



arm

# TF-M eRPC Test Framework

Kevin Peng  
Feb 2023

# Agenda

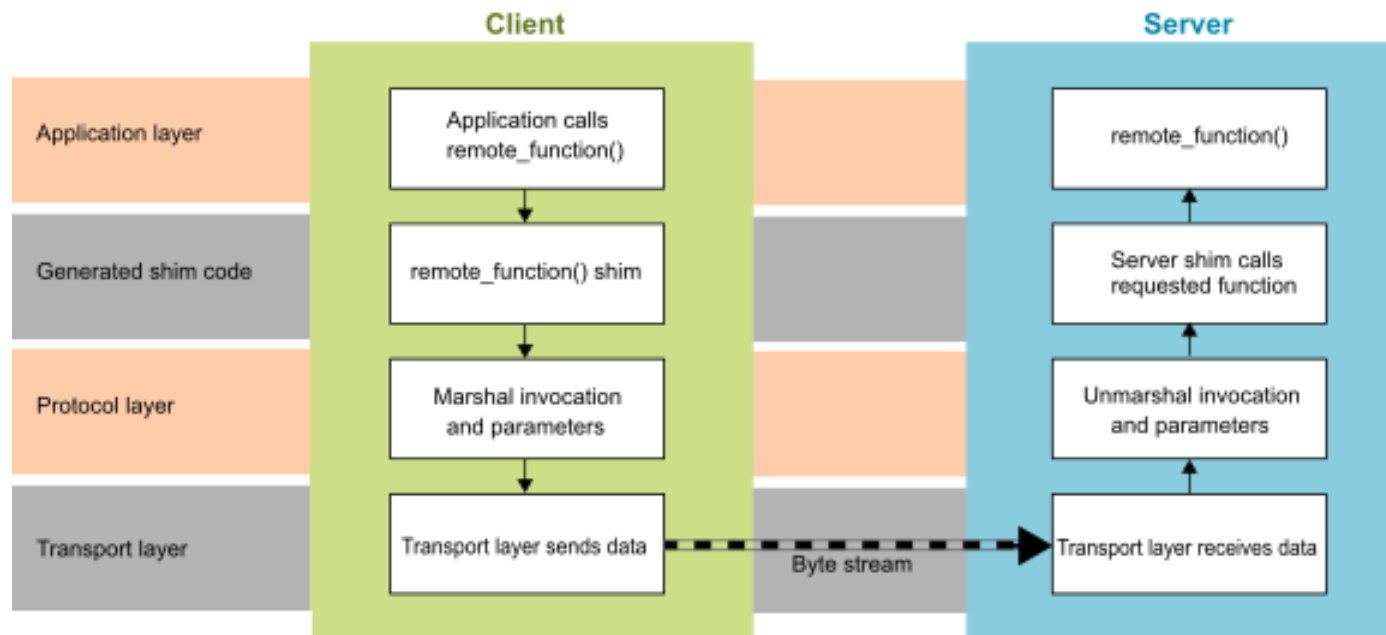
- + Background
- + Introduction to RPC and eRPC
- + The TF-M eRPC Test Framework
- + The eRPC Gen Tool
- + Usage & Integration
- + Current Status and Future Plans
- + Q & A

# Background

- + Constraint memory resources on M devices
- + Sometimes size of test suites would be large that could not fit into the devices
- + Have to split the test suites and download image and run for multiple times
- + Time consuming and problematic for collecting final results
- + The eRPC test framework is to solve these issues

# RPC and eRPC

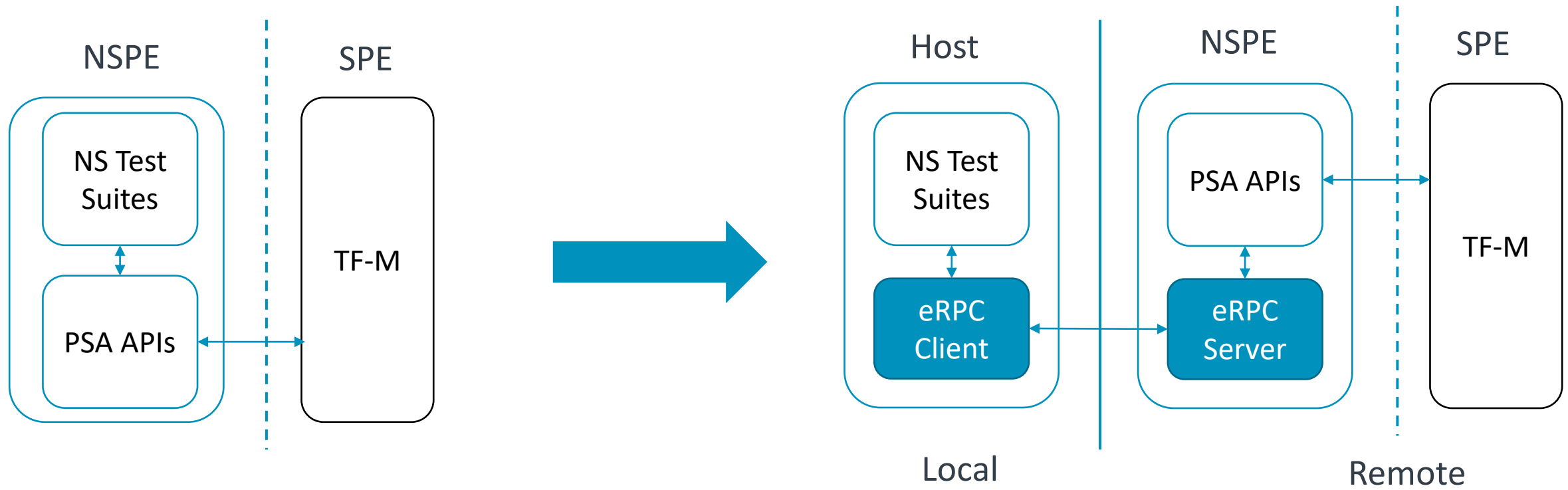
- + Remote Procedure Call – Calling software on another device
  - The codes are the same whether the execution is local or remote
  - Client-server interaction
- + eRPC (Embedded RPC) is an open-source Remote Procedure Call (RPC) system
  - Lightweight
  - Auto-gen of shim layers



# TF-M eRPC Test Framework

## + RPC APIs

- NS PSA Client APIs ONLY
- Secure APIs in the future



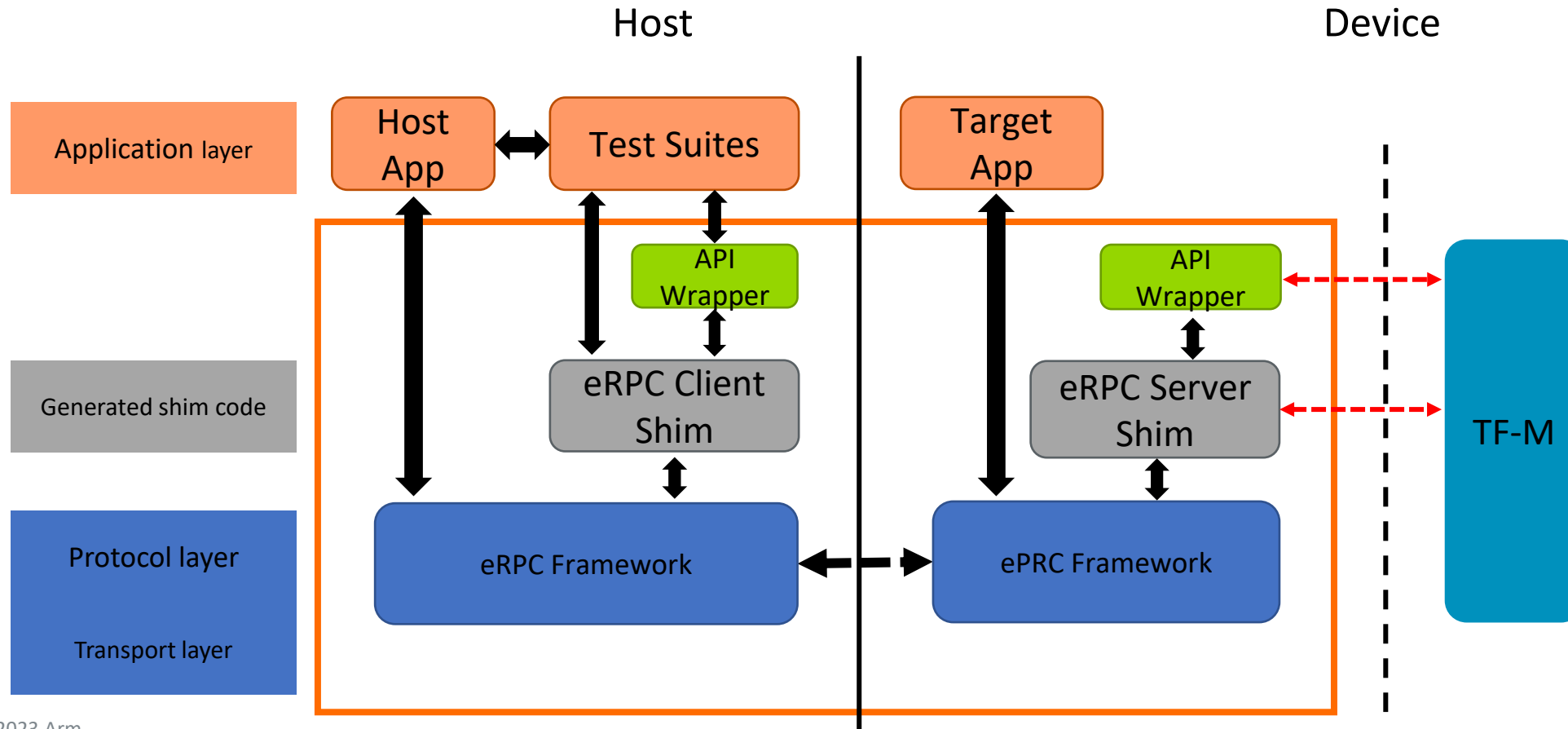
# Benefits

- + Off-load the memory footprint on the resource constraint M-class devices
- + Host gets the test result by return values instead of parsing the test logs
- + Updating test codes without downloading new images to devices

# TF-M eRPC Test Framework – Cont'd

## + API Wrapper

- The eRPC does not support all types of parameters
- The `psa_call` API needs a wrapper to convert its parameters the ones that the eRPC supports



# The eRPC Gen

The tool helps to generate eRPC shim layers

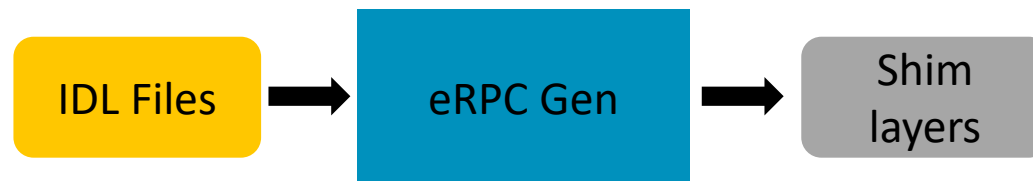
```
@c:include("psa/client.h")

program tfm_erpc

@external type psa_handle_t = int32
@external type psa_status_t = int32

@group(psa_client_api) interface psa_client_api {
  ... psa_framework_version() -> uint32
  ... psa_version(uint32 sid) -> uint32
  ... erpc_psa_call(psa_handle_t handle, int32 t, list<binary> erpc_in_vec, inout list<binary>
    erpc_out_vec) -> psa_status_t
}

@group(psa_connection_api) interface psa_connection_api {
  ... psa_connect(uint32 sid, uint32 ver) -> psa_handle_t
  ... psa_close(psa_handle_t handle) -> void
}
```





# Usage and Integration of the TF-M eRPC Framework

- + Enable the `CONFIG_TFM_ERPC_TEST_FRAMEWORK` config to build the framework.
- + On the device side, a target app is provided and enabled by default to start the eRPC service on boot
- + On the client side
  - An `erpc_client` CMake library is provided for applications for build part
  - Applications is responsible for initializing the transportation layer
  - Then call the `erpc_client_start(erpc_transport_t transport)` interface to start the client

# Current Status and Future Plans

## + Current status

- The eRPC test framework is [upstreamed](#)
- Working on porting the NS test suites to host
- Working on documentations

## + Future plans

- Investigation to support Secure tests
- Interactive mode

arm

Q & A

arm

Thank You

Danke

Gracias

Grazie

谢谢

ありがとう

Asante

Merci

감사합니다

धन्यवाद

Kiitos

شكرًا

ধন্যবাদ

תודה