Entry and exit of secure world

Introduction

Depending on configuration of the system secure world can be entered during different conditions. This document will describe only the configuration proposed here.

Monitor vector is VBAR_EL3 in AArch64 and MVBAR in ARMv7/AArch32.
State vector is any of:

- VBAR_EL2 or VBAR_EL1 (secure and non-secure) for AArch64, depending on configuration of hypervisor
- HVBAR or VBAR (secure and non-secure) for ARMv7, depending on configuration of hypervisor

The processor is configured to use:

- Monitor vector for FIQ exceptions while SCR_NS is set and state vector when SCR_NS is cleared
- Monitor vector for SMC exceptions
- State vector for IRQ exceptions

Interrupts handled by secure world are sent as FIQs and interrupts handled by normal world are sent as IRQs.

Since IRQs are received using the state vector the actual vector used depends on the current state of the CPU. If the NS (non-secure) bit in SCR (Secure Control Register) is set then either HVBAR or VBAR (non-secure) is used when receiving the IRQ, if the NS bit in SCR is cleared the secure VBAR is used instead. This has the consequence that secure world can receive IRQ that are supposed to be handled by normal world. When secure world receives an IRQ it has to be forwarded to normal world for processing.

The monitor

The monitor manages all entry and exit of secure world. To enter secure world from normal world the monitor saves the state of normal world (general purpose registers and system registers which are not banked) and restores the previous state of secure world. Then a return from exception is performed and the restored secure state is resumed. Exit from secure world to normal world is the reverse.

Some general purpose registers are not saved and restored on entry and exit, those are used to pass parameters between secure and normal world (see ARM_DEN0028A_SMC_Calling_Convention for details).

Entry and exit of Trusted OS

On entry and exit of Trusted OS each CPU is uses a separate entry stack and runs with IRQ and FIQ blocked.

During the entry phase checks are performed to see that the CPU may execute in secure world (could have restriction that only one CPU may execute in secure world, etc) and which context to start/resume execution in. Only when a context has been restored may IRQ and FIQ be unblocked.

On exit IRQ and FIQ are blocked, the context is saved and the entry stack is used again.

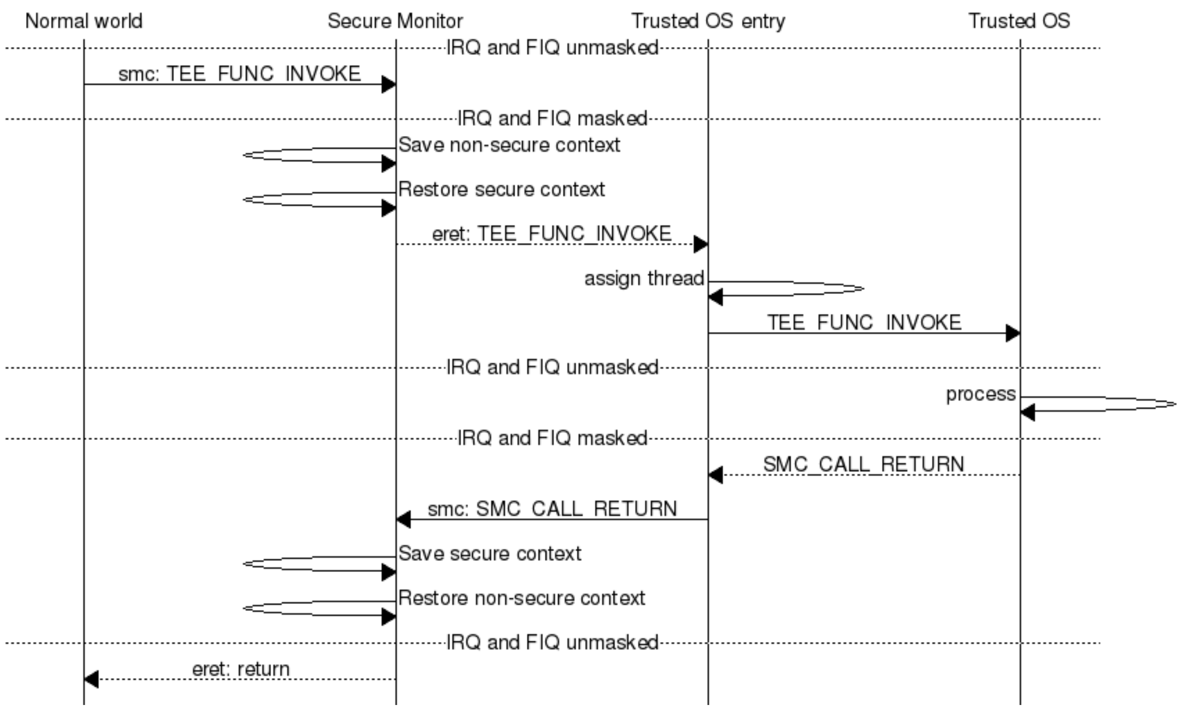

Fig: SMC entry of secure world

Forward IRQ from secure world to normal world

When an IRQ is received in secure world as an IRQ exception then secure world:

1. saves thread context (entire state of all processor modes for ARMv7)
2. blocks FIQ (IRQ is already blocked).
3. switches to entry stack
4. issues an SMC with a value to indicates to normal world that an IRQ has been delivered and last SMC call should be continued

The monitor restores normal world context with a return code indicating that an IRQ is about to be delivered. Normal world issues a new SMC indicating that it should continue last SMC.

The monitor restores secure world context which locates the previously saved context and checks that it's a return from IRQ that is requested before restoring the context and lets the secure world IRQ handler return from exception where the execution would be resumed.

Note that the monitor itself does not know/care that it has just forwarded an IRQ to normal world. The bookkeeping is done in the thread handling in Trusted OS. Normal world is responsible to decide when the secure world thread should resume execution. If secure world really need to execute something at a specific time it has to do that in FIQ context.

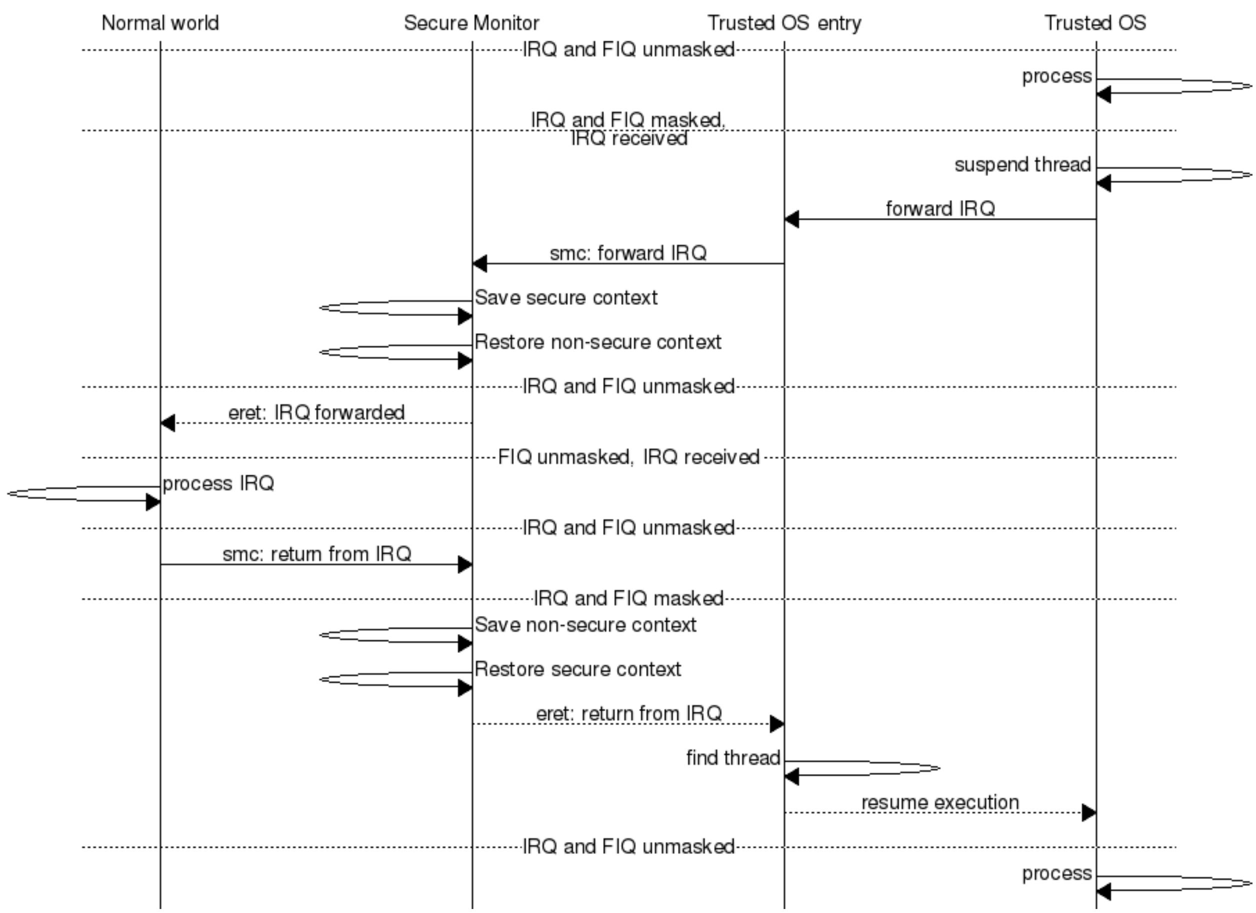

Fig: IRQ received in secure world and forwarded to normal world

Deliver FIQ to secure world

A FIQ can be received during two different states, either in non-secure world (SCR_NS is set) or in secure world (SCR_NS is cleared). When the secure monitor is active (ARMv8 EL3 or ARMv7 Monitor mode) FIQ is masked. FIQ reception in the two different states is described below.

Deliver FIQ to secure world when SCR_NS is set

When the monitor gets an FIQ exception it:

1. saves normal world context and restores secure world context from last secure world exit (which will have IRQ and FIQ blocked)
2. Clears SCR_FIQ when clearing SCR_NS
3. sets “FIQ” as parameter to secure world entry
4. does a return from exception into secure context
5. secure world unmasks FIQs because of the “FIQ” parameter
6. FIQ is received as in exception using the state vector
7. secure world issues an SMC to return to normal world
8. monitor saves secure world context and restores normal world context
9. does a return from exception into restored context

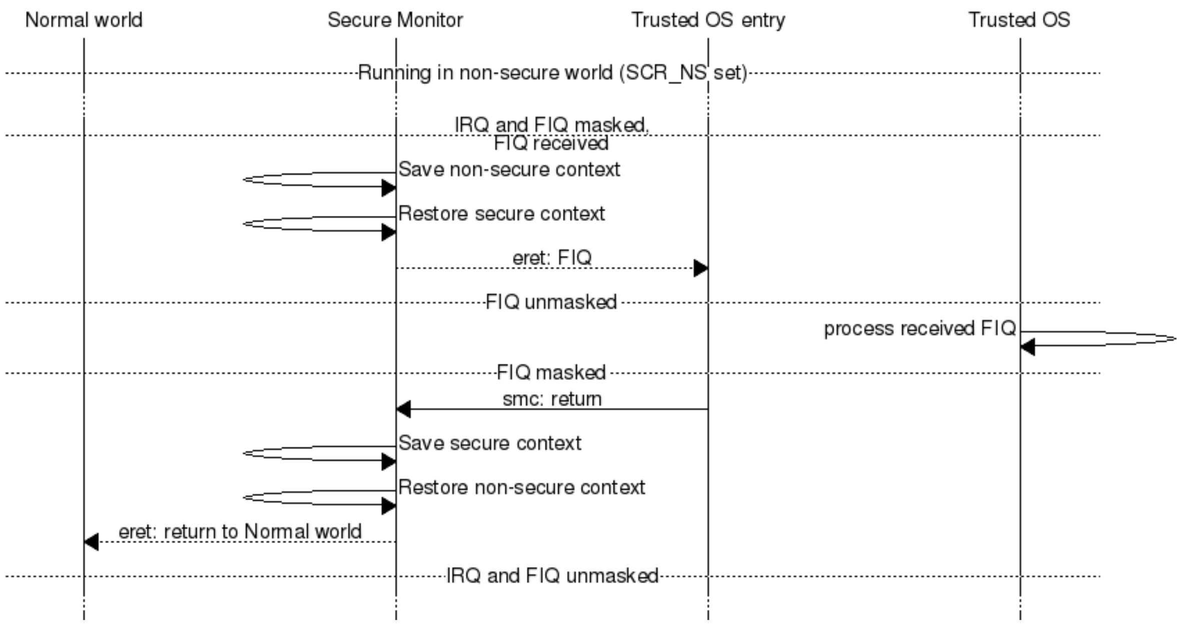

Fig: FIQ received when SCR_NS is set

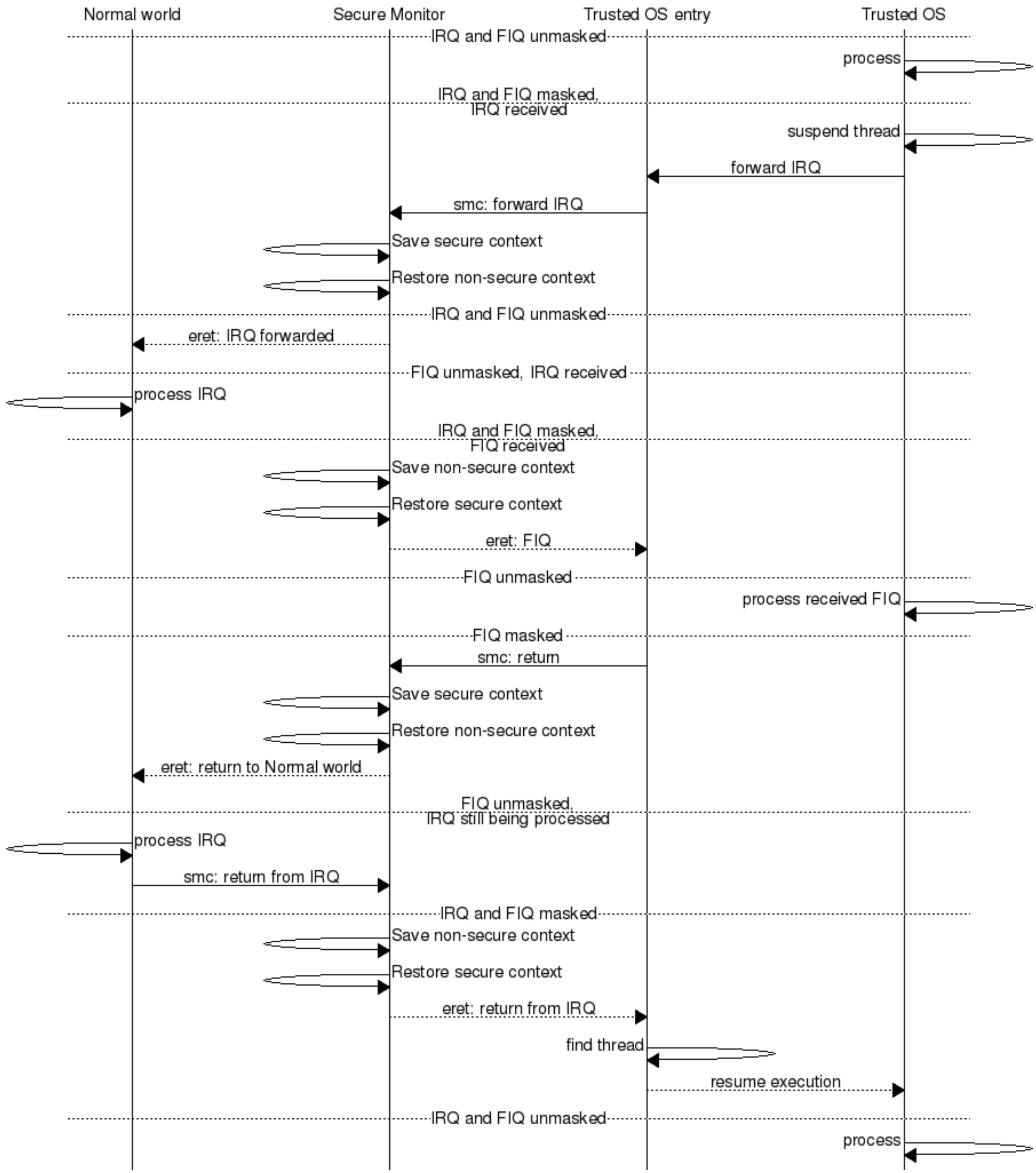

Fig: FIQ received while processing an IRQ forwarded from secure world

Deliver FIQ to secure world when SCR_NS is cleared

Since SCR_FIQ is cleared when SCR_NS is cleared a FIQ will be delivered using the state vector (VBAR) in secure world. The FIQ is received as any other exception by Trusted OS, the monitor is not involved at all.