Trusted Firmware-M and Hybrid platforms

Choose proper features for cases

TF-M

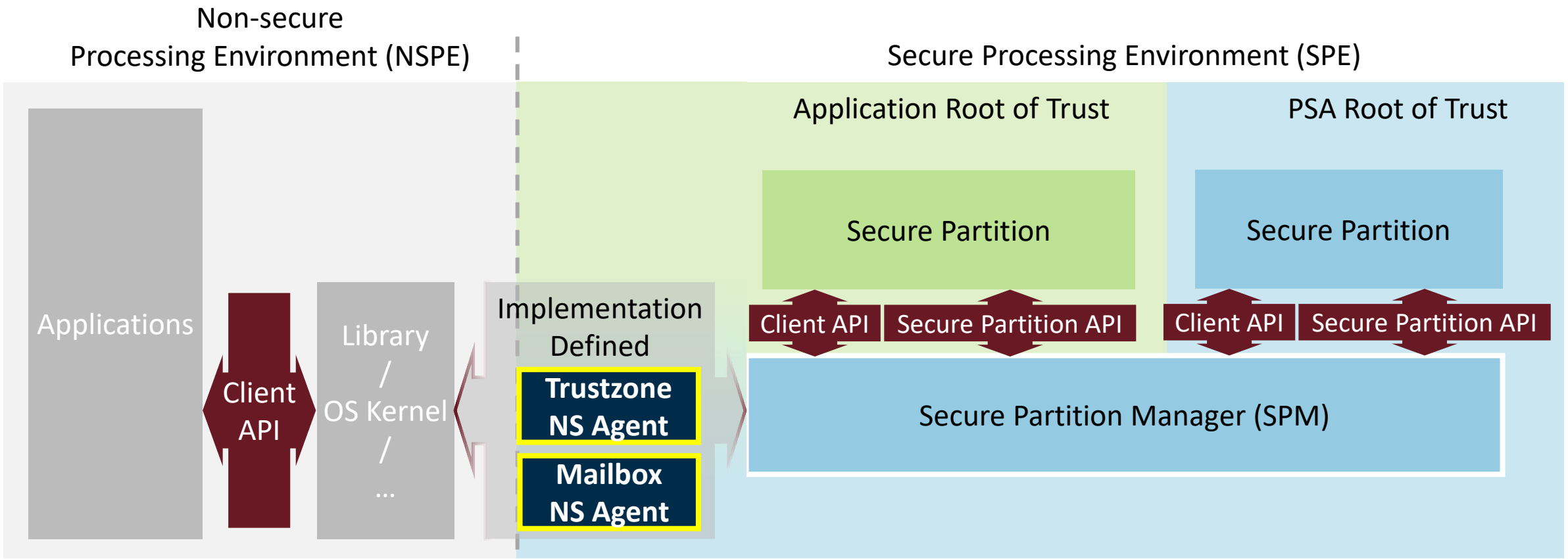
2023 Sep

Terminology

Mailbox	This item represents the inter-core communication mechanism includes both hardware and software.
Client	Firmware Framework-M (FF-M) concept, a role that accesses secure services. A client can be a non-secure role such as non-secure program, or it can be a Secure Partition accessing depending services.
Local client	The client is in the same core as the secure services and accesses secure services via. Trustzone-M
Remote client	The client is not in the same core as the secure services and accesses secure services via. the mailbox.
NS Agent	A component who converts customized Non-secure client access into FFM-compliant access.
Hybrid Platform	A platform containing A and M-profile or multiple M-profile cores.

Check FFM for other FFM defined concepts, such as SPE, NSPE, Secure Partition and SPM.

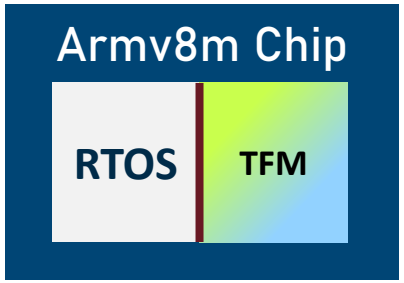
Firmware Framework-M (FF-M) and Trusted Firmware-M (TF-M)



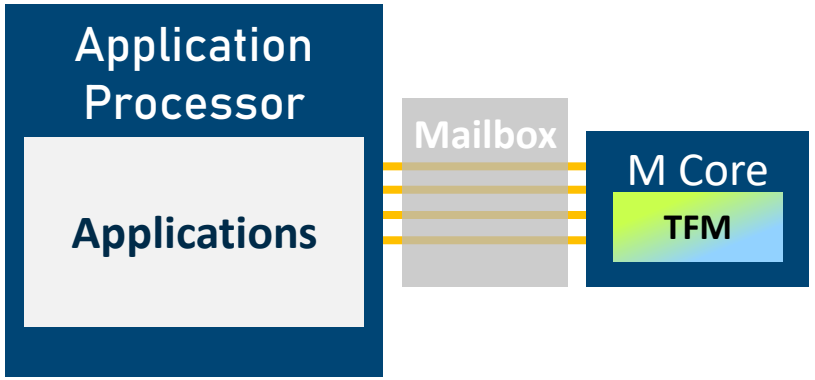
TF-M complies to FF-M and has its own Implementation-defined components (■).

arm

Integration examples



Single Arm v8-M Chip with Trustzone-M

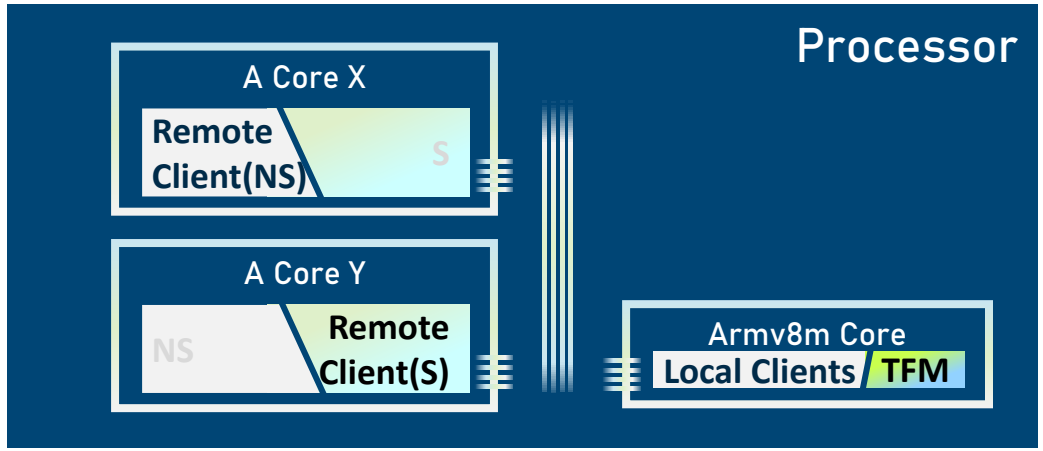


M-profile core and Application Processor (M or A-Profile) interact via mailbox

Common Background Info

Multiple Clients	TF-M shall handle requests from multiple clients via FFM Client APIs and each NSPE client shall be uniquely identifiable.
Local Client real-time support	Considering the real-time requirements of NSPE, SPE can be preempted by NSPE events to response in time.
Local Client concurrency	SPE passive secure context management: NSPE assists secure context management by calling secure functions.
Remote Client serving performance	SPE works on a standalone core, it has bandwidth and capacity limit naturally. Hence it is hard to define a hard limit for service response time especially in multiple clients' cases.
Clients' trust independence	The Non-secure clients (Both remote and local clients) do not trust each other. Hence, one non-secure service access request can't be forwarded by another non-secure client.

Hybrid Platforms and scenarios brought



- The platform has a processor with A+M or M+M cores. The M core for secure services with Trustzone-M supported.
- TF-M running in Trustonze-M secure state.
- Both local and remote clients need to access secure services in TFM.

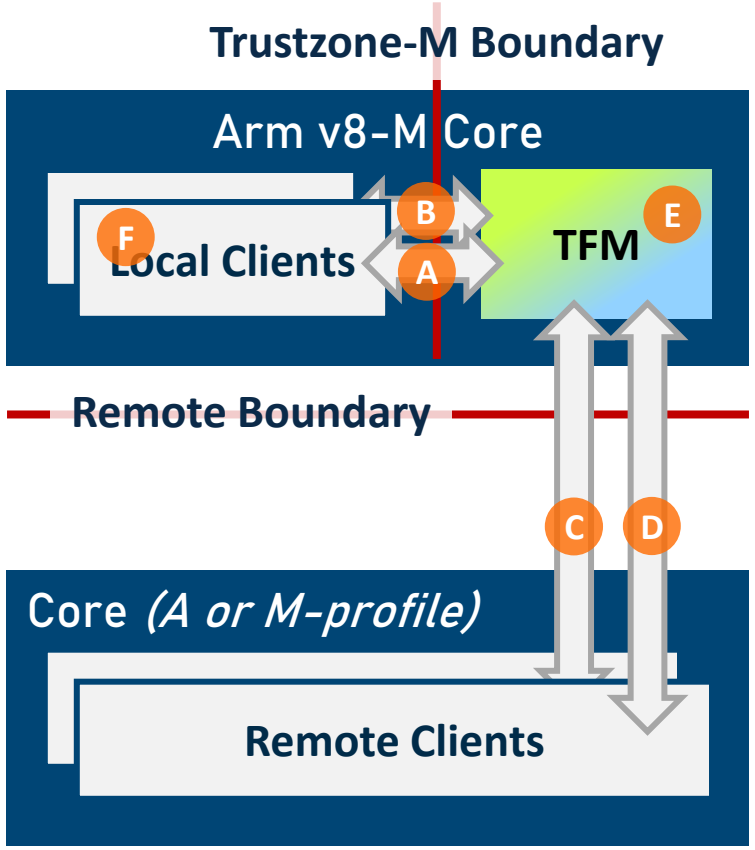
- Example scenarios:
  - TFM handles requests from remote secure context directly and act as main security gatekeeper – Quick remote response.
  - NSPE of M (Local clients) contains offload calculation operations and then expects less impact from SPE.

Use cases recap since last forum

Hybrid platforms bring quite difference user cases as it involve two client types.

Following the discussion in TF-M tech forum on June 22.
Items discussed:

1. Multiple clients support for both Trustzone-M (A,B) and remote communication (C,D).
2. Coexistence of local and remote clients (A,B,C,D).
3. Specific real-time oriented services in TF-M (E).
4. Real-time feasibility for local clients (F).
5. SPE releases execution context during idle operations.



Use Cases -> Requirements: Solid items

No	Use case	Requirements
1	Remote and local clients coexist	A system shall be able to serve both local and remote clients in a single configuration
2	Secure service interrupts and real-time characteristics	Secure service interrupts shall be handled within the time constraints. The time-bound shall be short enough for most real-time purposes. (Not Hybrid Platform specific but emphasized in the last discussion).

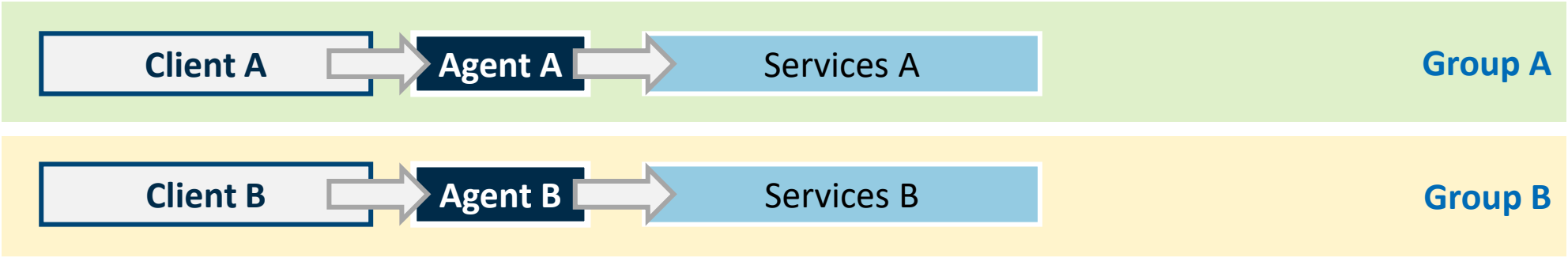
Use Cases -> Requirements: Flexible items

No	Use case	Requirements	Comments
4	Real-time feasibility for local clients	As mentioned in general requirements.	Reason for ‘flexible’: There are also scenarios that NSPE has no real-time requirements.
5	SPE releases execution to local NSPE during SPE IDLE	IDLE: When SPE needs to wait for interrupts/events. This increases the execution efficiency for NSPE.	• An important requirement for specific hybrid platforms. • Need extra integration actions.
6	In time response for remote clients	As mentioned in general requirements, remote clients expects to be serviced in a pre-defined time.	User cases depended. A ‘reliable’ response might have higher priority than ‘fast’.

Balance the requirements

- The flexible items shows various dependency inputs (scenarios and so on).
- But a M-core can’t afford a ‘Have them all’ solution.
- Remote M-core might be designed with physical isolating consideration, and Trustzone-M is a plus isolation inside M-core.
  - It is possible to assign Local NSPE software with security attribute for this case.

- **Balance the requirements and then choose features**
  - Feature for identifying the requirements to be balanced: Grouping clients and their respective depending services by TF-M dependency policy.



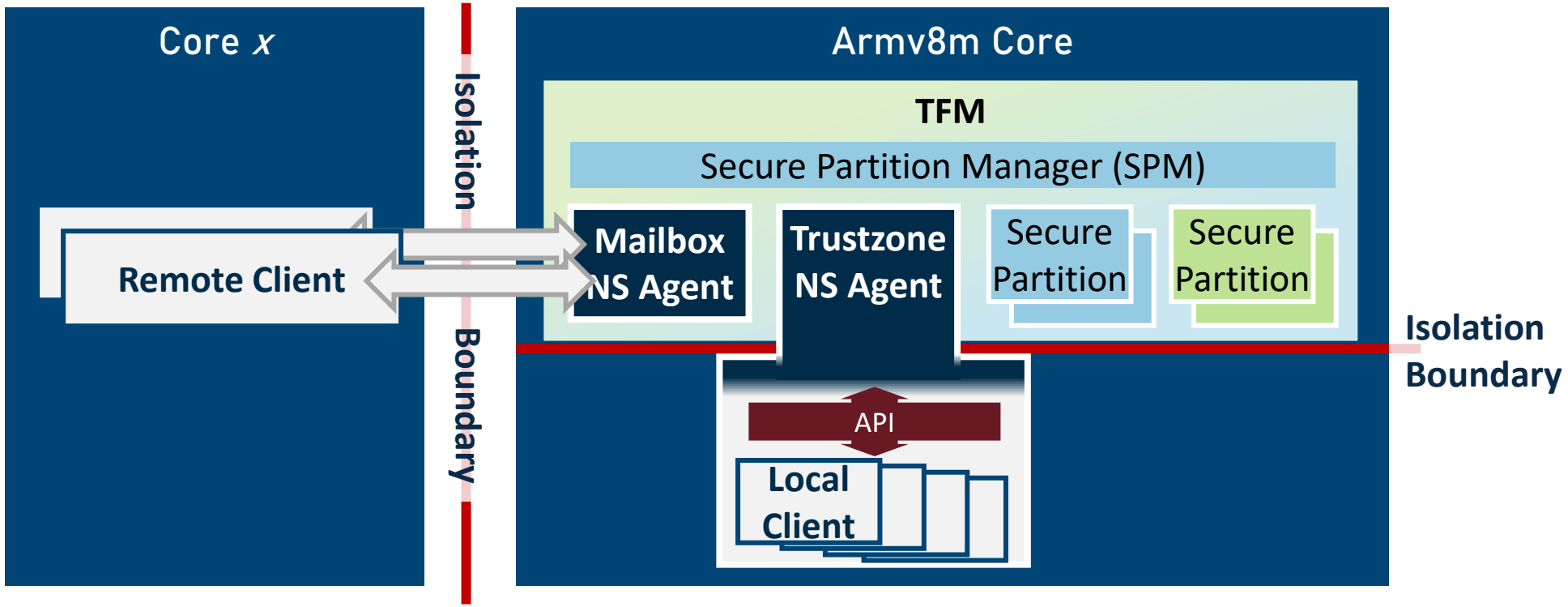
arm

3 solutions proposal

1. SPE schedules the whole system
2. NSPE schedules the whole system
3. Priority-boosted Mailbox NS Agent

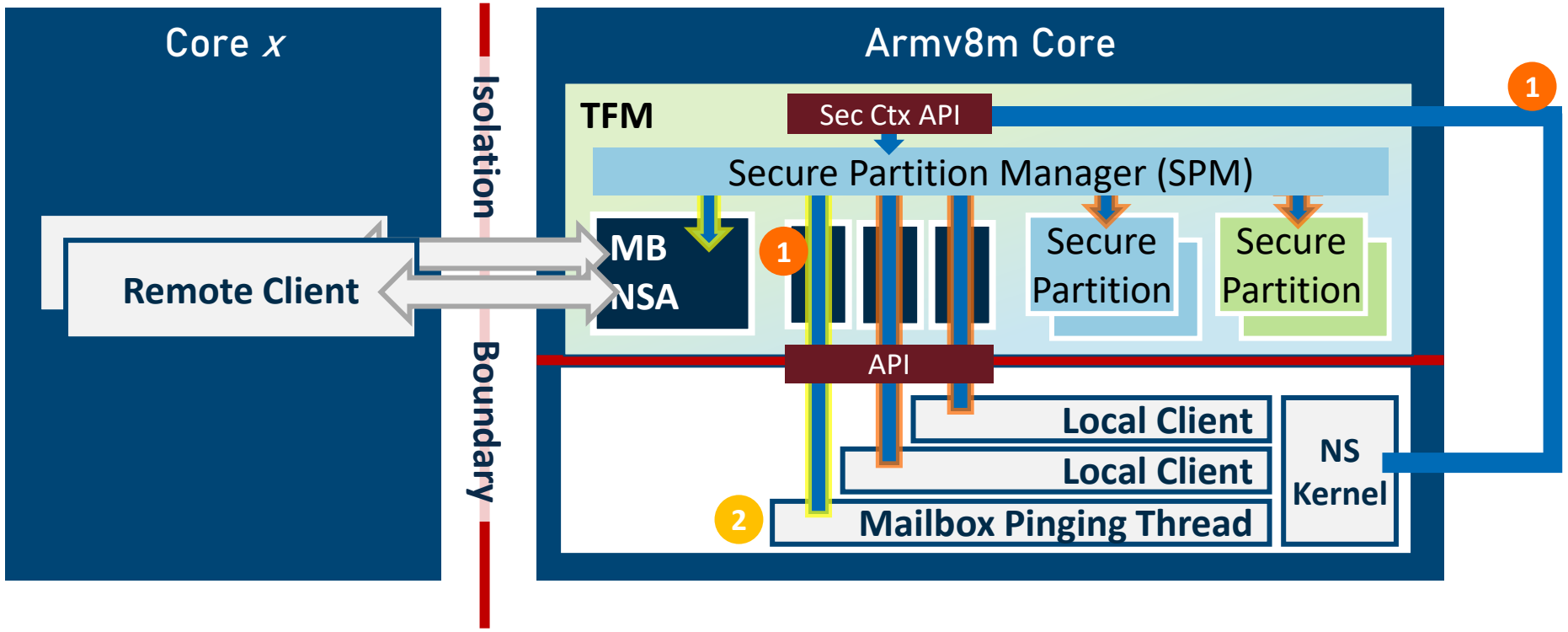
And the revisit to the NSPE focused solution.

Solution 1: TFM schedules the whole system



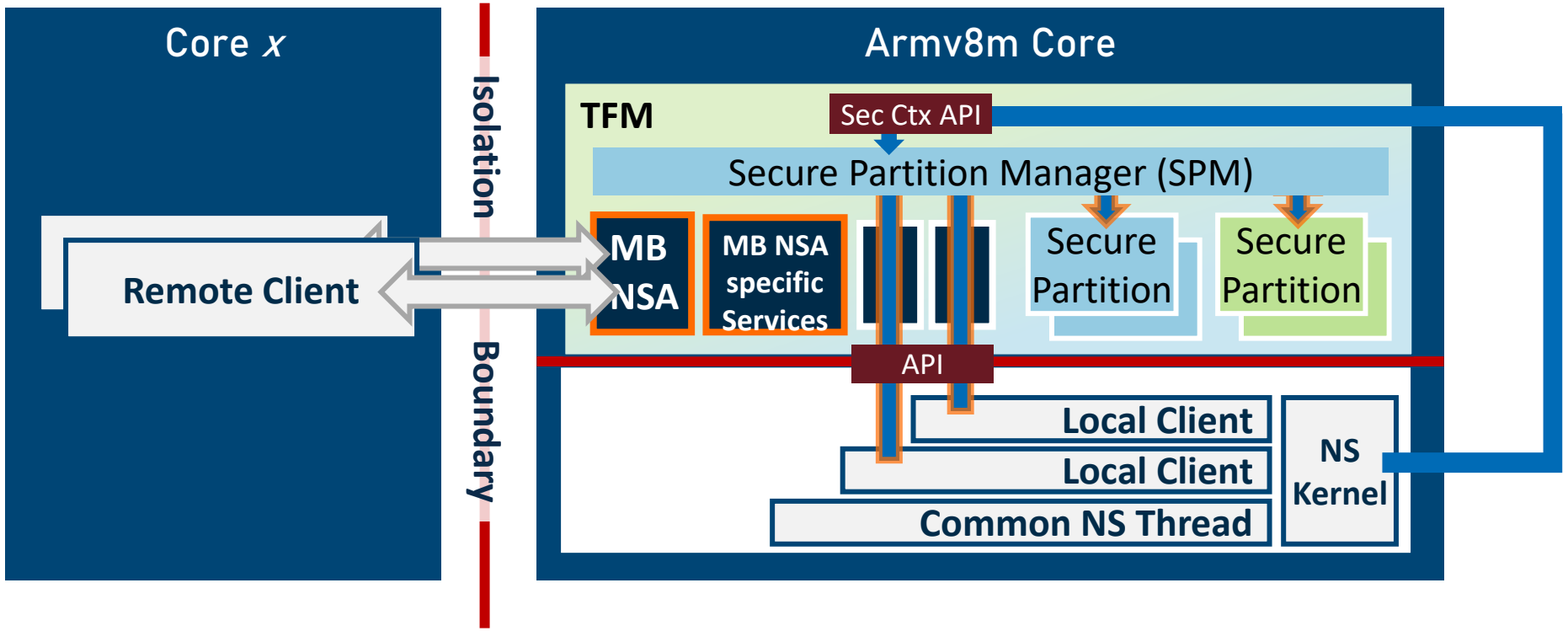
- SPM schedules everything, by boosting the secure execution priority.
- Easy to integrate and use.
- Trustzone NS Agent acts as a virtual machine.
- Besides Client API, extra mechanism is still needed to tell SPM which Local Client is running.
- Local NSPE software is unlikely to be a generic RTOS.

Solution 2: NSPE software schedules the system



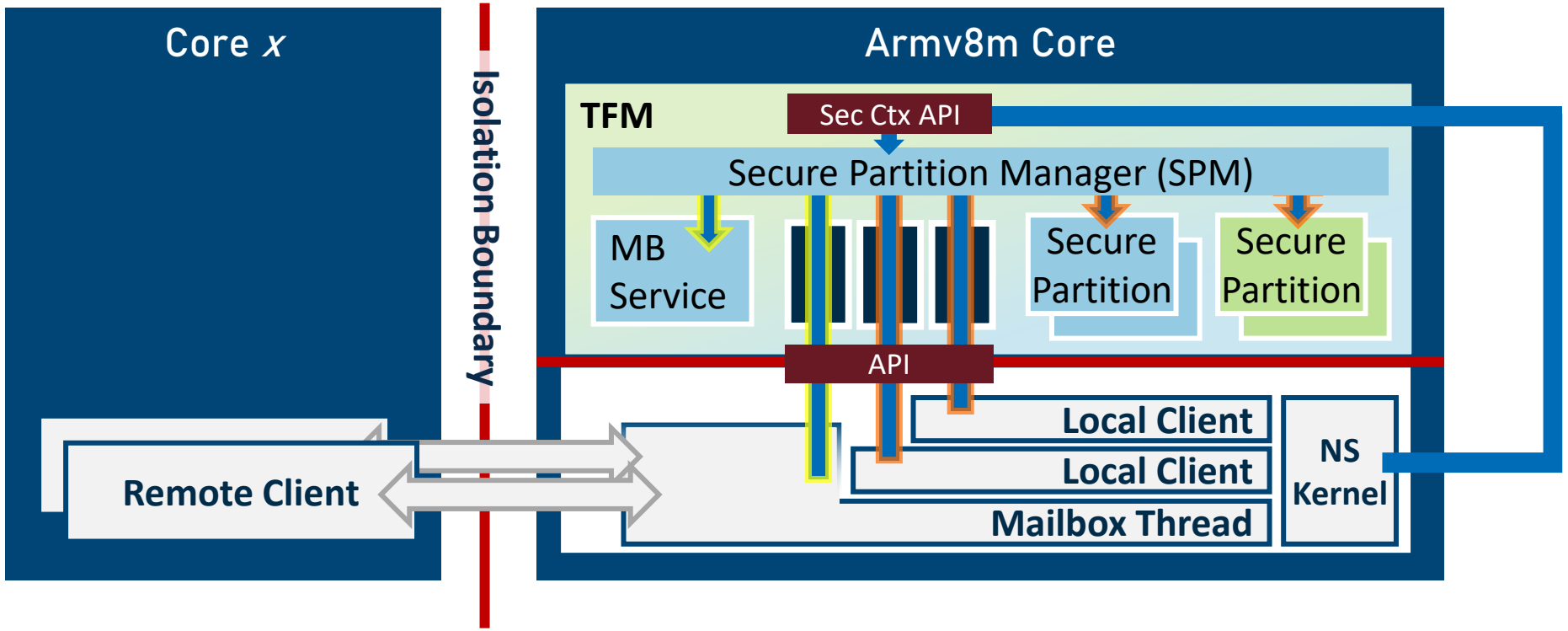
1. Each local client has a dedicated secure context in Trustzone NS Agent when accessing secure services. Secure services can be preempted whenever non-secure interrupts happen.
2. **Extra integration item**: A dedicated NSPE thread needs to be available to call a ‘Pinging’ service to wake up the mailbox message loop.

Solution 3: Priority-boosted Mailbox NS Agent



1. Similar to solution 2, but boosted Mailbox NS Agent's execution priority in case it never gets scheduled when the system is busy.
2. **Extra integration item**: The same as solution 2. And no doorbell thread needed.
3. This solutions involves remote specific services to avoid the remote being blocked by local clients.

Revisit the generic NSPE focused solution



1. Parts of mailbox work moved to local NSPE (Interrupts or more).
2. When secure operation required, call into Mailbox Secure Service.
3. It can be generic enough if just some secure-key related crypto operations are required. The service can even be saved.
4. It is a generic TFM usage so won't be listed in the comparison table of the following pages.

Comparisons

Topic	Solution 1	Solution 2	Solution 3
Local client real-time support	Not guaranteed	Guaranteed	Balanced
Remote client response in-time	Guaranteed	Not guaranteed	Balanced
Additional TFM features required	• Secure context management • Priority configuration options	• Secure context management	• Secure context management
NSPE integration items (Local and remote)	• Call secure context management interfaces when scheduling. • Adapt with Mailbox NS Agent	• Call secure context management interfaces. • Adapt with Mailbox NS Agent • A dedicate thread to wake mailbox NS agent up (One more mailbox NS Agent, one more dedicated thread).	• Call secure context management interfaces. • Adapt with Mailbox NS Agent • A dedicate thread to wake mailbox NS agent up (**optional**).

TFM features to support the proposed solutions

Feature	Solution 1	Solution 2	Solution 3
Secure Context Management	Required	Required	Required
Mailbox NS Agent API	Required	Required	Required
Secure priority boost	Required (Full SPE)	Not required	Required (Specific mailbox agent only)
Agent wakeup service	Not required	Required	Optional (Nice to have)
Release execution when IDLE	Not required	Required	Optional (Nice to have)

1. Dedicated solutions can be activated by combining related features.
2. Solution 1 is the reference solution; others need extra integration tasks.

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