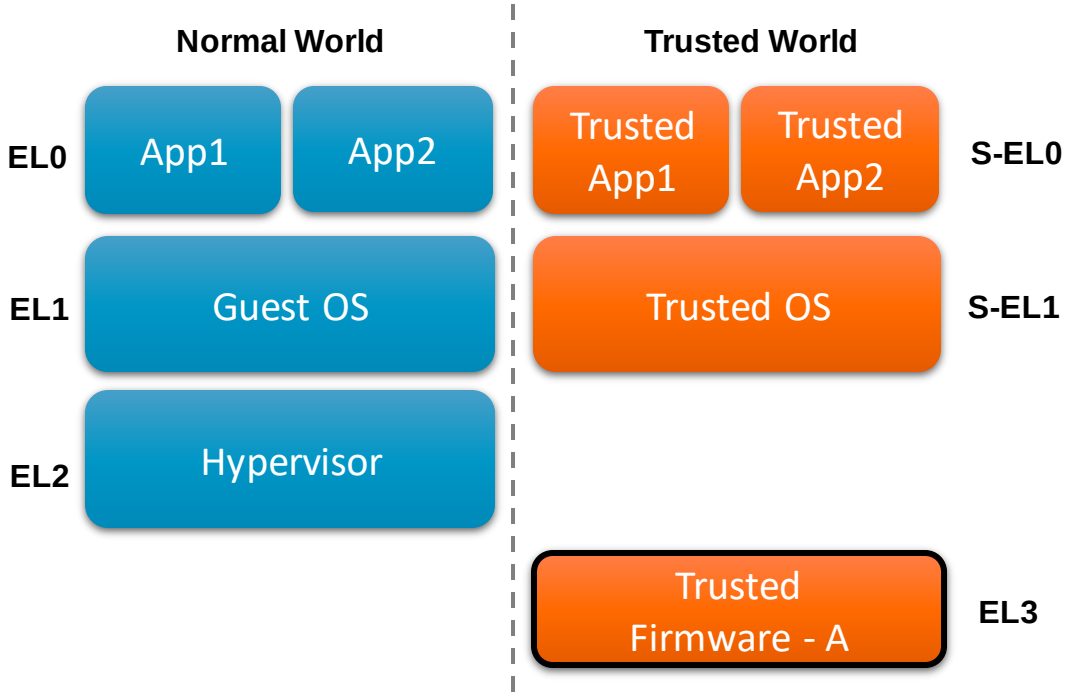
- OSFC
- September 2018

Overview of Trusted Firmware-A

arm

What is Trusted Firmware-A?

- Reference implementation of Trusted World software (EL3) for A-class devices
- Foundation to build a Trusted Execution Environment
- Designed for code reuse and ease of portability to new platforms
- Open source project on Github
- BSD-3-Clause license
- About 30 platform ports available upstream
- Well-established project (created in 2013)



arm

Main features of TF-A

- Secure device initialization
- Modular boot flow
- Trusted boot
- Secure monitor that handles world switching
- Integration with Trusted OS through a Secure-EL1 Dispatcher
- Recovery mode through firmware update
- Provides run-time services to lower exception levels
- Handles power management

TrustedFirmware.org

TF-A will soon become a Linaro community project

- Open-governance open source software project (TF-A + TF-M)
- Operated independently from the main Linaro organization
- Project membership equally open to Linaro members and non-members

Benefits for members:

- Take partial ownership of a project you depend on
- Ensure your dependencies are maintained and continually validated (CI, Board Farm)
- Reduce internal maintenance costs by pushing generic features you need
- Help ensure that the open source community supports Trusted Firmware interfaces and features

Contact board@trustedfirmware.org for more information

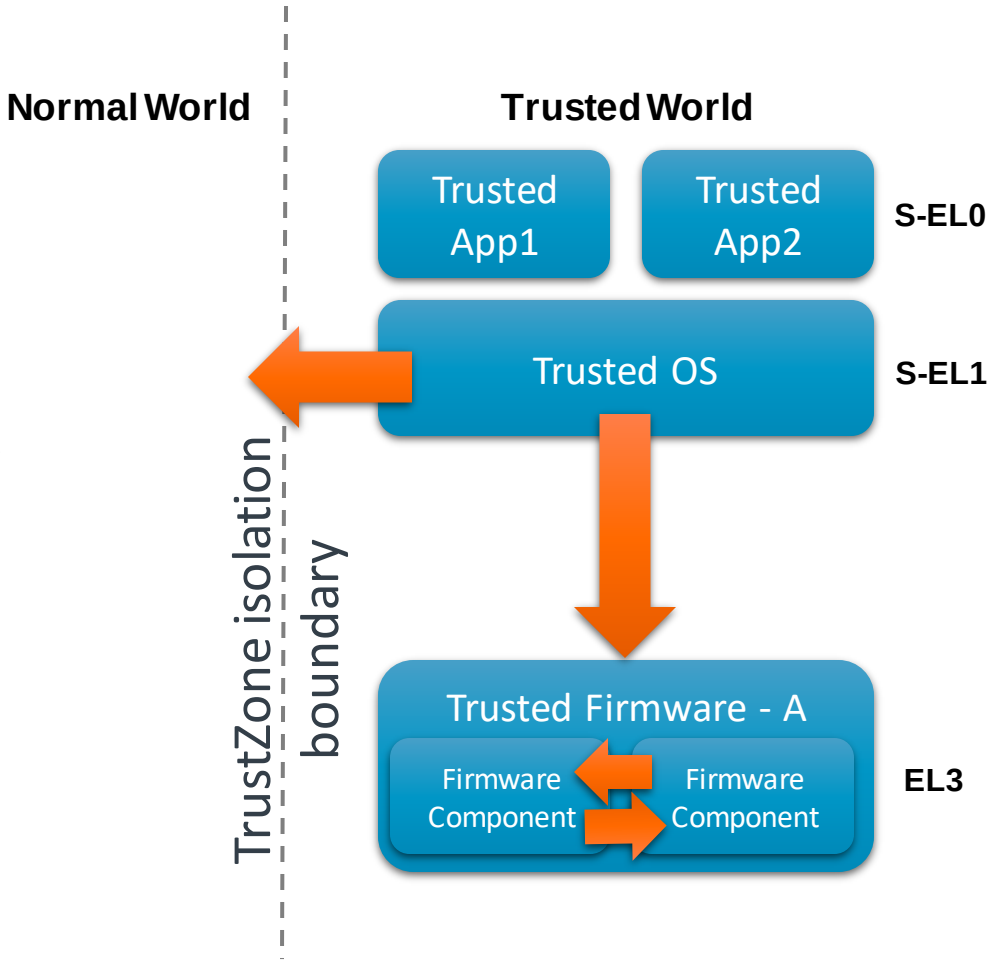
arm

Today's challenges in Trusted World

arm

No proper isolation between Trusted World agents

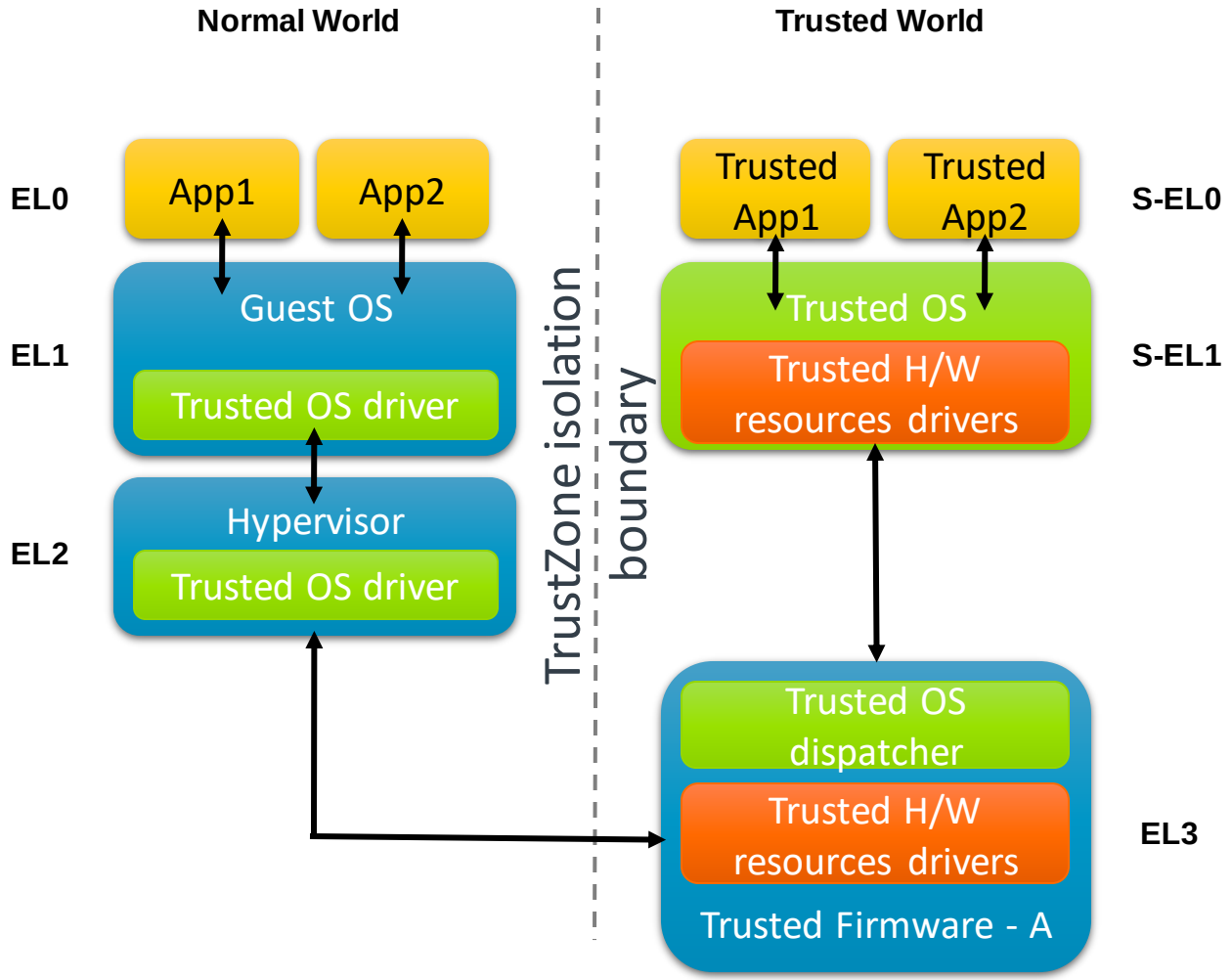
- Arm TrustZone™ only provides isolation between Trusted World and Normal World
- EL3 and S-EL1 have same level of access to:
  - System address map
  - Physical interrupts
- Firmware in EL3 cannot be isolated from Trusted OS
- EL3 firmware components cannot be isolated from each other
- Normal world cannot be protected from Trusted OS in S-EL1



arm

Fragmentation of Trusted World agents

- Limited portability of applications across Trusted OS
- Code is coming from different vendors
- Cooperation and mutual trust is needed

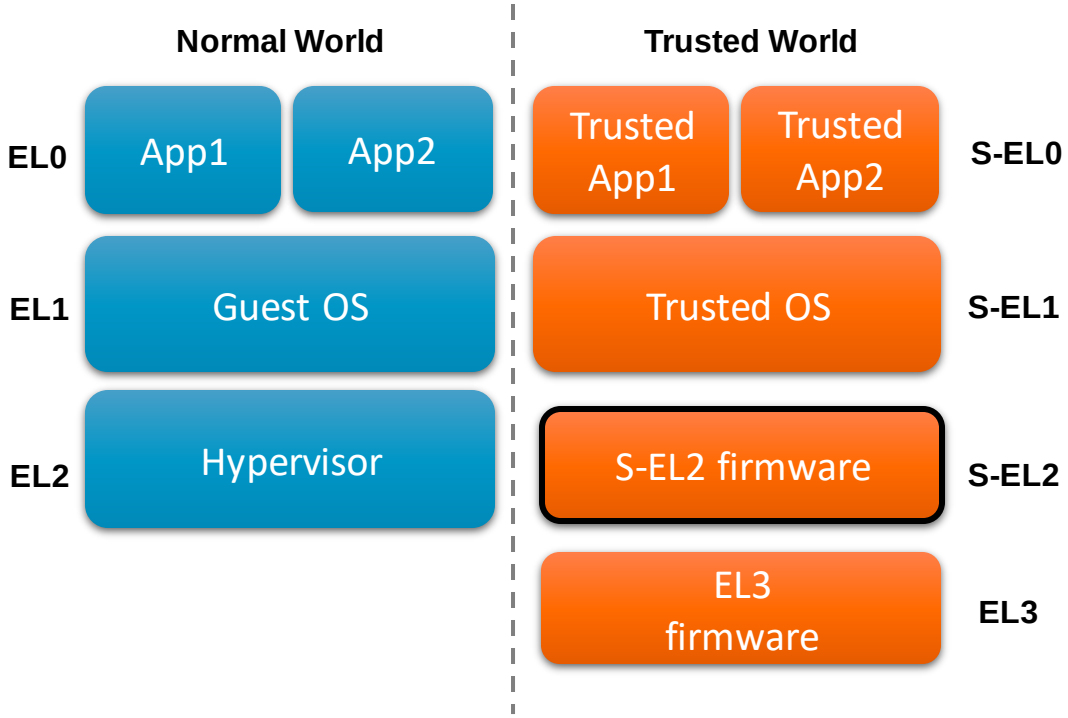


Generic software
Trusted OS specific
Application specific
Silicon vendor specific

Solution: Virtualization in the Trusted World

- Armv8.4 will introduce secure virtualization extensions (new S-EL2)
- Restrain access to physical memory and physical interrupts
- S-EL2 controls stage-2 address translations
- Physical interrupts are routed to S-EL2

- Need a software architecture to leverage this...



arm

Secure partitions

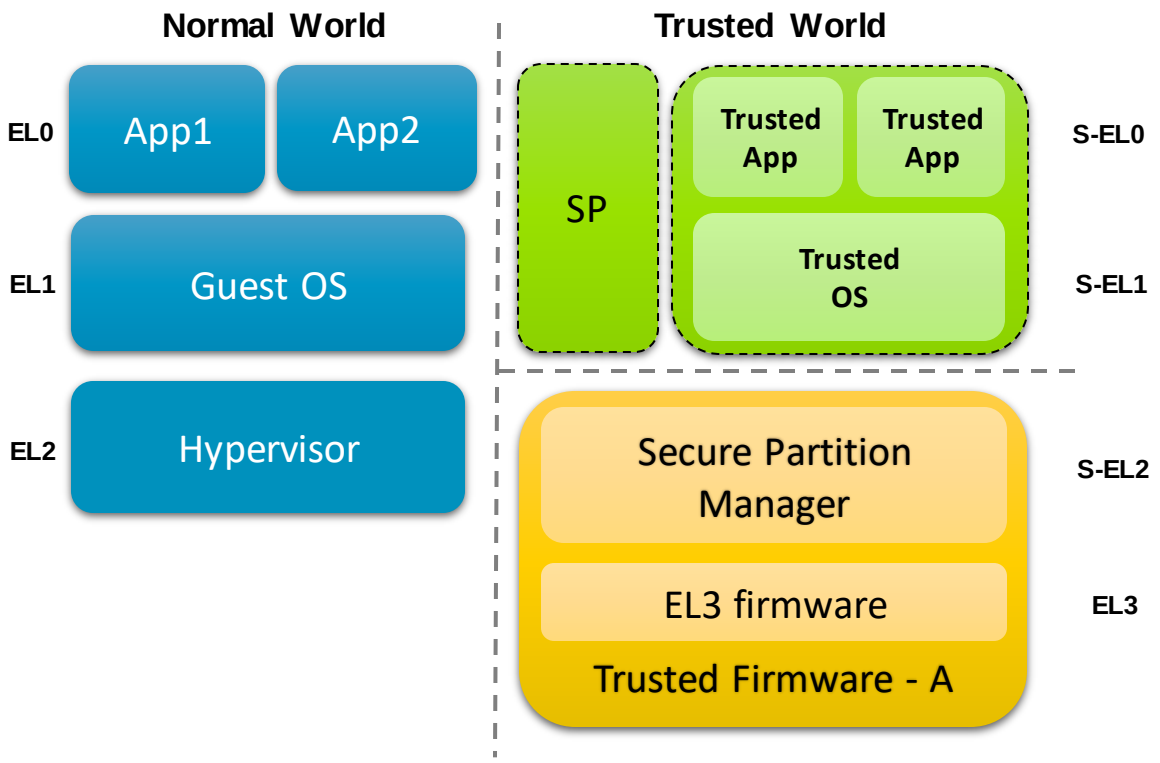
arm

Introducing secure partitions

- Low privileged software sandbox in Trusted World
- Runs under control of higher privileged software
- Provides secure services
- Use cases:
  - Platform-specific services (e.g. RAS error handling)
  - Device-specific services (e.g. cryptographic hardware)
  - Higher-level services
    - Secure storage
    - Secure payment
    - Key management
    - Secure firmware update
- OS agnostic by standardizing communication interfaces

Secure partition manager (SPM)

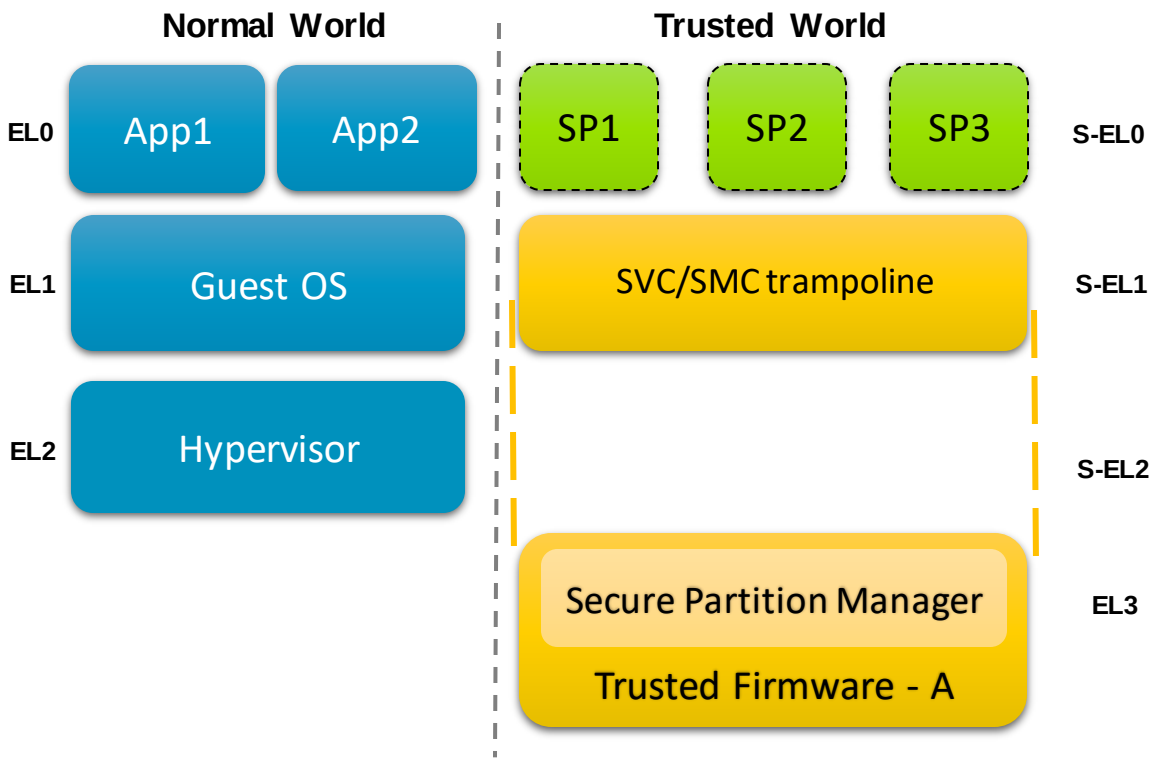
- Highly-privileged software entity managing all secure partitions
- Conceptually similar to a hypervisor managing Guest OS



arm

Secure partitions before Armv8.4

- SPM currently implemented in TF-A at EL3
- Shim layer to relay SVC from S-EL0 into SMC to EL3
- Enable early development of SPM code before S-EL2 availability



arm

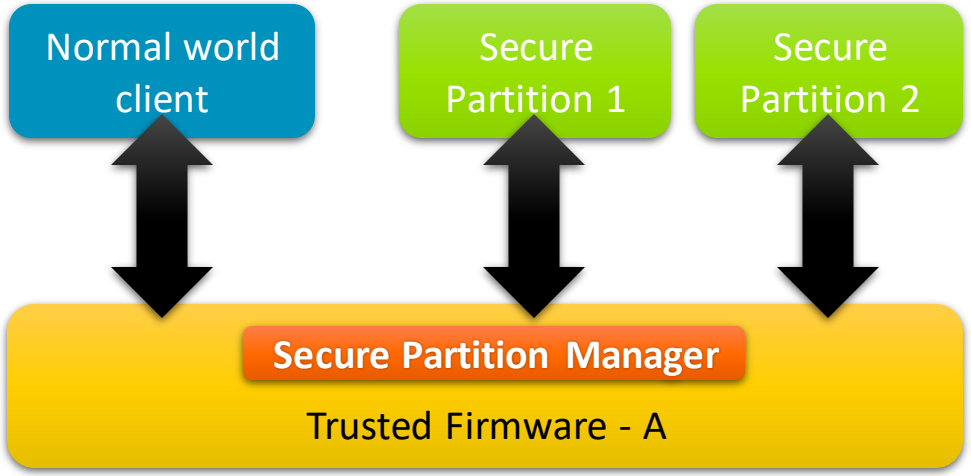
Secure partitions

Deep dive

arm

Run-time model of Secure Partitions

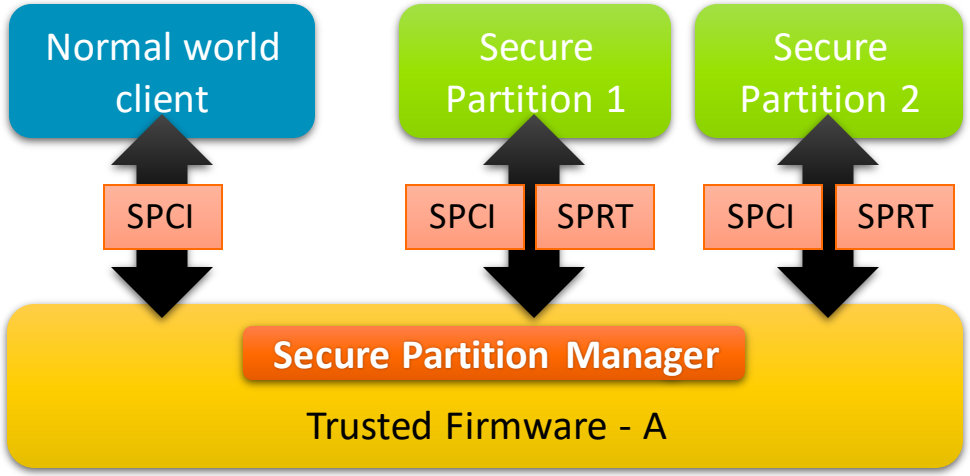
- Channel-based client/server model
- Server = a secure service in a secure partition
- Client = normal world client, e.g.:
  - Bootloader
  - Guest OS, native OS or hypervisor
- Client = another secure partition that depends on it to fulfill its own services
- Secure partitions are structured as a loop that:
  1. waits for input
  2. processes the input
- SPM acts as a proxy in between clients and secure partitions
- Normal world software remains in charge of scheduling



arm

Standardized communication interfaces

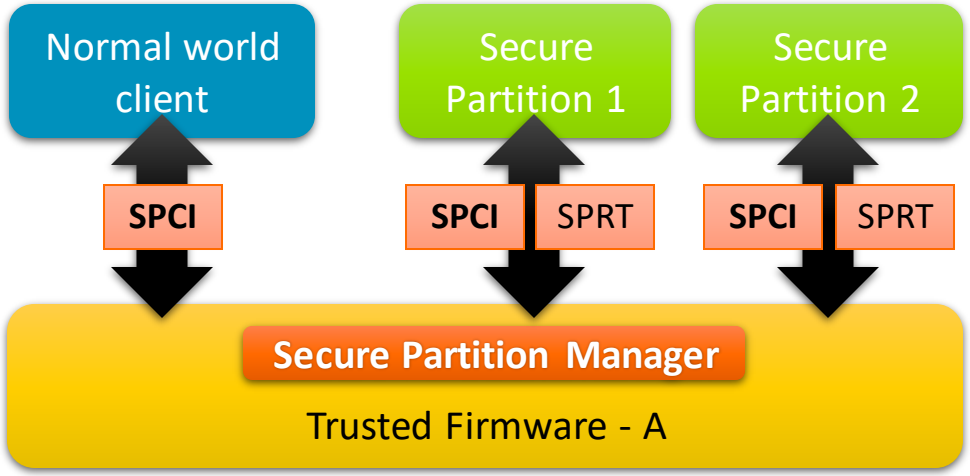
- Clients request services using a standardized communication interface
  - ➢ Secure Partition Client Interface (SPCI)
- Secure services handle requests using a standardized run-time interface
  - ➢ Secure Partition Run-Time (SPRT)
- Arm architectural specifications under development
- Trusted Firmware-A provides a reference implementation



arm

Secure Partition Client Interface (SPCI)

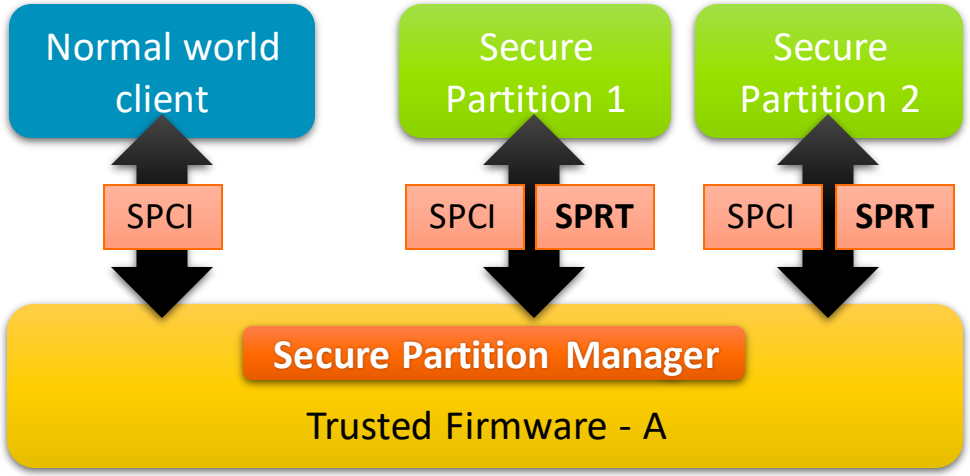
- Collection of ABIs provided to clients and implemented by SPM
- Accessible through SMC (or SVC) conduit
- Allows clients to:
  - Discover secure services
    - Each service has a unique ID
  - Establish a communication channel with a secure service
  - Exchange messages with secure services
  - Share memory with secure services



arm

Secure Partition Run-Time (SPRT)

- Collection of ABIs provided to secure partitions and implemented by SPM
- Accessible through SMC (or SVC) conduit
- Allows secure partitions to:
  - Retrieve client requests
  - Send responses back to client
  - Receive interrupts targeted to it
- Provides a common programming model for all secure partitions
  - Ease the development of secure services
  - Improve robustness and security



arm

References

- Arm Trusted Firmware-A source code on Github

https://github.com/ARM-software/arm-trusted-firmware/

- Arm Trusted Firmware as a Linaro community project

https://www.trustedfirmware.org

- Secure Partition Manager design document

https://github.com/ARM-software/arm-trusted-firmware/blob/master/docs/secure-partition-manager-design.rst

- Arm Secure-EL2 extension

https://community.arm.com/processors/b/blog/posts/architecting-more-secure-world-with-isolation-and-virtualization

arm

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