Hafnium – VHE Support

Enabling S-EL0 Partitions on ARMv8.4+

Raghu K
Mayur G

Agenda

- Requirement
- FF-A S-EL0 Partitions as a solution
- Explore S-EL0 partitions solution space
- Proof-of-concept status
- Takeaways

Requirements

- Requirements (Driven by Data Center environments)
  - Minimize code in Secure World
    - Better security - lower attack surface
    - RAS, StMM, Secure Storage, TPM (not always) are typical use cases
    - No known use cases today for DRM, Global Platform API’s, RPMB etc
  - Minimize cycle stealing from Normal World
    - Extremely sensitive to jitter
    - No scheduler in Secure World
    - Secure Interrupt handling required, but steals cycles
    - Ideally, Normal World voluntarily provides secure world cycles
  - Upstream with long term support
  - Standards based solution only (FF-A)
  - Portable between Pre-ARMv8.4 and ARMv8.4+ Platforms (Re-usable solutions)

FF-A S-EL0 Partitions

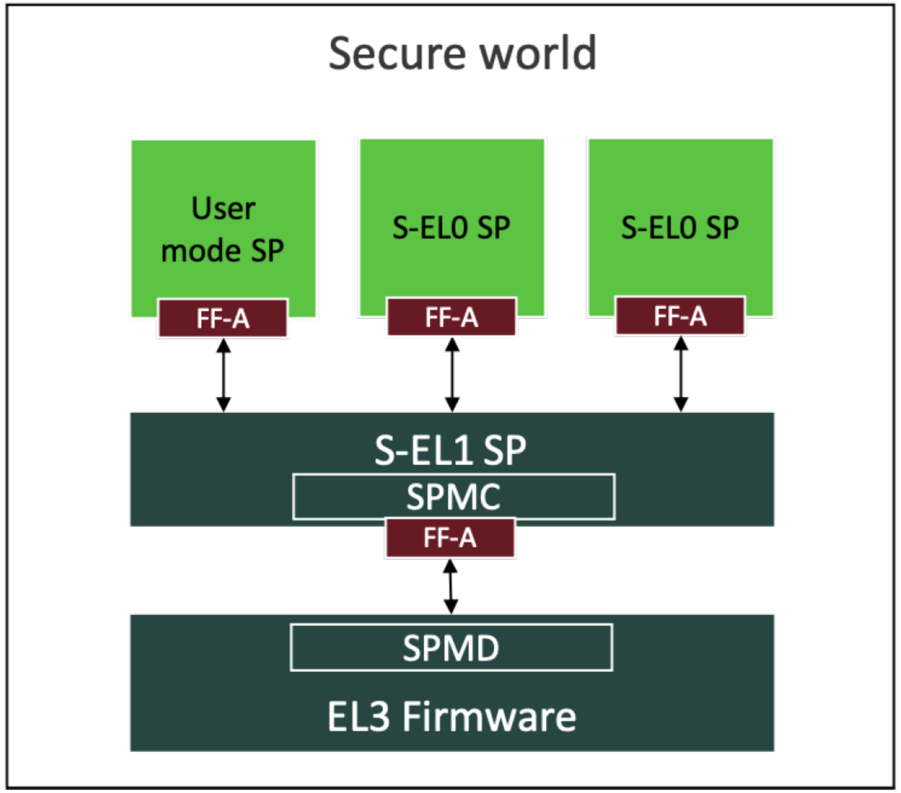
- FF-A S-EL0 Partitions is simplest tool sufficient to meet requirements
  - Avoids *fully featured* Trusted OS'es – less code ~= less jitter, less code ~= better security
  - Most Secure World code isolated in lowest privilege level – Better Security
  - Simple Interrupt Handling Models in EL0
  - Re-usable Pre-ARMv8.4 and Post-ARMv8.4 (EL0 only code)

S-EL0 Partitions Solution Space

- Trusted OS only solution (No S-EL2)
- SPMC + SPMD in EL3
- Hafnium + Trusted OSs
- Hafnium + S-EL1 Shim + S-EL0 partition
- Hafnium + VHE

Trusted OS (No S-EL2)

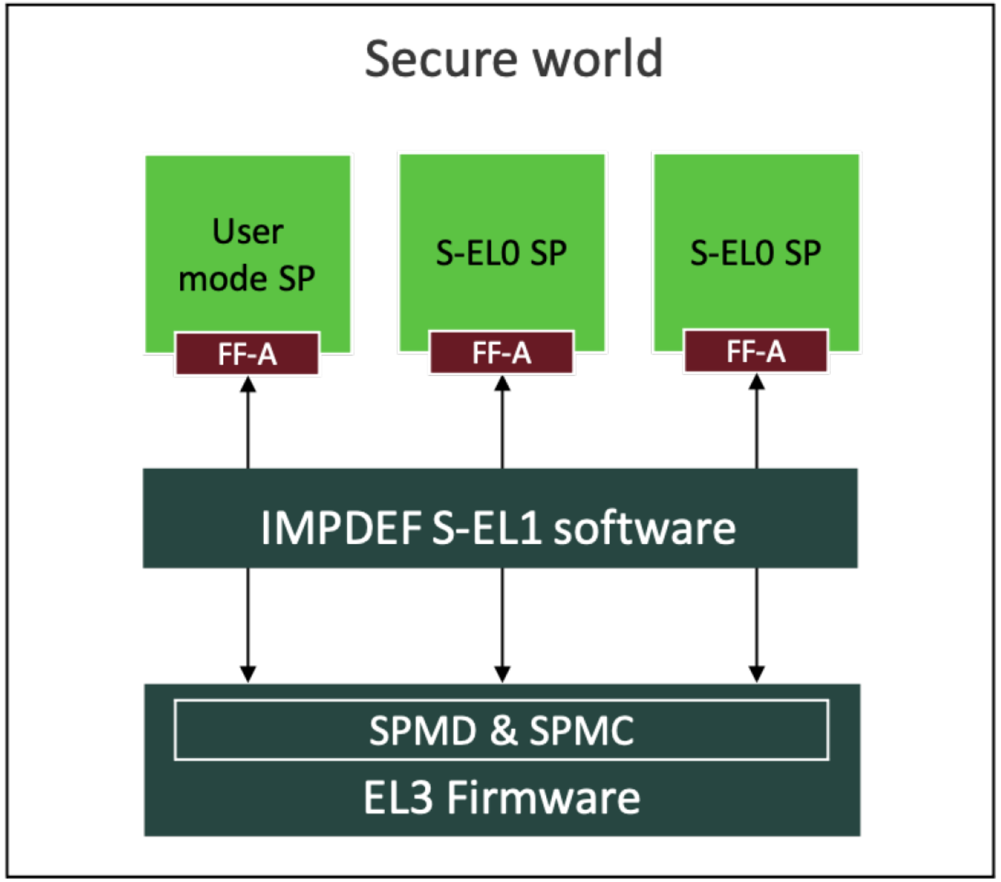
- No need for a traditional Trusted OS
- Large(er) attack surface
  - No known use case for DRM, Global platform API's etc.
- FF-A support limited and retrofitted
- Don't want to be tied to a Trusted OS
  - Not (entirely) ARM standard's based
  - (May) Require Trusted OS specific drivers
- Designed with mobile devices in mind
  - Does it scale to 100's of cores?
  - Can we influence design?
  - Can we make it work on a highly configurable system without recompile?



Source: FF-A Spec

SPMC + SPMD in EL3

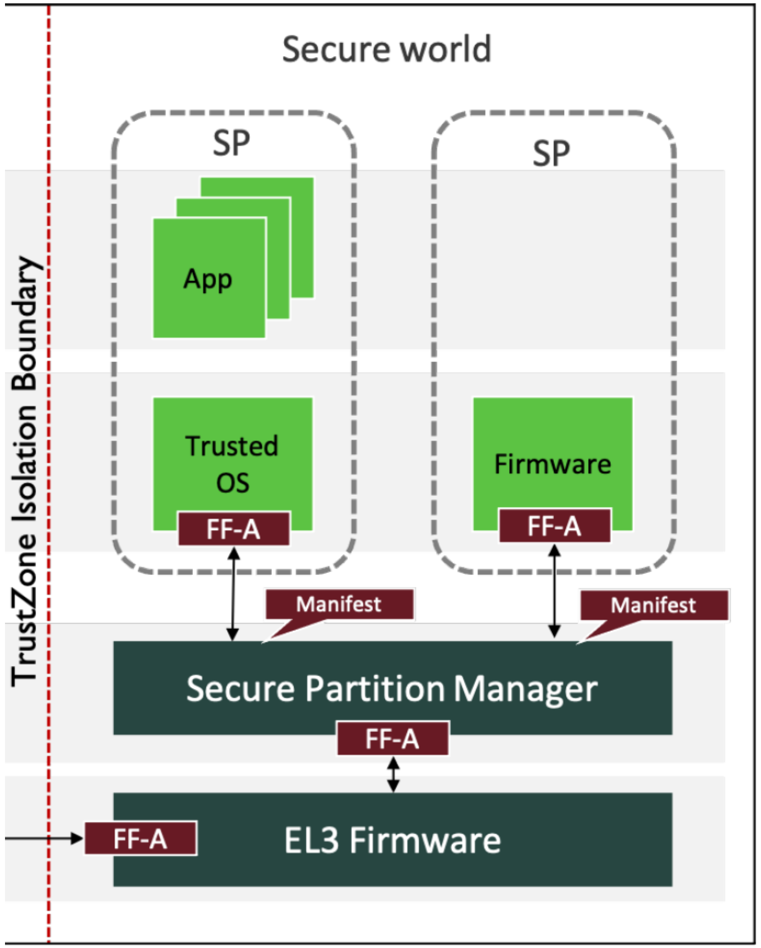
- Theoretically, this would be ideal solution
  - Assumption - We will put bare bones SPMD & SPMC required for SP's to work
- However:
  - Not ARM's main enablement model
    - No plan to support multiple partitions in this model
    - Support for StMM only
  - Not ideal considering ARM CCA



Source: FF-A Spec

Hafnium + Trusted OSs

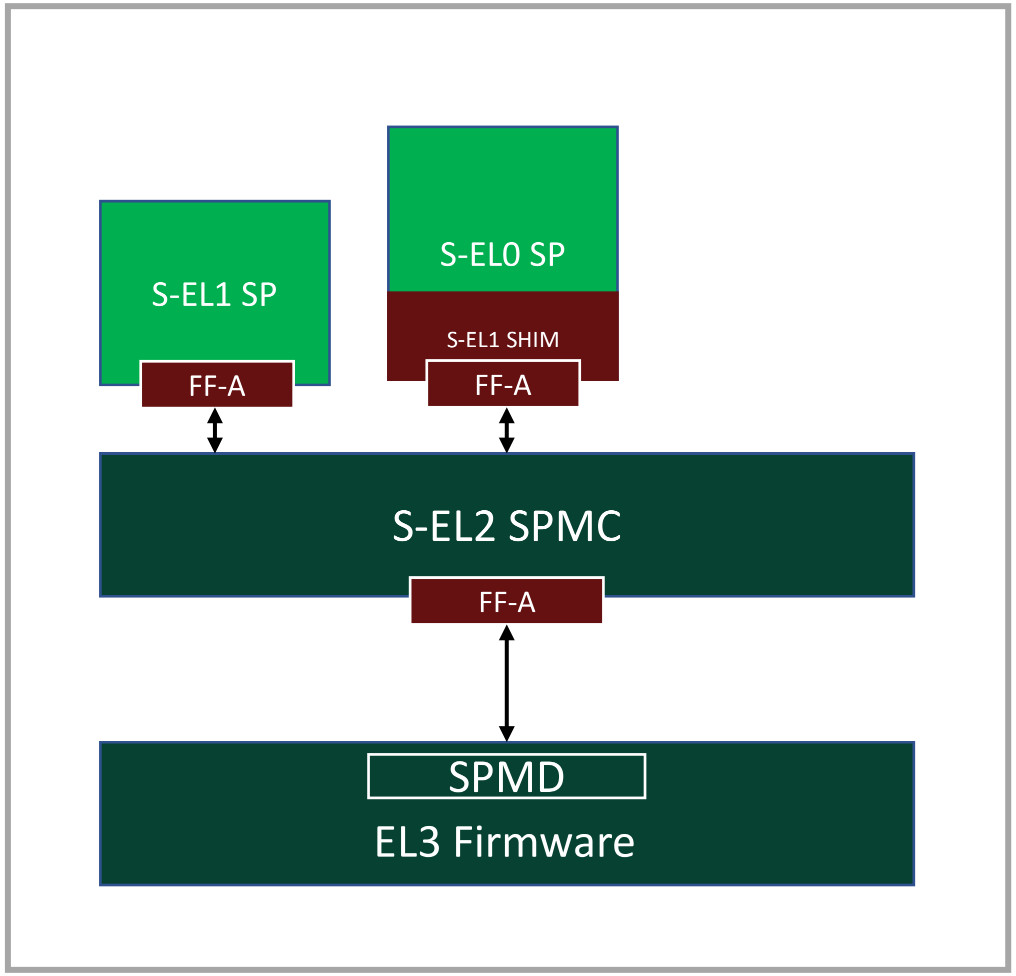
- All reasons for not using Trusted OSs
- Running firmware in S-EL1 is not portable between Pre-ARMv8.4 and Post-ARMv8.4 platforms
- Don’t really need Virtualization for currently know use cases
    - Don’t need to run multiple Trusted OSs
    - Avoid virtualization over head
        - More expensive translation table walks (2-stages, 16 memory accesses on a TLB miss)
        - Large context to be switched (EL0 + EL1 registers)
        - Lower jitter from secure world code



Source: FF-A Spec

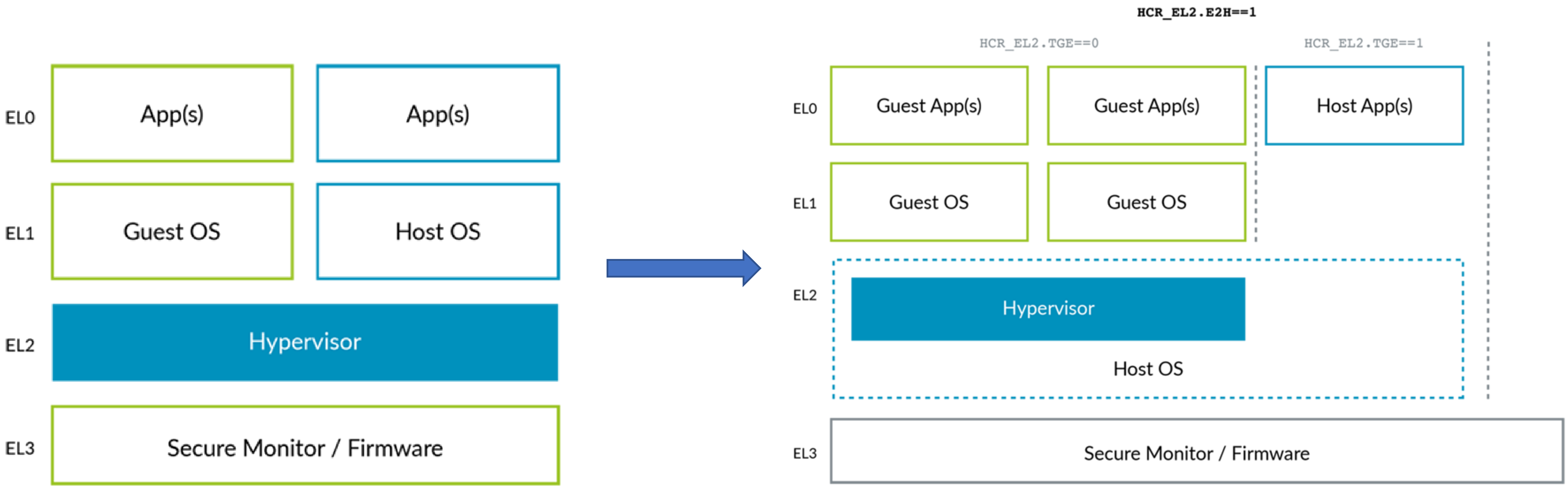
Hafnium + S-EL1 Shim + S-EL0 Partition

- Virtualization overheads described previously
- S-EL1 partition treated as S-EL0 partition for interrupt handling, scheduling models etc. - not ideal
- Otherwise acceptable solution architecturally
  - SPMC only needs to support S-EL1 partitions
- However, will implementation be clean?
  - Should Shim be part of hafnium or S-EL0 partition?
    - Hafnium – Need code to recognize such partitions and support it – cannot treat as vanilla S-EL1 partition
    - S-EL0 – Needs to be built differently for Pre-v8.4 or Post-v8.4
  - Who handles FF-A memory management transactions?
    - S-EL1 shim – Shim bloat
    - S-EL0 – Need to ask shim to map/unmap memory in stage-1, S-EL0 now aware of existence of S-EL1 shim.



What is VHE (Virtualization Host Extensions)?

- Supports running unmodified OSs in EL2, without using EL1
- Better virtualization performance



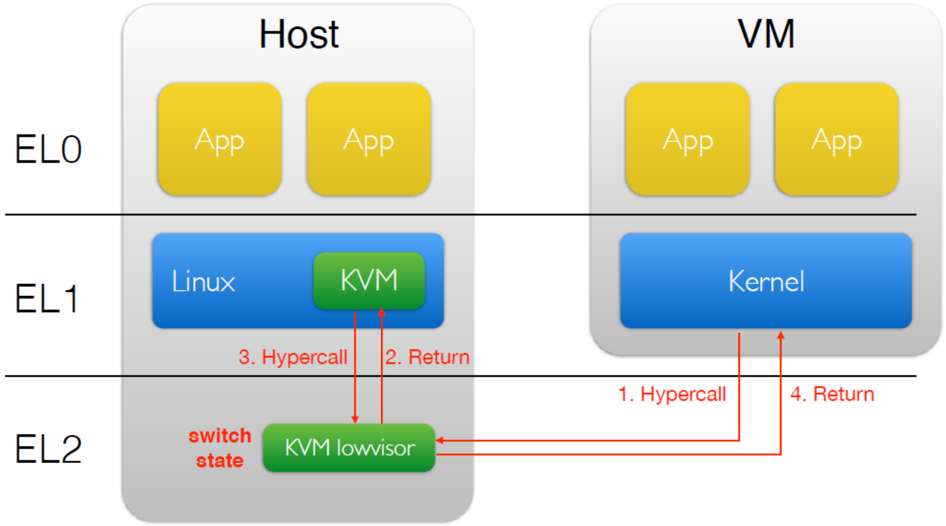
EL0
App(s)
App(s)
EL1
Guest OS
Host OS
EL2
Hypervisor
EL3
Secure Monitor / Firmware

HCR_EL2.E2H==1
HCR_EL2.TGE==0
HCR_EL2.TGE==1
EL0
Guest App(s)
Guest App(s)
Host App(s)
EL1
Guest OS
Guest OS
EL2
Hypervisor
Host OS
EL3
Secure Monitor / Firmware

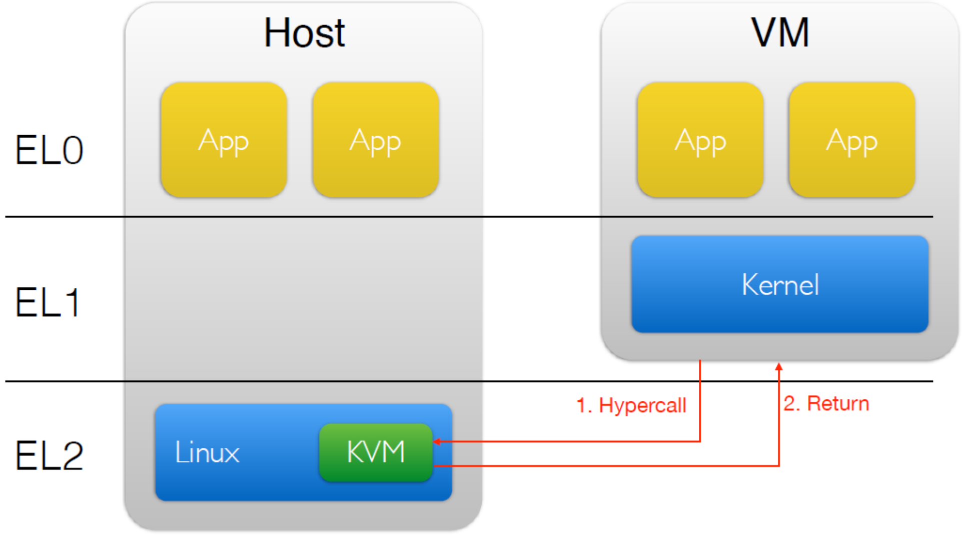
Source: Learn the architecture - VHE

Linux/KVM - VHE

KVM/ARM Split-Mode



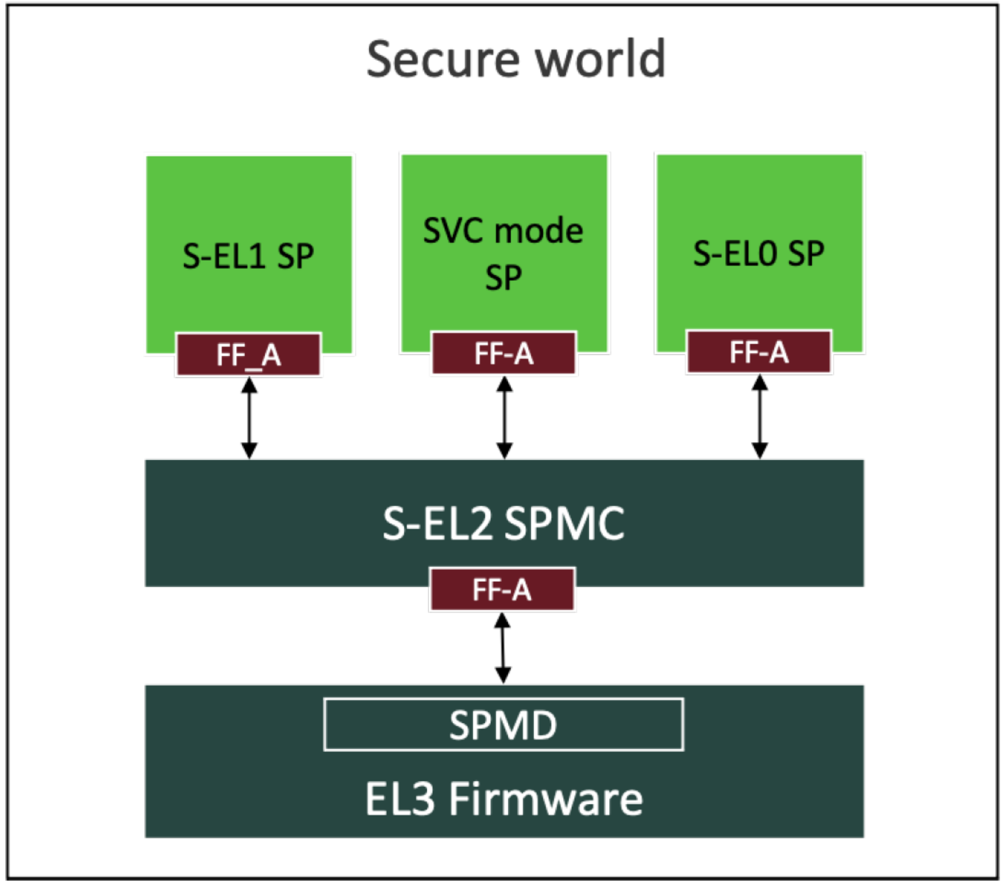
What if we could do this?



Source: To EL2 and Beyond

Hafnium + VHE

- Explicit support in FF-A spec for VHE
- Built with FF-A in mind (mostly)
- Better model considering ARMv9 changes
- Avoid virtualization overhead
- No legacy (not much legacy)
  - Lower attack surface
  - Fresh start – Ability to influence scalability issues for large systems from ground up
- Flexible - Can use both S-EL0 and S-EL1 SP if needed
- How is this different than a Trusted OS?
  - It is not – VHE turns hafnium into an FF-A only Trusted OS with nothing else!
  - It is also a hypervisor, if/when needed.



Source: FF-A Spec

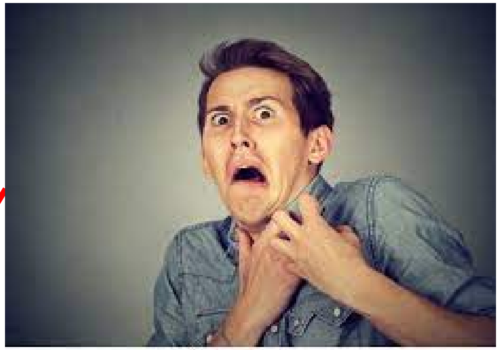
Hafnium + VHE – The Bad

- Maintenance and support for S-EL0 partitions and tests.
- Lower interrupt handling efficiency relative to S-EL1 partitions
  - Due to supported interrupt models by FF-A (by design)
- Even with S-EL0 support, not ideal situation code wise
  - Increased code size - Initial support will likely have both S-EL1 and S-EL0 support even though we may not need S-EL1
  - Hope to get to a world where hafnium can be compiled with support for only S-EL0 partitions

Solution vs Requirements

	Minimize Secure World Code	Minimize Cycle Stealing, Jitter	Upstream + LTS	Standard Based (FF-A)	S-EL0/1 FW Portable between Pre v8.4 and Post v8.4	Practical Issues
Trusted OS (No-SEL2)	• Larger attack surface relative to a S-EL0 solutions • No known use case for fully featured Trusted OS	• Meets/Can meet requirements	• Meets/Can meet requirements	• FF-A support limited and retrofitted currently	• Meets/Can meet requirements	• TOS Designed with mobile devices in mind. • Lots of legacy and potentially more effort to make it scale to servers.
SPMD + SPMC in EL3	• Meets/Can meet requirements	• Meets/Can meet requirements	• Limited support expected (single partition, StMM only)	• Meets/Can meet requirements	• Meets/Can meet requirements	• Not a great solution considering ARMv9.
Hafnium + Trusted OSs	• Larger attack surface in S-EL1 relative to a S-EL0 solutions • No known use case for fully featured Trusted OS	• Virtualization over head – Larger context switches, penalty on TLB misses etc	• Meets/Can meet requirements	• Meets/Can meet requirements	• Running firmware in S-EL1 is not portable and binary compatible between Pre v8.4 and Post v8.4	• TOS designed with mobile devices in mind. • Lots of legacy and potentially more effort to make it scale to servers.
Hafnium + S-EL1 Shim + S-EL0 partition	• Meets/Can meet requirements	• Virtualization over head – Larger context switches, penalty on TLB misses etc	• Limited support expected currently	• Meets/Can meet requirements	• S-EL0 partitions not portable and binary compatible between Pre v8.4 and Post v8.4 platforms	• Possibility of ending up with heavy shim and higher maintenance overhead
Hafnium + VHE	• Meets/Can meet requirements	• Meets/Can meet requirements	• Meets/Can meet requirements(assuming patches merge)	• Meets/Can meet requirements	• Meets/Can meet requirements	• Maintenance/support required • Interrupt handling efficiency may be lower for S-EL0 partition vs S-EL1

POC – Status, Opens



shutterstock.com · 497606257

- ~40 patches – includes changes to hafnium and tests
- Testing
  - Tested on Qemu (EL0 partitions)
  - Tested on FVP (EL0 and S-EL0 partitions)
  - ~75 EL1 VM test cases ported to EL0 including memory management, messaging, interrupts etc.
  - Existing S-EL1 test infrastructure leveraged to run basic S-EL0 tests on FVP
- Commits labeled with "VHE" for easy revert, Feature under build flag
- Opens
  - EL0 partition mapped RWX, so disable WXN – Tooling issue, to be fixed soon
  - Context switch – Not lightweight yet, switches EL1 state
  - Secure Interrupt handling support
  - Test code duplication – clean up
  - Can we do a hafnium build with purely S-EL0 support? Reduce attack surface even further!
  - PSCI interactions?
  - New issues that come up…

Takeaways

- Call to action
  - Encourage other ARM vendors to use S-EL0 partitions, if you don’t require virtualization in Secure World
  - Review and Merge Patches
  - Support/Run Trusted Services as Hafnium S-EL0 partitions
- Thanks
  - NVIDIA – Mayur G
  - ARM – Achin G, Olivier D

INTENTIONAL BLANK SLIDE

Trusted OS (No S-EL2) – Backup

Optee Feature list

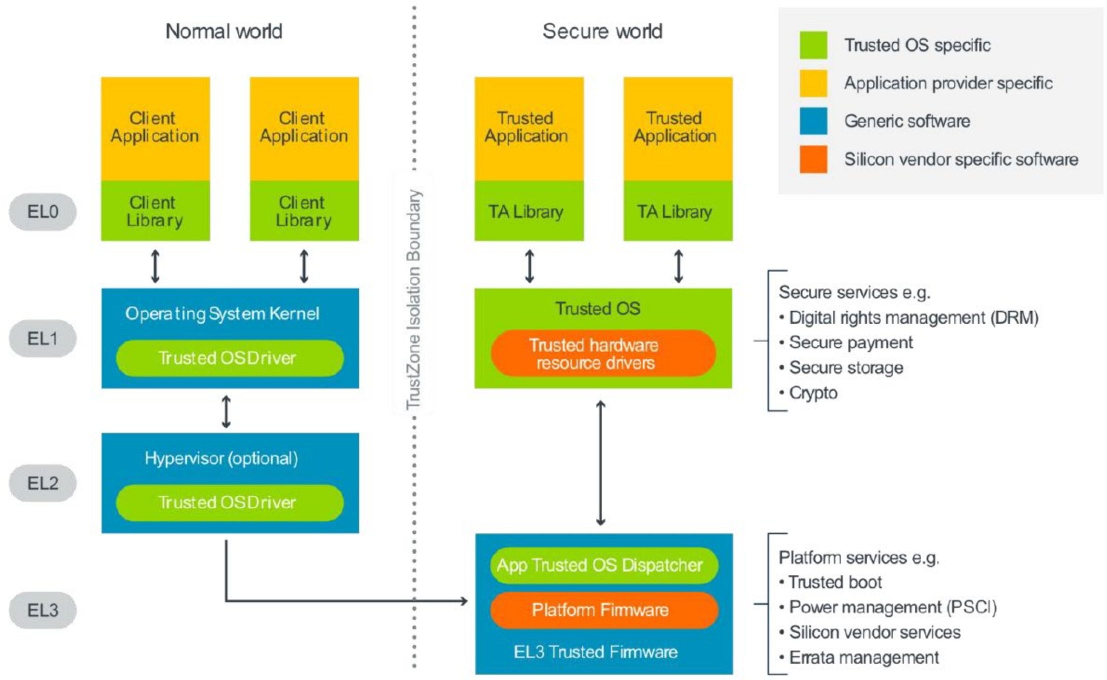
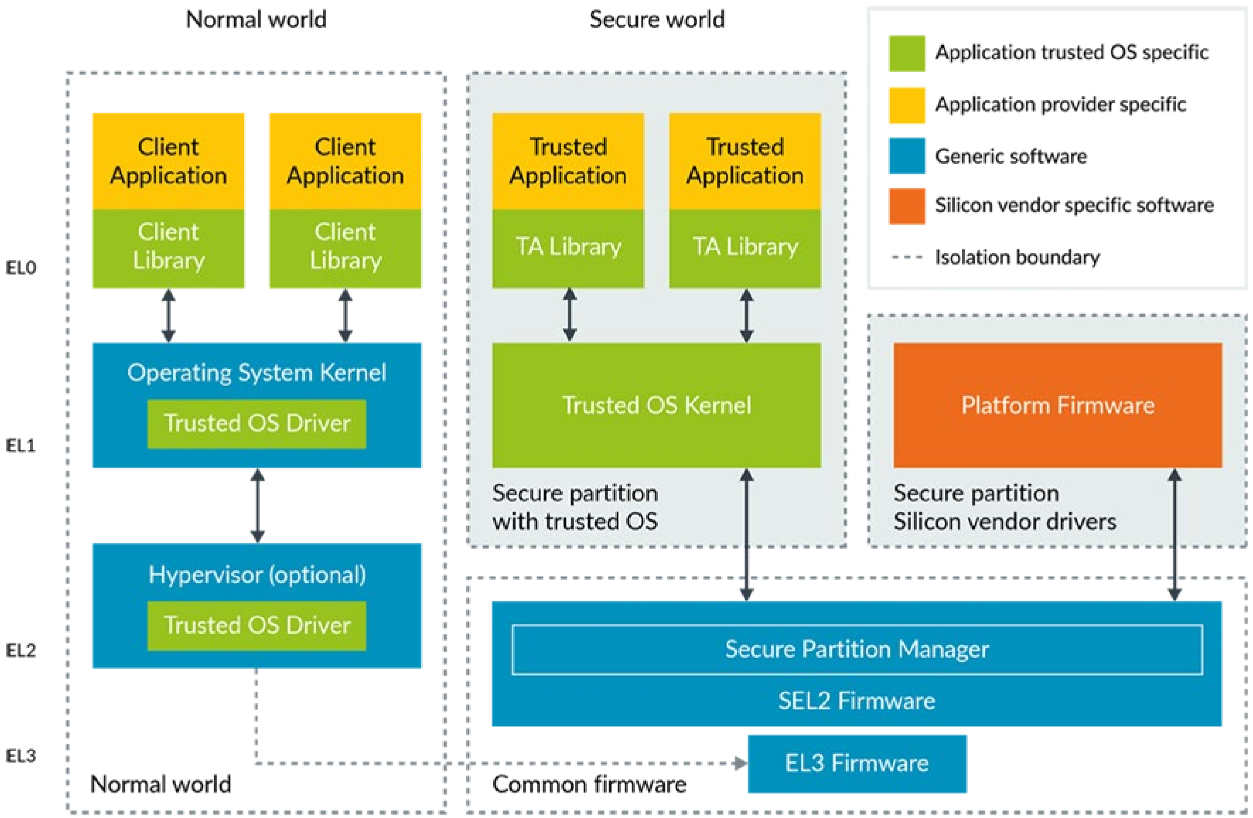


Figure 1 Typical services and partitioning of software agents in the Secure world

Source: Isolation using virtualization in the secure world

Hafnium + Secure OSs



Source: [Learn the architecture - Secure Virtualization]()