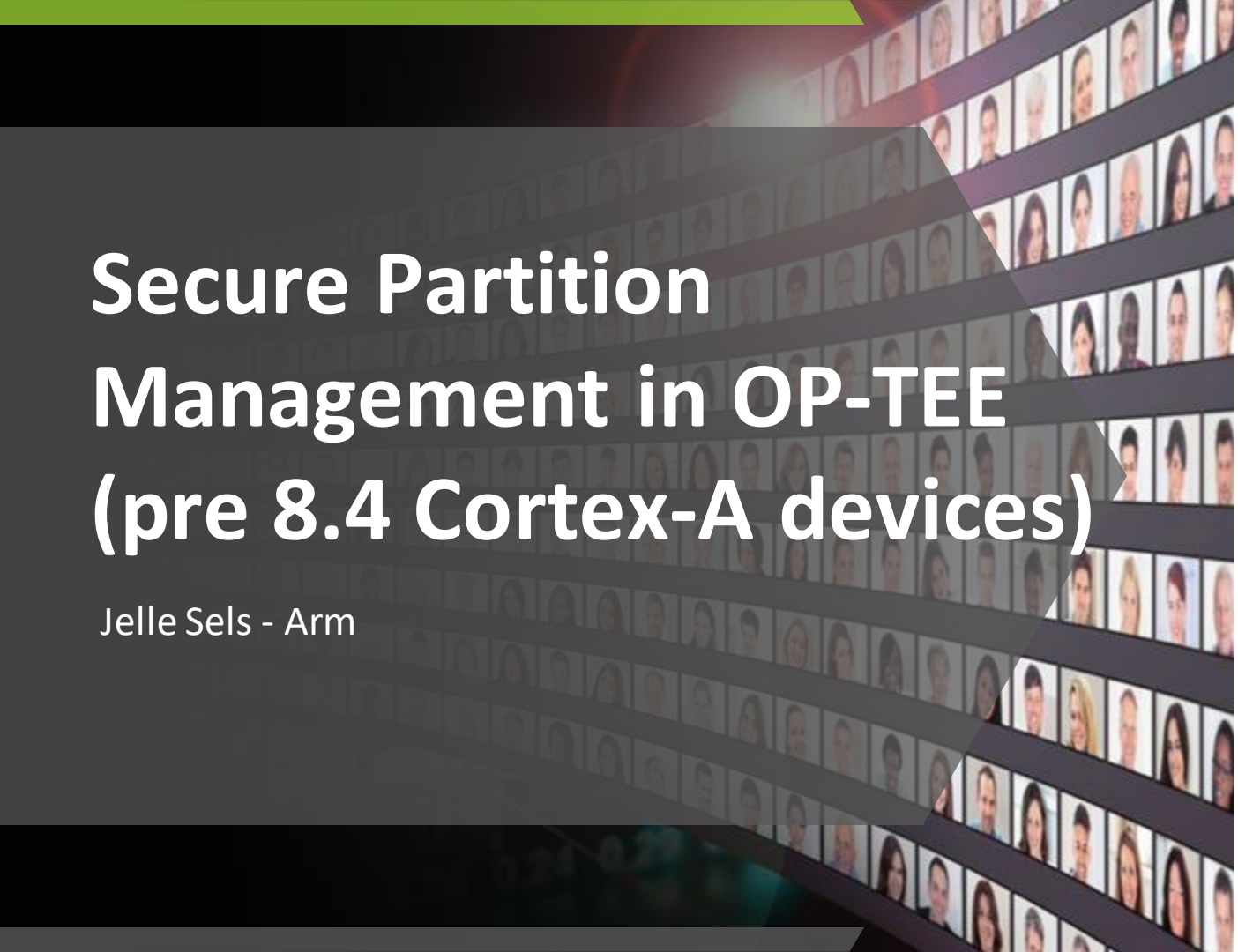


# Secure Partition Management in OP-TEE (pre 8.4 Cortex-A devices)

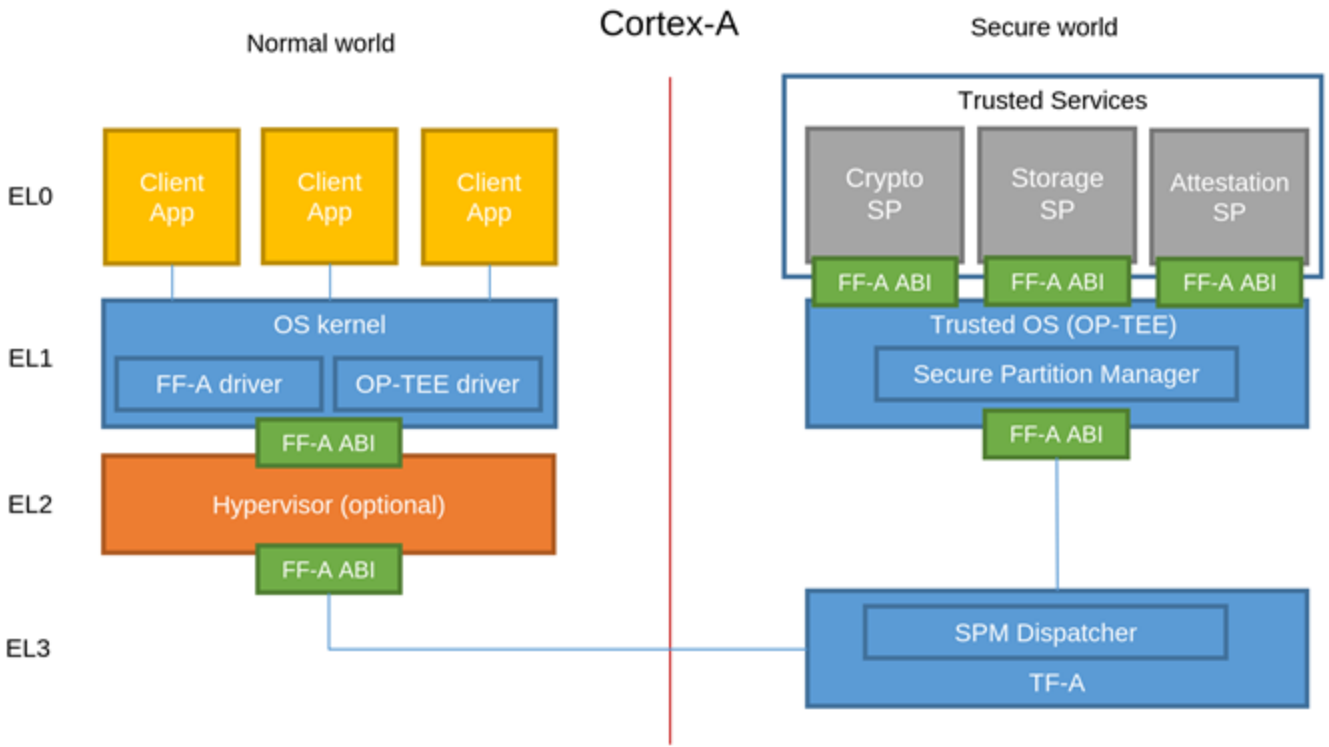
Jelle Sels - Arm

Linaro Connect
VIRTUAL • SPRING 2021

# What is FF-A?

- Firmware Framework for Armv8-A
  - https://developer.arm.com/documentation/ddi0487/latest/
- Defines a set of concepts an ABI's for handling Secure Partitions
- Secure Partition Manager Dispatcher (SPMD) is running in S-EL3 (part of Trusted-Firmware-A)
  - Routes messages between the Normal World and the SPMC
- Secure Partition Manager Core (SPMC) can run in S-EL1, S-EL2 or S-EL3
  - Responsible for managing Secure Partitions (SPs)

# FF-A pre 8.4 overview



# What is the SPMC?

- Secure Partition Manager Core
- Responsible for managing (SPs)
- OP-TEE is used as the standard S-EL1 SPMC implementation(pre 8.4 Cortex-A devices)
- Secure Partitions run in S-EL0 to ensure isolation between different SPs, without the use of a Secure Hypervisor
- OP-TEE implementation is based on FF-A version 1.0

# SPMC core responsibilities

- Loading and managing SPs
- Manifest data handling
- Managing Endpoint ids
- Inter-partition communication at run-time between
  - S-Endpoints
  - S-Endpoints and NS-Endpoints
- Rx/Tx buffer handling
- Memory management for SPs
- Device management
  - Interrupts
  - Memory regions

# Difference between TA's and SP's

## Trusted Applications

- Clean context for each invocation, user data is stored in sessions
- Can be dynamically loaded during run-time
- Multiple entry-points, run to completion
- Can have multiple sessions
- Uses OP-TEE syscall to communicate with outside world
- LibTA exports syscalls to Trusted Application

## Secure Partitions

- Are kept in memory, same data between invocation
- All SP's are loaded during OP-TEE initialization
- One entry point, resumes from where it was interrupted
- Can only have one active run-time
- Prefered way of communicate with outside world using FF-A messages
- LibSP exposes FF-A functionality to the Secure Partition

# Secure Partitions context

- Trusted services layer has been introduced
  - Abstraction layer which combines the TA and SP functionality
  - Makes it possible for SPs to use the standard OP-TEE functionality to load SPs, use OP-TEE memory management and have a SP SVC handler

- For each SP we have only one sp_session
  - sp_session and sp_ctx are used to store SP specific data
  - sp_sesson's are created ones at initialization and never deleted

- All general purpose registers values of the SP are stored in the sp_regs of the sp_ctx
- Open_sp_sessions linked list keeps track of all active SPs

# Secure Partitions initialization

- SP are loaded as the last step in the OP-TEE initialization process
- SP are embedded into the OP-TEE image
  - Same process as EARLY_TA
  - SPs are added to the SP_STORE at compile time
    - SP_PATHS in the upstream version
    - EARLY_TA in the proof of concept version
- Ldelf is used to load the SPs
- Patches in process to use SPs which are loaded by BL-2

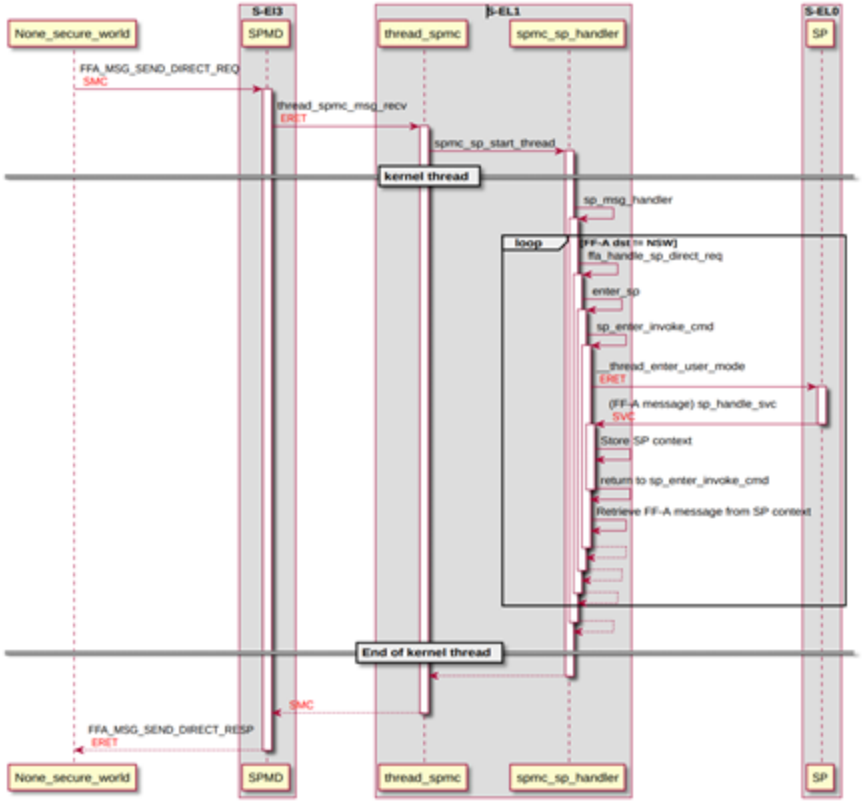
# OP-TEE SPMC components

The OP-TEE SPMC consists of 2 main components:

- Thread_spmc
  - Handles all messages from the Secure World
  - Handles all message for OP-TEE as a SP
  - Starts a new thread and runs the spmc_sp_handler when a DIRECT_REQ messages is received for a SP

- Spmc_sp_handler
  - Handles all FF-A messages from/to the SPs
  - Runs on a thread
  - Uses sp_enter to enter in the SP context

# FF-A messages flow



# Entering/Exiting SPs

- sp_enter is used to restore and continue the SP execution
  - FF-A arguments are copied into the SP execution context
  - Calls enter_invoke_cmd to restore the SP context and jump into S-EL0

- sp_handle_svc is called for each SP syscall
  - Stores the SP context
  - Returns to S-EL1
  - sp_enter copies the FF-A arguments from the SP context and passes it back to the spmc_sp_handler
  - spmc_sp_handler processes the new FF-A message

# LibSP

- Provides the basic framework to build SPs
- Exports all FF-A messages to the SP as C functions
- Part of the Trusted Services project
  - https://www.trustedfirmware.org/
  - https://connect.linaro.org/schedule/#
- Example SPs can be found in the Trusted Service project

# OP-TEE as a Secure Partition

- OP-TEE export itself as a Secure Partition
- Used to wrap OP-TEE specific calls into FF-A messages
  - SPMD only allows FF-A messages to be forwarded between the Secure and the Normal World

- FF-A messages are unpacked in the OP-TEE kernel and handled as normal OP-TEE calls
- Can be used to run Trusted Applications next to Secure Partitions.
- Can be used as a standalone S-EL1 SPMC or run under a secure hypervisor (S-EL2)
- Developed by
  - Jens Wiklander, Linaro
  - Marc Bonnici, Arm
  - Achin Gupta, Arm

# Proof of concept

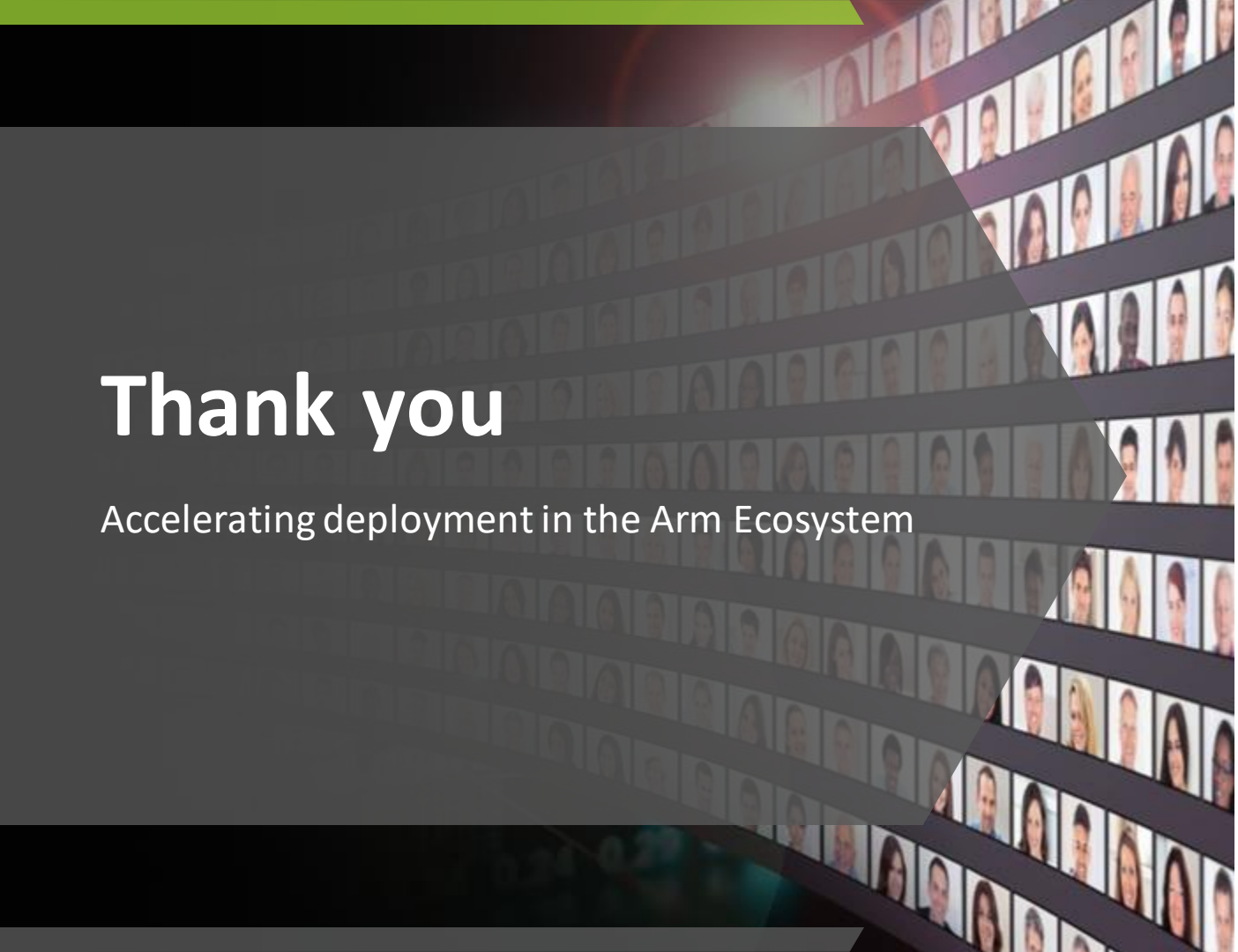
- Code
  - https://review.trustedfirmware.org/admin/repos/OP-TEE/optee_os
  - PSA-development branch
- Supported
  - Manifest file handling for device regions and interrupts
  - Handling DIRECT_REQ and DIRECT_RESP messages
  - RxTx buffer management
  - Managing Endpoint ids
  - Memory region mapping
  - Device region
  - S-EL1 SPs
  - Interrupts (in review)
  - RPMB (in development)

# Upstreaming to Github

- We are planning to upstream all functionality to Github
- Repository
  - https://github.com/OP-TEE/optee_os
- Some functionality already upstreamed
  - Loading and running SPs
  - Handling DIRECT_REQ and DIRECT_RESP messages
  - RXTX buffers management (currently in review)

# More information

- Mail to the OP-TEE mailing list
  - op-tee@lists.trustedfirmware.org
- Presentations
  - LVC20-112 PSA Secure Partitions in OP-TEE
  - LVC21-109: Introducing the Trusted Services project
  - LVC21-303: Secure Partition Manager evolution (Armv8.4 Secure EL2)
  - LVC21-305: OP-TEE as a Secure Partition running on SPM using ARMv8.4-A SEL2 feature



# Thank you

Accelerating deployment in the Arm Ecosystem

Linaro Connect
VIRTUAL • SPRING 2021