

Default implementation of tfm_ns_interface_dispatch()



Current state of the code

-  - Provided by TFM
-  - Expected to be provided by user

PSA functions (e.g. *psa_connect*, *psa_call*, ...)



tfm_ns_interface_dispatch() which should provide:

- Multithread safety based on the NS OS (or bare metal implementation)
- Other application needs

Example implementation

tfm_ns_interface_dispatch()

```
int32_t tfm_ns_interface_dispatch(vneer_fn fn,
                                uint32_t arg0, uint32_t arg1,
                                uint32_t arg2, uint32_t arg3)
{
    int32_t result;

    /* TFM request protected by NS lock */
    while (os_wrapper_mutex_acquire(ns_lock_handle, OS_WRAPPER_WAIT_FOREVER)
           != OS_WRAPPER_SUCCESS);

    result = fn(arg0, arg1, arg2, arg3);

    while (os_wrapper_mutex_release(ns_lock_handle) != OS_WRAPPER_SUCCESS);

    return result;
}
```

tfm_ns_interface_dispatch()

```
uint32_t tfm_ns_interface_init(void)
{
    void *handle;

    handle = os_wrapper_mutex_create();

    if (!handle) {
        return OS_WRAPPER_ERROR;
    }

    ns_lock_handle = handle;
    return OS_WRAPPER_SUCCESS;
}
```

Problem statement

Currently TF-M requires users to provide implementation of *tfm_ns_interface_dispatch()*.

The problem here is that each user have to implement this function, which means that users need to read documentation, write, test and maintain the code. This is not ideal user experience.

Default (example) implementation of this function should be suitable for most of the use cases. So the proposal is to provide implementation of this function as part of NS interface *install* folder and have *os_wrapper_**() function be provided by NS OS.

This solution is same as one done for multicore mailbox, where NS OS is expected to provide implementation of *tfm_ns_mailbox_os_**() functions.

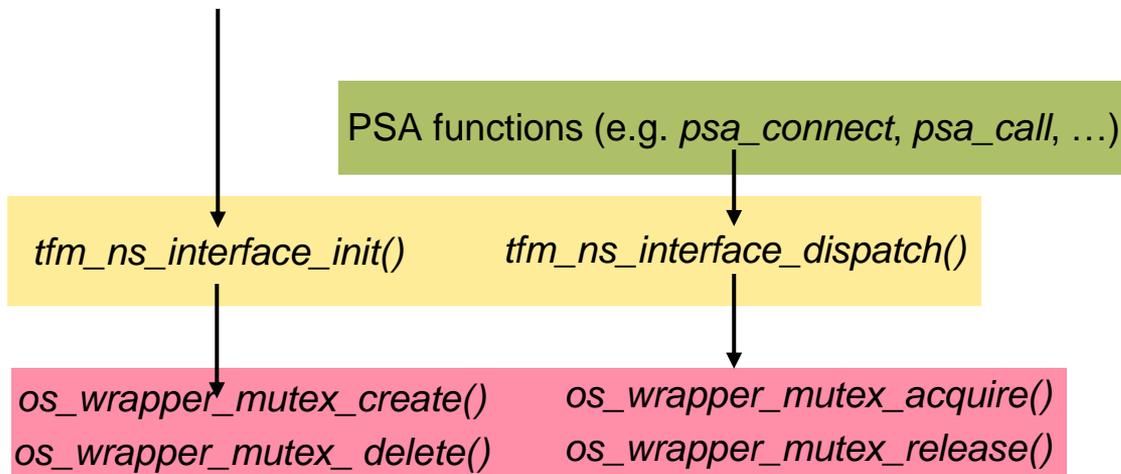
This way most of the users will have less code to write and maintain. Users that previously used custom implementations of this functions can still use them instead of the default one.

Firs step to this improvement was done by creating a patch to FreeRTOS to provide NS OS wrapper functions (see implementation [here](#)).

See next slide for more details.

Structure change proposal

- Provided by TFM
- Provided by NS OS
- Default implementation is provided by TFM, but can be replaced with user custom implementation



Proposed supported use cases (NS OSs)

Following use cases (NS OSs) are proposed to be supported by default:

- Free RToS
- CMSIS RToS v2
- Bare metal environment
- Zephyr (If the community is interested)

This list covers lots of the use cases and is suitable for lots of users.



Part of your life. Part of tomorrow.