

A row of wind turbines stretches across a beach at sunset. The sky is a mix of blue and orange, with scattered white clouds. The ocean waves are visible in the foreground. The overall scene is serene and modern.

arm

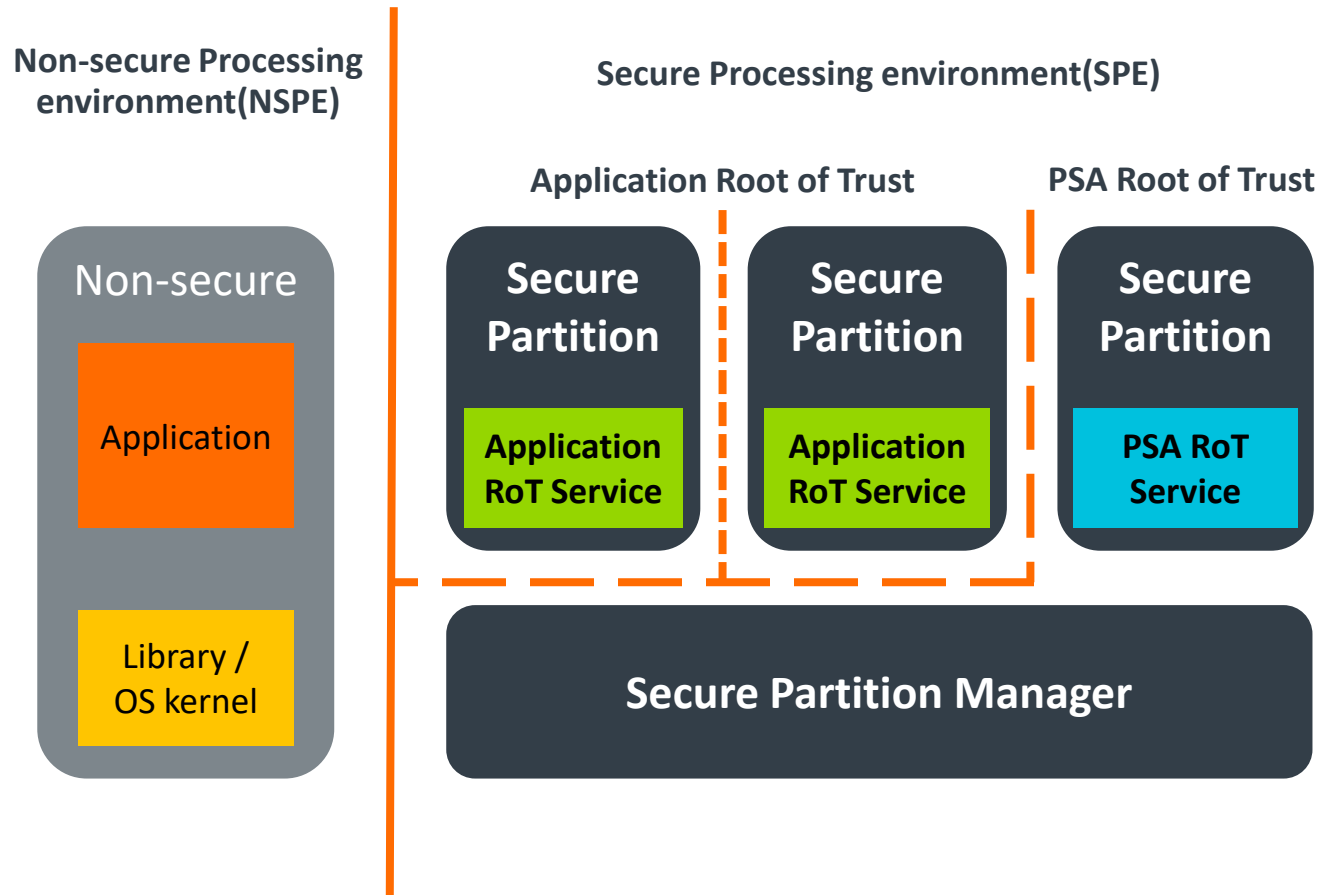
# Secure Partition Runtime Library on IoT Device

Ken Liu, Edison Ai

# Content

- Platform Security Architecture(PSA) Firmware Framework
- What is in Secure Partition Runtime Library?
- Placement of the Secure Partition Runtime Library
- Secure Partition Runtime Library Design
- Project Status

# Platform Security Architecture(PSA) Firmware Framework



## What is Secure Partition?

- Unit of isolation
- Execution environment for RoT service:
  - Provide access to resources
  - Protect code and data
  - Mechanisms to interact with other components

- **Isolation level 1** - Protect SPE from NSPE
- - - - - **Isolation level 2** - Protect PSA RoT from App RoT
- . - . - . **Isolation level 3** - Protect each Secure partition from other secure partition

# What is in Secure Partition Runtime Library?

## C Runtime:

- printf/assert
- malloc/free/realloc
- memcmp/memcpy/memmove/memset

## PSA APIs:

- Client APIs
- Secure partition APIs
- RoT service APIs

## Secure Partition

## Miscellaneous APIs:

- Boot data API
- Others

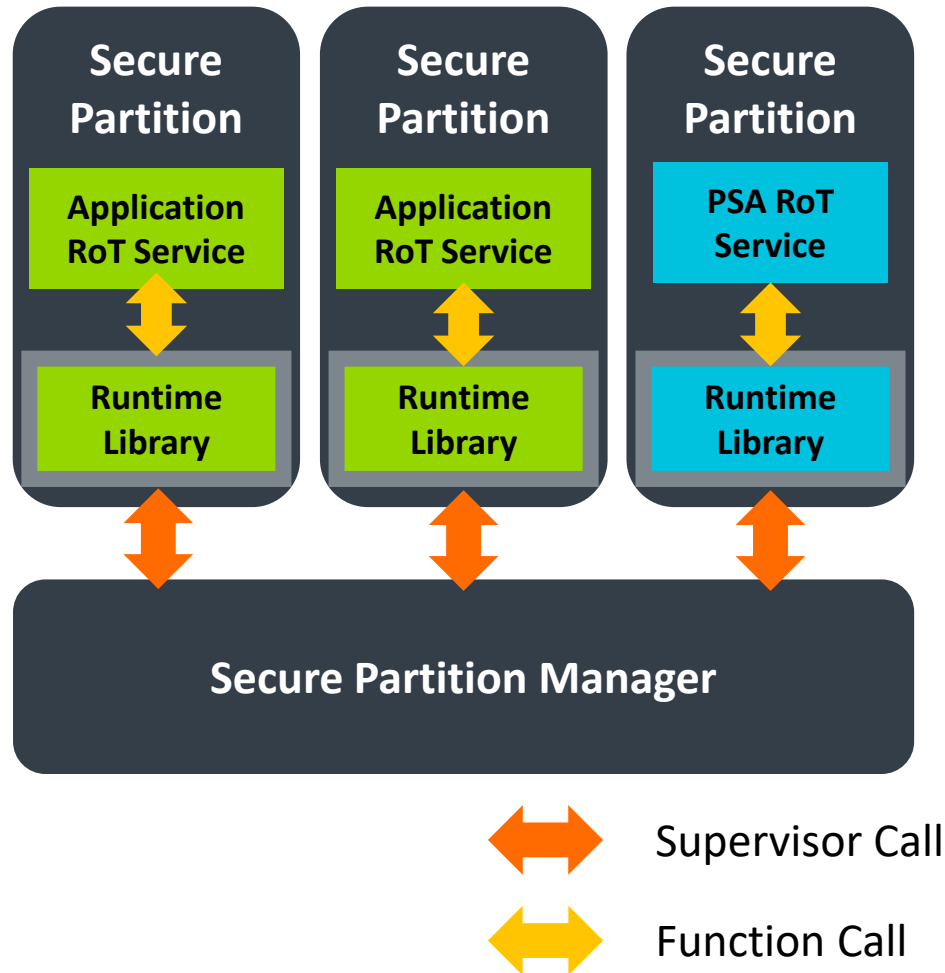
## Secure partition launcher:

- Initial necessary data for secure partition before entering main entry

- But all these APIs MUST be implemented with security consideration.
- What is requested in PSA FF:
  - Only Code is executable
  - Only private data is writable
  - Private data must be isolated

# Placement of the Secure Partition Runtime Library

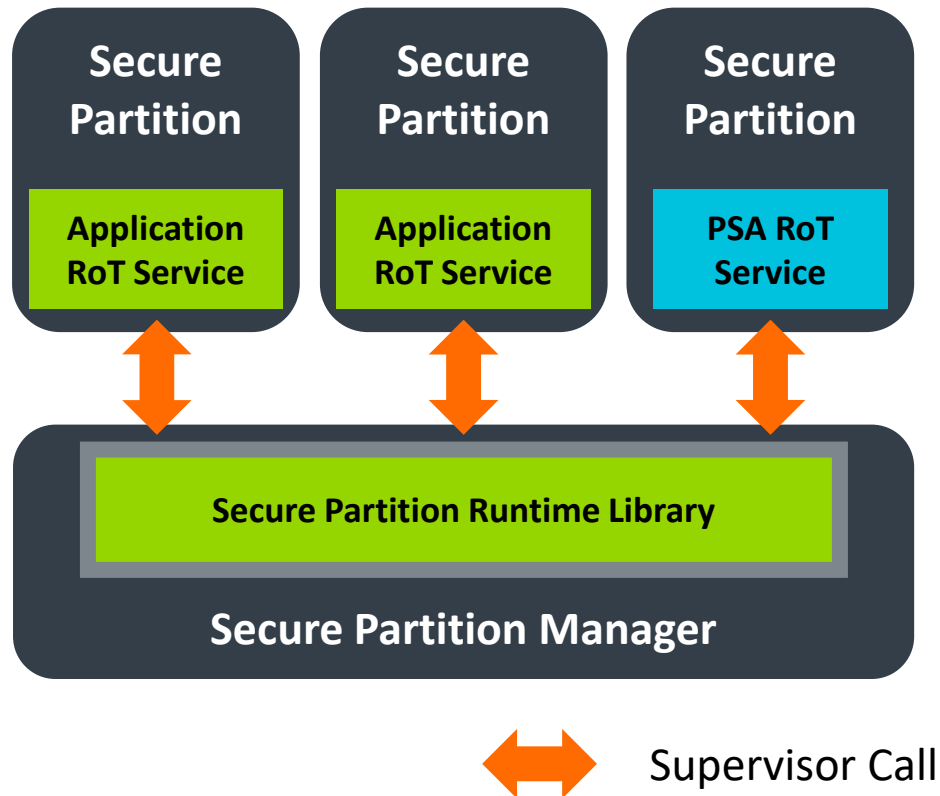
## Option 1: Per-Partition Library



- Image size increases much.
- Hard to put into a single loaded Image.

# Placement of the Secure Partition Runtime Library

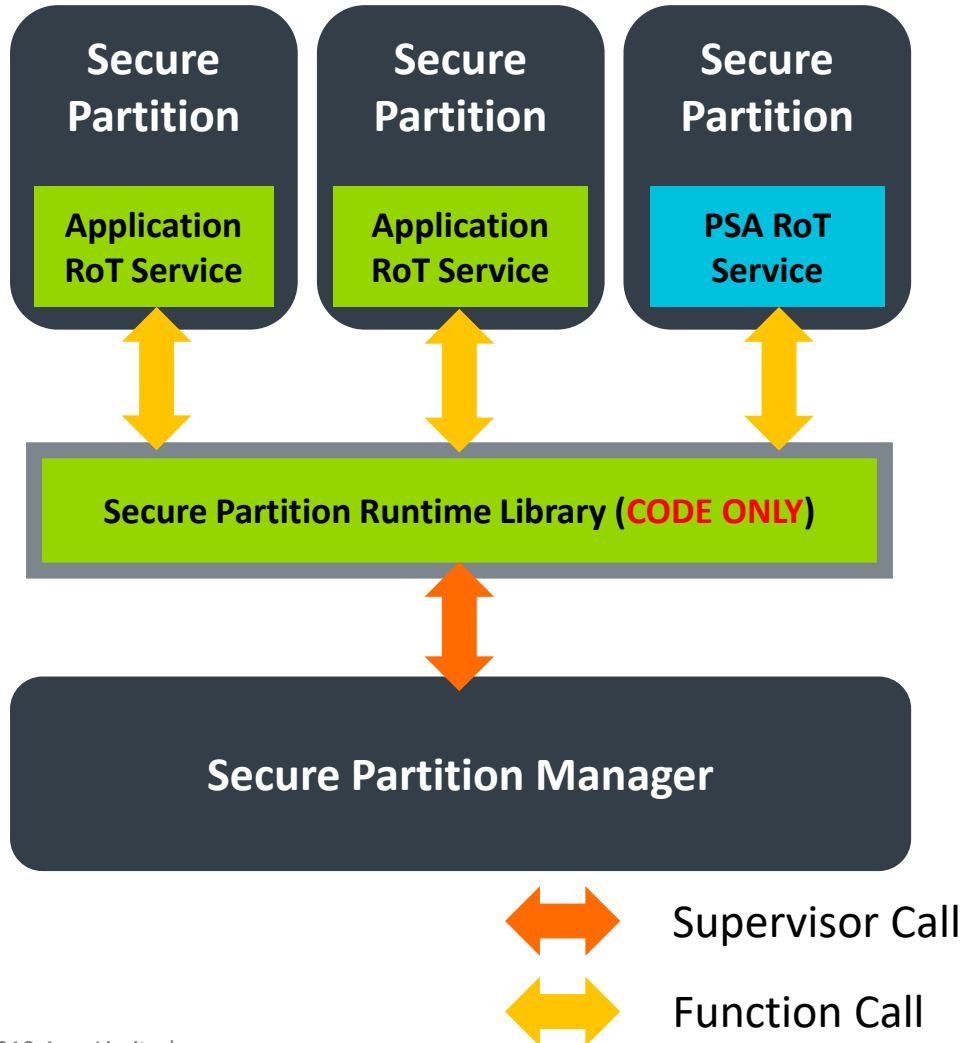
## Option 2: All by Supervisor Call



- SPM consume more execution time.
- Library execution cannot be preempted.
- Library code runs under unnecessary privileged level.

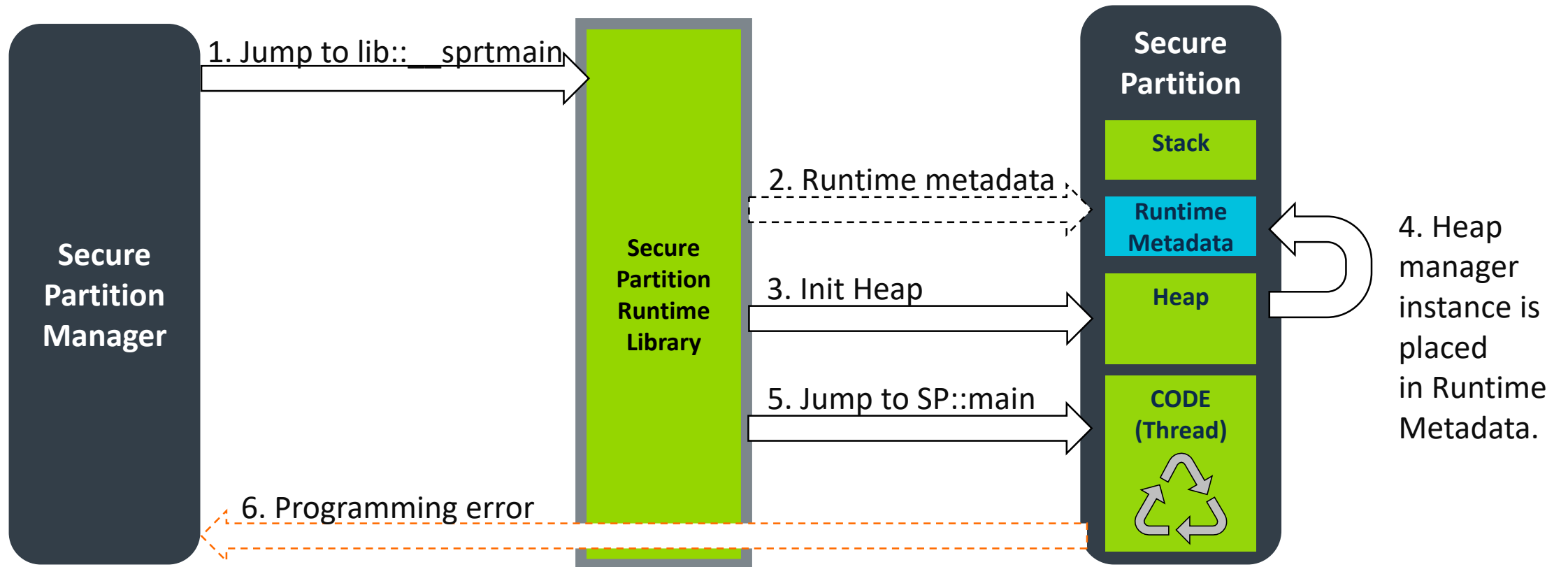
# Placement of the Secure Partition Runtime Library

## Option 3: Partition Shared Library



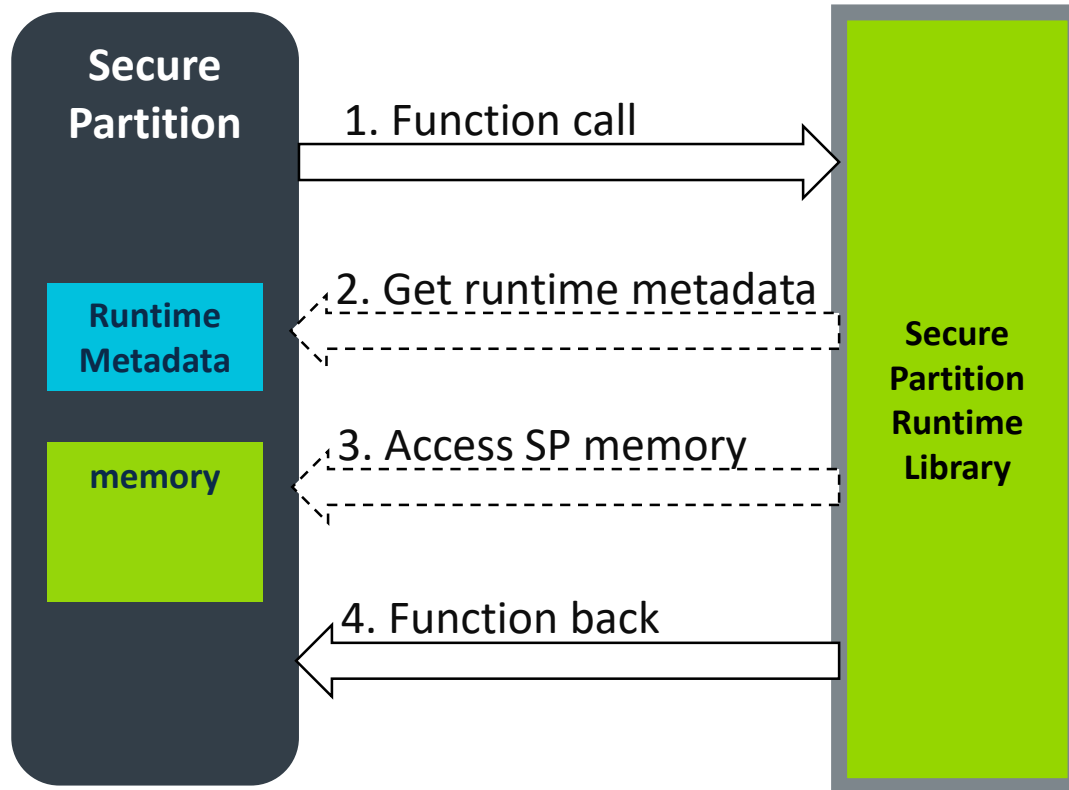
- Fast and efficient.
- Dedicated MPU region needs to be reserved for library.
- A well-balanced implementation.
- THIS IS IT.

# Launch Partitions with Secure Partition Runtime Library



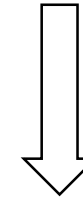


# Function call to Secure Partition Runtime Library



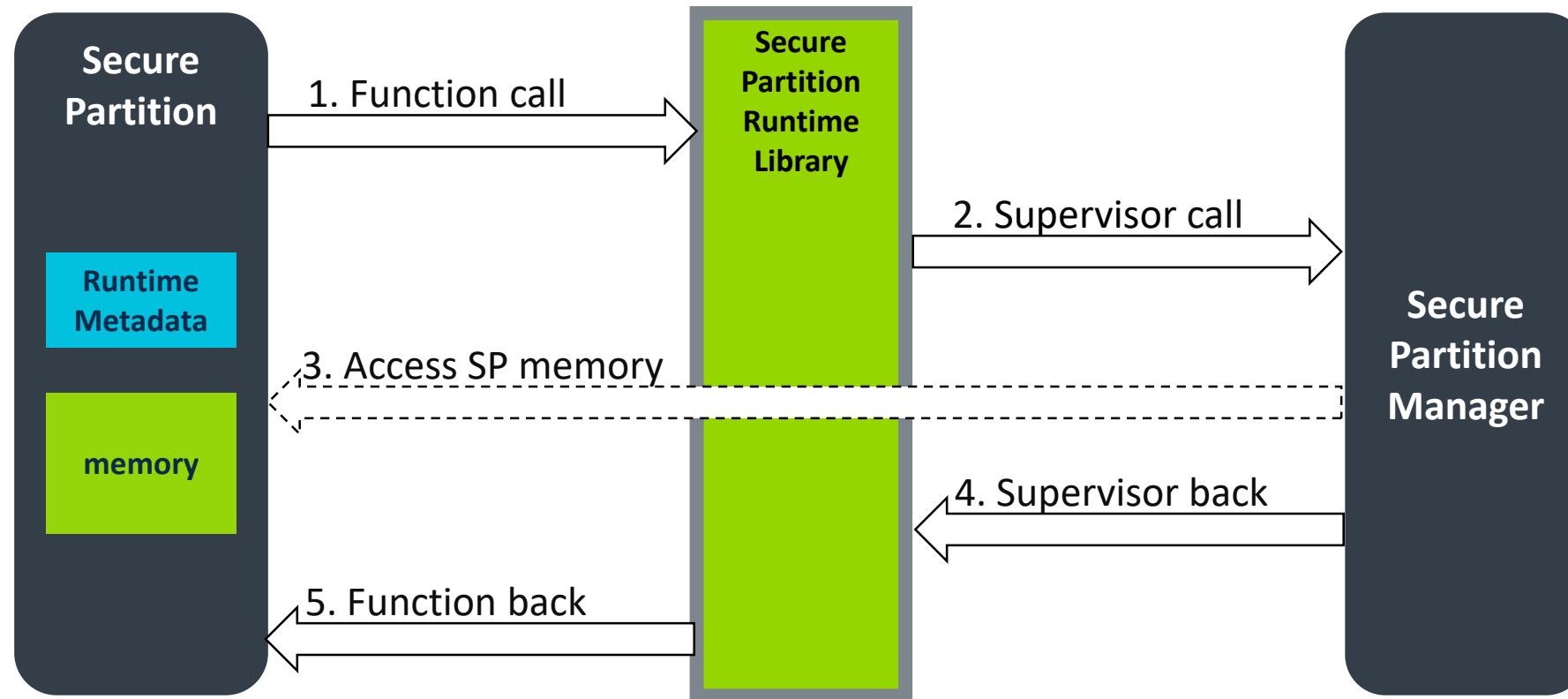
Function with Implied Parameters Passing:

```
void *malloc(size_t sz)
```



```
void *_malloc_impl(size_t sz, void *p_inst)
```

# Supervisor Call of Secure Partition Runtime Library



# Project Status

- Secure Partition Runtime Library Design had been public: [https://git.trustedfirmware.org/trusted-firmware-m.git/tree/docs/design\\_documents/tfm\\_secure\\_partition\\_runtime\\_library.rst](https://git.trustedfirmware.org/trusted-firmware-m.git/tree/docs/design_documents/tfm_secure_partition_runtime_library.rst)
- Secure Partition Runtime Library APIs status
  - PSA FF APIs – 60%
  - C Runtime – 30%
  - Secure Partition launcher – 10%
  - Miscellaneous APIs – 30%
- TF-M Source code: <https://git.trustedfirmware.org/trusted-firmware-m.git/>

arm

Q & A

arm

Thank You

Danke

Merci

谢谢

ありがとう

Gracias

Kiitos

감사합니다

धन्यवाद

شكرًا

תודה