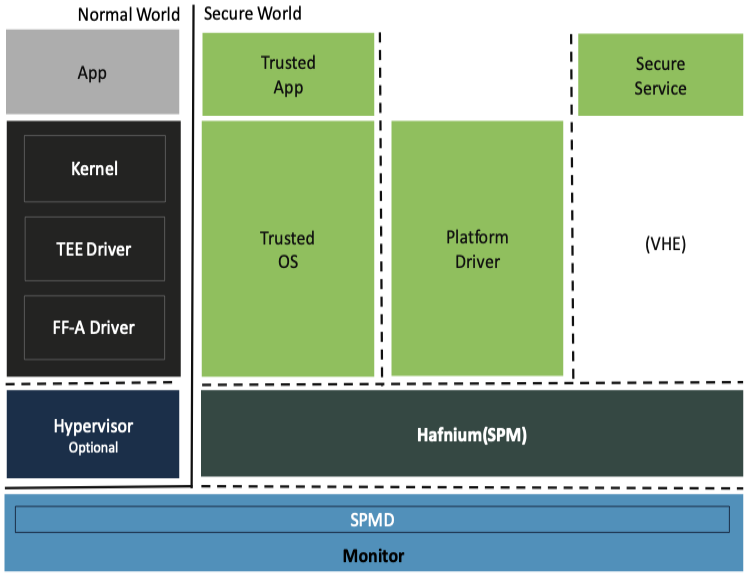
Inside Hafnium: Open source Secure Partition Manager for Arm-A profile architecture

Agenda

- Why Secure Partition Manager?
- Introduction to Hafnium
- New features after 2021
    - Improved secure interrupt handling
    - Improved FF-A Memory management
    - IPI support
    - And many more …
- Future roadmap
    - SP lifecycle management and Live activation support
    - NUMA aware resource allocation
- Resources

Why Secure Partition Manager?

- A Secure Partition (SP) refers to a virtual machine deployed in the secure world.
- A Secure Partition Manager (SPM) is responsible for managing one or more SPs.
- The SPM is typically split into two distinct components residing at same or different exception levels:
    - SPMD: Dispatcher component
    - SPMC: Core component
- SPMD is always located at EL3 and is mainly responsible for managing world switch events.



Introduction to Hafnium

- Hafnium is a reference SPMC implementation following the FF-A (Firmware Framework for Arm A-profile architecture) specification.
- An open source project hosted under trustedfirmware.org
- Leverages Arm's virtualization extensions, specifically FEAT_SEL2 introduced in Armv8.4, to allow multiple SPs to run concurrently on a A-profile SoC.
- Each SP provides a Trusted Execution Environment to run a Trusted OS and/or a Trusted Application.
- An SP can either be a MP SP with pinned execution contexts or a UP SP.
- SPMC acts as a Hypervisor for SPs:
    - Limits memory use and handles all system calls from SP.
    - Enforces spatial isolation and protects critical resources shared among SPs.
    - Facilitates communication between SPs and normal world endpoints.

New features

- Several features were added and numerous improvements to existing features since the last update in Linaro Connect 2021
- Support for S-EL0 secure partitions by leveraging VHE extensions.
- Support for FF-A boot protocol
- Support for SP's execution context to send Inter-Processor interrupts.
- Improved FF-A setup and discovery:
    - Permits discovery of features and properties of endpoints deployed in a system
- Improved FF-A Notifications:
    - Support for ABIs to permit VM to SP and SP to SP asynchronous signaling; no waiting
    - Has non-blocking semantics; CPU cycle allocation done by primary scheduler
    - Support for SRI (Scheduler receiver Interrupt) and NPI (notification pending interrupt)

New features continued ...

- Secure interrupt handling:
  - Support signaling virtual secure interrupt to an execution context of an SP
  - Target of the secure interrupt could be either S-EL0 or S-EL1 partition
  - Secure interrupt could trigger either when execution is in Secure or normal world
  - SPs can enable, disable and re-route interrupts among their execution contexts.
  - Enable Group0 (EL3) interrupt delegation to Secure Monitor (TF-A BL31).
- Advanced SIMD:
  - SPs are permitted to use FPU/SIMD extensions but not SVE and SME
  - Hafnium saves/restores the incoming SVE, SME (and FPU/SIMD) state across world switches.
- Support for MTE stack tagging
- Support for producing HOB structure and passing HOB list to SP (such as StMM)

New features continued ...

- Improved support for FF-A v1.2 memory management protocol:
  - Different endpoints (in secure as well as normal world) share, lend or donate memory in a structured way using transaction descriptors
  - Ability to specify access controls, memory attributes and permissions for each transaction
  - Partition managers maintain global handles for each memory operation
  - Support for transmission of fragmented descriptors
  - Public interfaces: FFA_MEM_SHARE, FFA_MEM_DONATE, FFA_MEM_LEND, FFA_MEM_RECLAIM, FFA_MEM_RETRIEVE_REQ, FFA_MEM_RELINQUISH etc.
  - Example: A Secure Partition needs to temporarily share a buffer with a VM and another SP

New features continued ...

- Support for FF-A v1.1 Indirect messaging interface
- Secondary CPU cold boot protocol:
    - Once the system is started and normal world is brought up, a secondary physical core is woken up by the ``PSCI_CPU_ON`` service invocation.
    - Each pinned execution context (EC) of every MP SP is woken up by SPMC, thereby giving an opportunity to the MP SP's EC on secondary core to initialize itself.
    - If a system only has UP SPs, then there are no pinned execution contexts to be resumed on secondary cores.
- Support for broadcasting CPU_OFF PSCI event to SPs through framework message provided:
    - The SP has subscribed to the CPU_OFF operation explicitly through its partition manifest.
    - The pinned execution context of the SP on the current CPU is in the WAITING state.

New features continued ...

- Emulate architectural timer and support system counter for SPs
- Support for VM availability messages
- Static DMA isolation of upstream peripheral devices assigned to SPs
- Support for partition runtime models, CPU cycle allocation modes and call chains:
    - Helps to enforce legal state transitions for execution contexts of SPs
    - Ensures there are no cycles that could lead to deadlock between endpoints
- Miscellaneous:
    - Support for FF-A Console Log ABI
    - Support for EL3 Logical Secure Partitions managed by SPMD
    - Several other features for compliance with FF-A v1.2 specification
    - Improved threat model assessment

New features continued ...

- Numerous improvement to build and test environments
- Several exhaustive tests suits added for compliance testing.

Future Roadmap

- FF-A SP lifecycle management and Live activation
  - Based on guidance added in FF-A v1.3 spec
  - Spec currently in Alpha phase and collecting feedback from partners
  - Adds support for abort handling and to put the SP in a specific state
  - New lifecycle states added (such as CREATED, STARTING, STOPPED, STOPPING)
  - Ability to restart an SP and live activate an SP without the need for a reboot
  - Hafnium team started implementation in coordination with LFA agent development in TF-A
- NUMA aware resource allocation
- FW Handoff boot info from TF-A(BL31) to SPMC(BL32)
- Support for new features introduced in FF-A v1.3 specification
- And many more ...

Resources

- Documentation: https://hafnium.readthedocs.io/en/latest/
- Git repo: https://git.trustedfirmware.org/plugins/gitiles/hafnium/
- Gerrit dashboard: https://review.trustedfirmware.org/q/project:hafnium/hafnium
- CI jobs: https://ci.trustedfirmware.org/view/Hafnium/
- FF-A specification: https://developer.arm.com/documentation/den0077

Thank You!