RISE

RISC-V Software Ecosystem

RISE Firmware - UEFI StandaloneMm Project
(Non-Hypervisor Platform)

Dec 06, 2023

Agenda

- Project Background and Overview
- EDK2 StandaloneMmPkg Porting status
- Context Support for OpenSBI Domain
- Inter-Domain SBI Messaging, RPMI Spec change
- Demo and Furture work

Background

- For general background on Management Mode (MM), as noted in the PI specification Volume 4: Management Mode Core Interface.

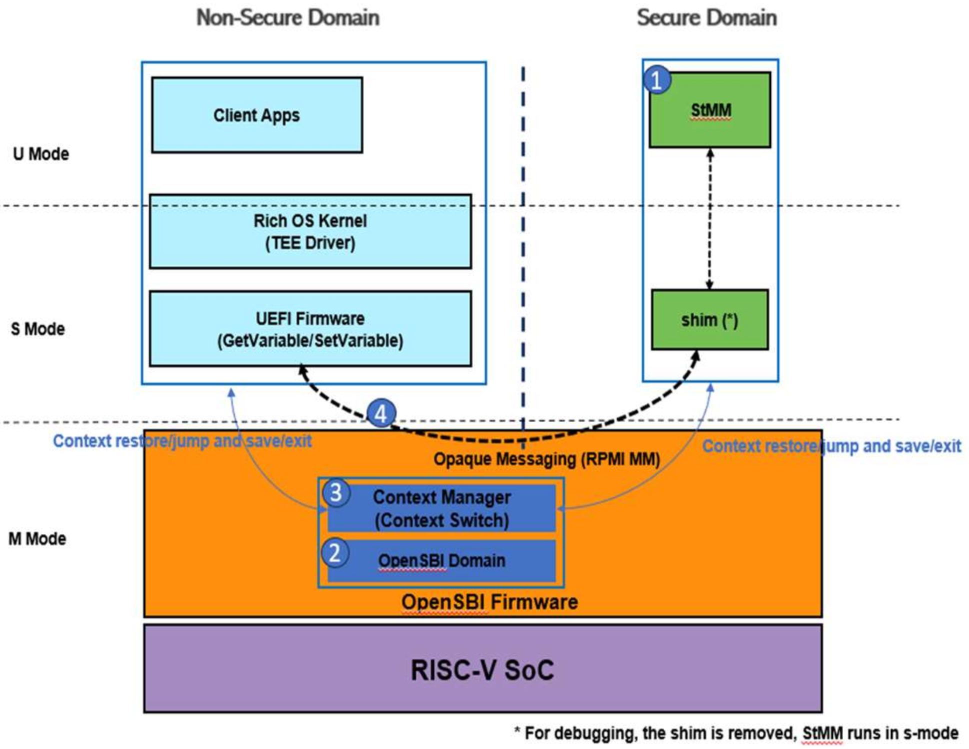
- MM is a generic term used to describe a secure execution environment provided by the CPU and related silicon that is entered when the CPU detects a Management Mode Interrupt (MMI).

- This Standalone MM project aims to port Tiano StandaloneMmPkg on RISC-V to support authenticated variable store and other MM scenarios.

https://uefi.org/specs/PI/1.8/index.html

Project Overview

- Port EDK2 StMM drive(StandaloneMmPkg) to RISC-V
- Use OpenSBI domain to isolate the underlying hardware (RAM and MMIO devices) and setup HARTs,
- Enhance the OpenSBI domain with context manager/switch feature.
- Use SBI RPXY interface and RPMI for inter-domain messaging



* For debugging, the shim is removed, StMM runs in s-mode

StandaloneMmPkg Proting status

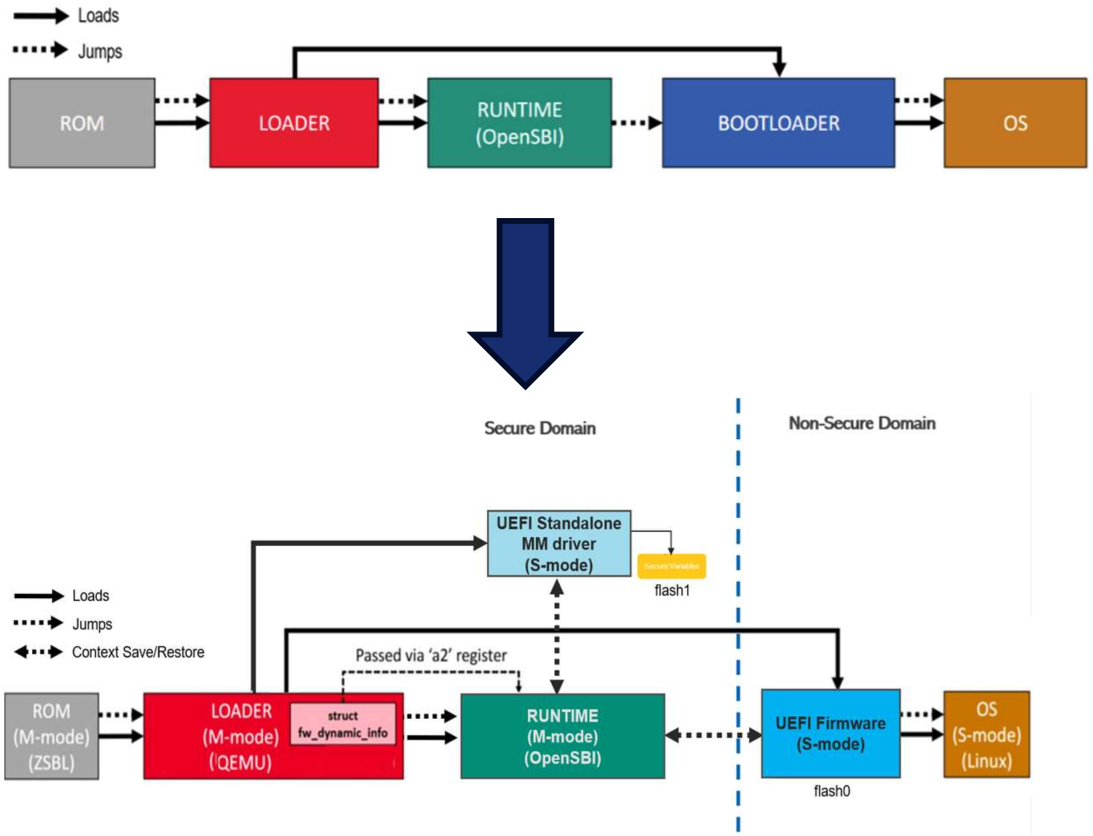
- Made StandaloneMmCpu platform independent
- Unified MM payload for ARM and RISC-V
- Unified MM entrypoint API for ARM and RISC-V
- Hob re-structure and MM entrypoint with HOB address
- Created riscv virt platform project files (currently in edk2 repo), may need to move the edk2-platforms

https://github.com/tianocore/edk2-staging/tree/RiscV64StandaloneMm

OpenSBI domain to isolate the environment

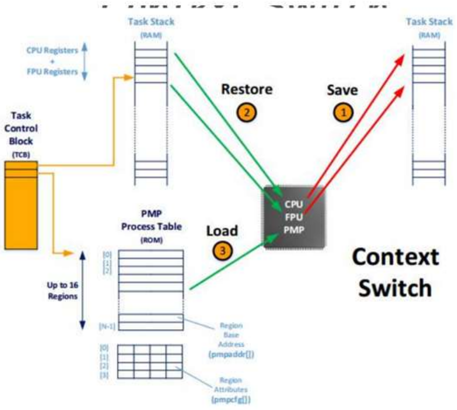
- U-boot SPL/QEMU is responsible to load both secure and non-secure domain firmware

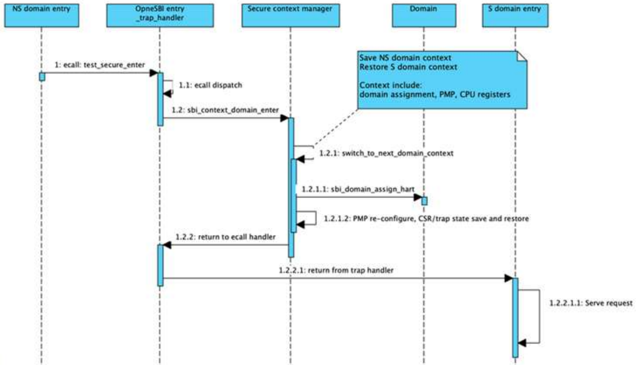
- Use OpenSBI domain to isolated the resource, use context switch between secure and non-secure domain to switch execution environment between secure and non-secure domain



Context manager for OpenSBI Domain (By Penglai)

- Context entry for each possible HART in a domain is saved
- Register and context save restore are achieved by change saved sbi_trap on the stack by _trap_handler() in fw_base.S
- The context manager api saves the context on current HART and switch to the context in new domain on the same HART
- The context enter/exit api can be triggered through a SBI ecall.




NS domain entry
OpneSBI entry _trap_handler
Secure context manager
Domain
S domain entry
1: ecall: test_secure_enter
1.1: ecall dispatch
1.2: sbi_context_domain_enter
Save NS domain context
Restore S domain context
Context include:
domain assignment, PMP, CPU registers
1.2.1: switch_to_next_domain_context
1.2.1.1: sbi_domain_assign_hart
1.2.1.2: PMP re-configure, CSR/trap state save and restore
1.2.2: return to ecall handler
1.2.2.1: return from trap handler
1.2.2.1.1: Serve request

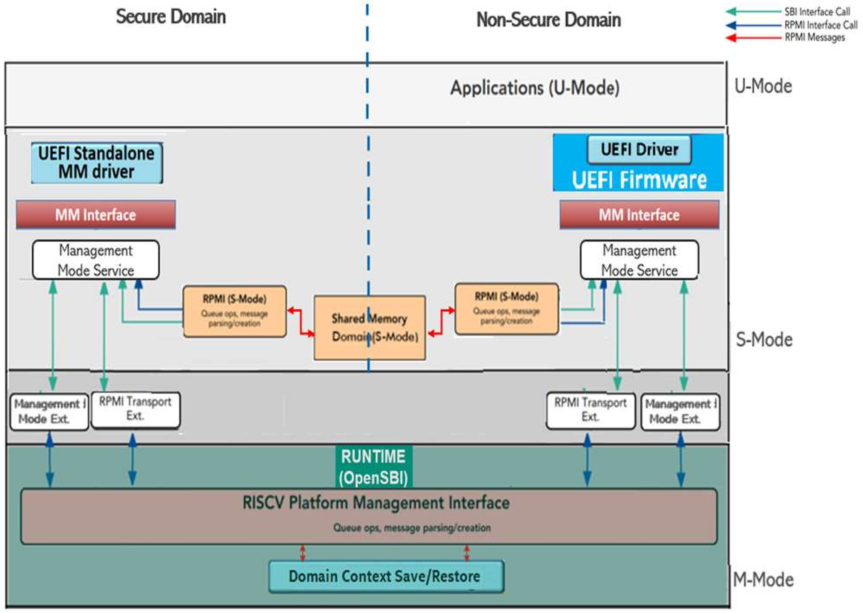
OpenSBI Patch: https://github.com/Penglai-Enclave/opensbi/commit/03ea2f3d7c900942bc903510c5ee9fc19008c3

Context Manager Doc: https://github.com/Penglai-Enclave/opensbi/blob/dev-context-management-v2.0/docs/context_manager.md

Test App: https://github.com/Shang-QY/test_context_switch

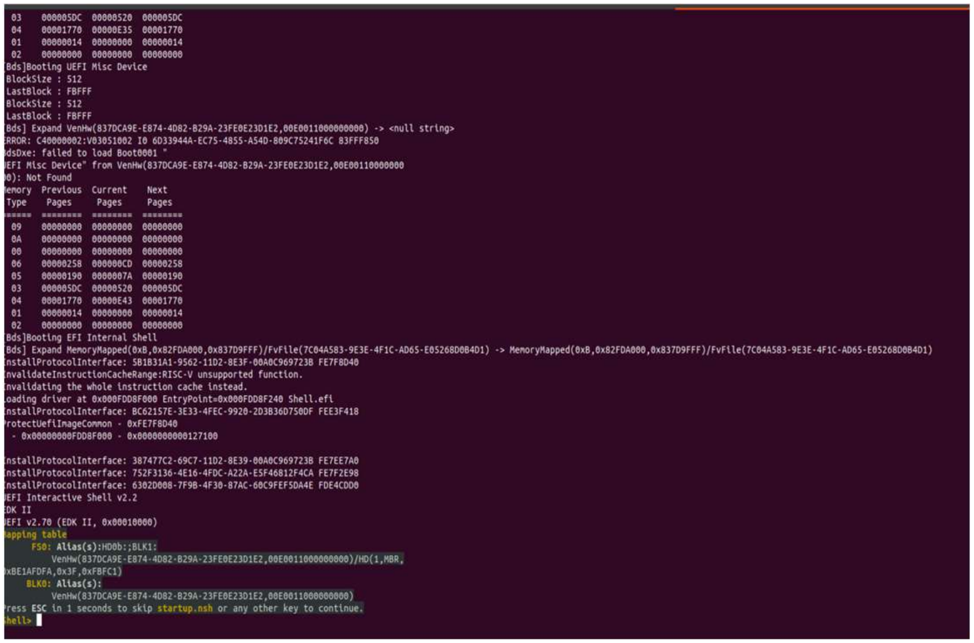
Inter-domain SBI messaging

- Leverage the SBI RPXY extension, which carries the RPMI message
- Put MM inter-doman messaging payload in shared memory
- Add an MM Service in RPMI with APIs:
  - MM_VERSION
  - MM_COMMUNICATE
  - MM_COMPLETE



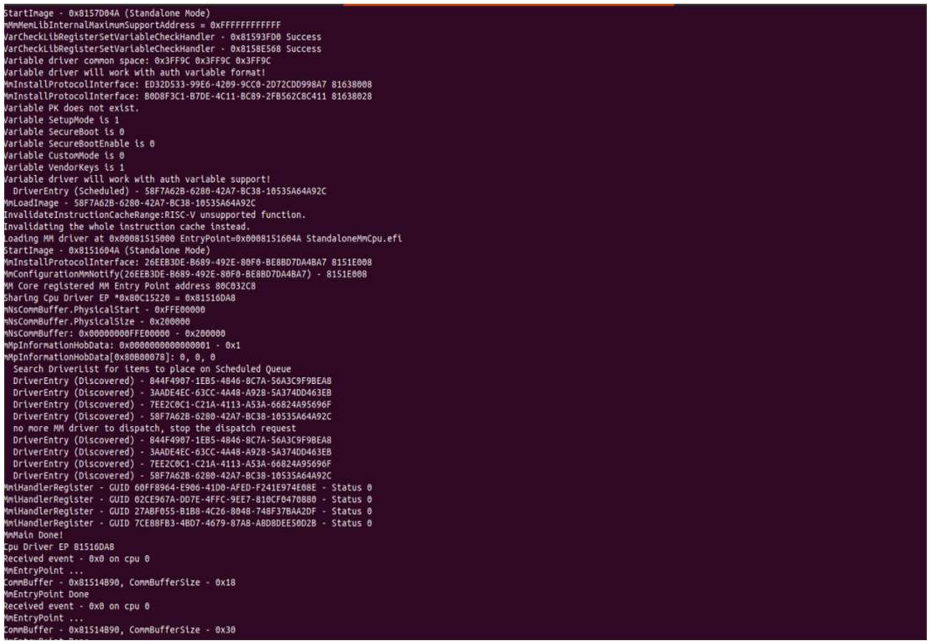
https://docs.google.com/document/d/199ar3Ddd-FIzP1FR3HOkbBf1BNvLUPvJ

Demo – Boot-up on QEMU Virt



```
 03    000005DC  00000520  000005DC
 04    00001770  00000E35  00001770
 01    00000014  00000000  00000014
 02    00000000  00000000  00000000
[Bds]Booting UEFI Misc Device
 BlockSize : 512
 LastBlock : FBFFF
 BlockSize : 512
 LastBlock : FBFFF
[Bds] Expand VenHw(837DCA9E-E874-4D82-B29A-23FE0E23D1E2,00E0011000000000) -> <null string>
RROR: C40000002:V03051002 I0 6D33944A-EC75-4855-A54D-809C75241F6C 83FFF850
dsDxe: failed to load Boot0001 "
EFI Misc Device" from VenHw(837DCA9E-E874-4D82-B29A-23FE0E23D1E2,00E00110000000
0): Not Found
emory  Previous  Current   Next
 Type   Pages     Pages    Pages
=====  ========  ========  ========
 09    00000000  00000000  00000000
 0A    00000000  00000000  00000000
 00    00000000  00000000  00000000
 06    00000258  000000CD  00000258
 05    00000190  0000007A  00000190
 03    000005DC  00000520  000005DC
 04    00001770  00000E43  00001770
 01    00000014  00000000  00000014
 02    00000000  00000000  00000000
[Bds]Booting EFI Internal Shell
[Bds] Expand MemoryMapped(0xB,0x82FDA000,0x837D9FFF)/FvFile(7C04A583-9E3E-4F1C-AD65-E05268D0B4D1) -> MemoryMapped(0xB,0x82FDA000,0x837D9FFF)/FvFile(7C04A583-9E3E-4F1C-AD65-E05268D0B4D1)
nstallProtocolInterface: 5B1B31A1-9562-11D2-8E3F-00A0C969723B FE7F8D40
nvalidateInstructionCacheRange:RISC-V unsupported function.
nvalidating the whole instruction cache instead.
oading driver at 0x000FDD8F000 EntryPoint=0x000FDD8F240 Shell.efi
nstallProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF FEE3F418
rotectUefiImageCommon - 0xFE7F8D40
 - 0x00000000FDD8F000 - 0x0000000000127100

nstallProtocolInterface: 387477C2-69C7-11D2-8E39-00A0C969723B FE7EE7A0
nstallProtocolInterface: 752F3136-4E16-4FDC-A22A-E5F46812F4CA FE7F2E98
nstallProtocolInterface: 6302D008-7F9B-4F30-87AC-60C9FEF5DA4E FDE4CDD0
EFI Interactive Shell v2.2
DK II
EFI v2.70 (EDK II, 0x00010000)
apping table
     FS0: Alias(s):HD0b:;BLK1:
          VenHw(837DCA9E-E874-4D82-B29A-23FE0E23D1E2,00E0011000000000)/HD(1,MBR,
xBE1AFDFA,0x3F,0xFBFC1)
    BLK0: Alias(s):
          VenHw(837DCA9E-E874-4D82-B29A-23FE0E23D1E2,00E0011000000000)
ress ESC in 1 seconds to skip startup.nsh or any other key to continue.
hell>
```

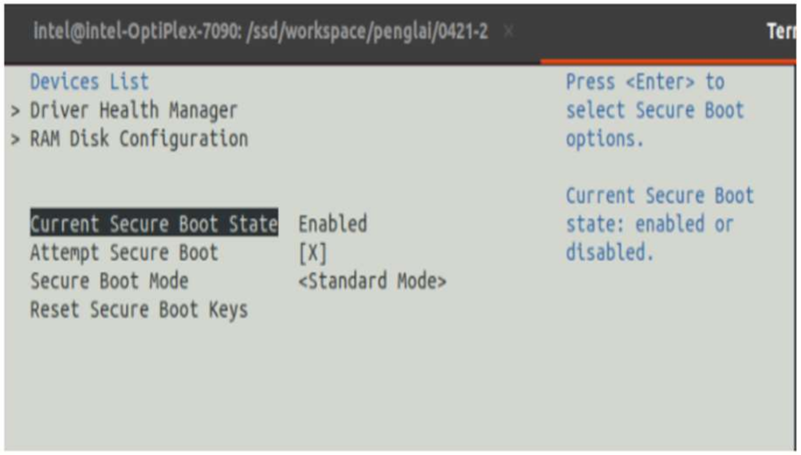
Non secure domain log – EDK2



```
StartImage - 0x8157D04A (Standalone Mode)
mMmMemLibInternalMaximumSupportAddress = 0xFFFFFFFFFFFF
VarCheckLibRegisterSetVariableCheckHandler - 0x81593FD0 Success
VarCheckLibRegisterSetVariableCheckHandler - 0x8158E568 Success
Variable driver common space: 0x3FF9C 0x3FF9C 0x3FF9C
Variable driver will work with auth variable format!
MmInstallProtocolInterface: ED32D533-99E6-4209-9CC0-2D72CDD998A7 81638008
MmInstallProtocolInterface: B0D8F3C1-B7DE-4C11-BC89-2FB562C8C411 81638028
Variable PK does not exist.
Variable SetupMode is 1
Variable SecureBoot is 0
Variable SecureBootEnable is 0
Variable CustomMode is 0
Variable VendorKeys is 1
Variable driver will work with auth variable support!
 DriverEntry (Scheduled) - 58F7A62B-6280-42A7-BC38-10535A64A92C
MmLoadImage - 58F7A62B-6280-42A7-BC38-10535A64A92C
InvalidateInstructionCacheRange:RISC-V unsupported function.
Invalidating the whole instruction cache instead.
Loading MM driver at 0x00081515000 EntryPoint=0x0008151604A StandaloneMmCpu.efi
StartImage - 0x8151604A (Standalone Mode)
MmInstallProtocolInterface: 26EEB3DE-B689-492E-80F0-BE8BD7DA4BA7 8151E008
MmConfigurationMmNotify(26EEB3DE-B689-492E-80F0-BE8BD7DA4BA7) - 8151E008
MM Core registered MM Entry Point address 80C032C8
Sharing Cpu Driver EP *0x80C15220 = 0x81516DA8
mNsCommBuffer.PhysicalStart - 0xFFE00000
mNsCommBuffer.PhysicalSize - 0x200000
mNsCommBuffer: 0x00000000FFE00000 - 0x200000
mMpInformationHobData: 0x0000000000000001 - 0x1
mMpInformationHobData[0x80B00078]: 0, 0, 0
 Search DriverList for items to place on Scheduled Queue
 DriverEntry (Discovered) - 844F4907-1EB5-4846-8C7A-56A3C9F9BEA8
 DriverEntry (Discovered) - 3AADE4EC-63CC-4A48-A928-5A374DD463EB
 DriverEntry (Discovered) - 7EE2C0C1-C21A-4113-A53A-66824A95696F
 DriverEntry (Discovered) - 58F7A62B-6280-42A7-BC38-10535A64A92C
 no more MM driver to dispatch, stop the dispatch request
 DriverEntry (Discovered) - 844F4907-1EB5-4846-8C7A-56A3C9F9BEA8
 DriverEntry (Discovered) - 3AADE4EC-63CC-4A48-A928-5A374DD463EB
 DriverEntry (Discovered) - 7EE2C0C1-C21A-4113-A53A-66824A95696F
 DriverEntry (Discovered) - 58F7A62B-6280-42A7-BC38-10535A64A92C
MmiHandlerRegister - GUID 60FF8964-E906-41D0-AFED-F241E974E08E - Status 0
MmiHandlerRegister - GUID 02CE967A-DD7E-4FFC-9EE7-810CF0470880 - Status 0
MmiHandlerRegister - GUID 27ABF055-B1B8-4C26-8048-748F37BAA2DF - Status 0
MmiHandlerRegister - GUID 7CE88FB3-4BD7-4679-87A8-A8D8DEE50D2B - Status 0
MmMain Done!
Cpu Driver EP 81516DA8
Received event - 0x0 on cpu 0
MmEntryPoint ...
CommBuffer - 0x81514B90, CommBufferSize - 0x18
MmEntryPoint Done
Received event - 0x0 on cpu 0
MmEntryPoint ...
CommBuffer - 0x81514B90, CommBufferSize - 0x30
```

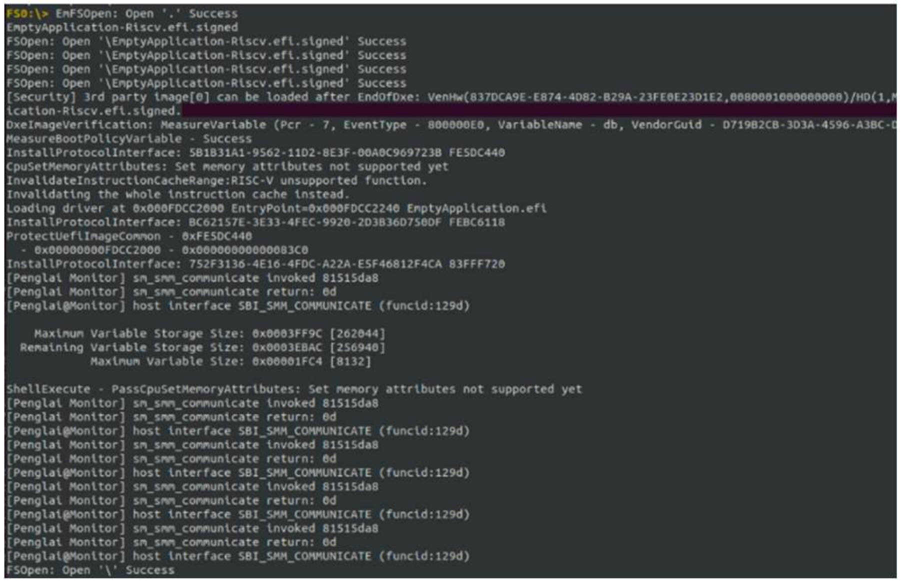
Secure domain log – StandaleonMm

Demo – UEFI Secure Boot



```
intel@intel-OptiPlex-7090: /ssd/workspace/penglai/0421-2                Ter

  Devices List                                  Press <Enter> to
> Driver Health Manager                         select Secure Boot
> RAM Disk Configuration                        options.


                                                Current Secure Boot
  Current Secure Boot State  Enabled            state: enabled or
  Attempt Secure Boot        [X]                disabled.
  Secure Boot Mode           <Standard Mode>
  Reset Secure Boot Keys
```

Enable UEFI Secure Boot

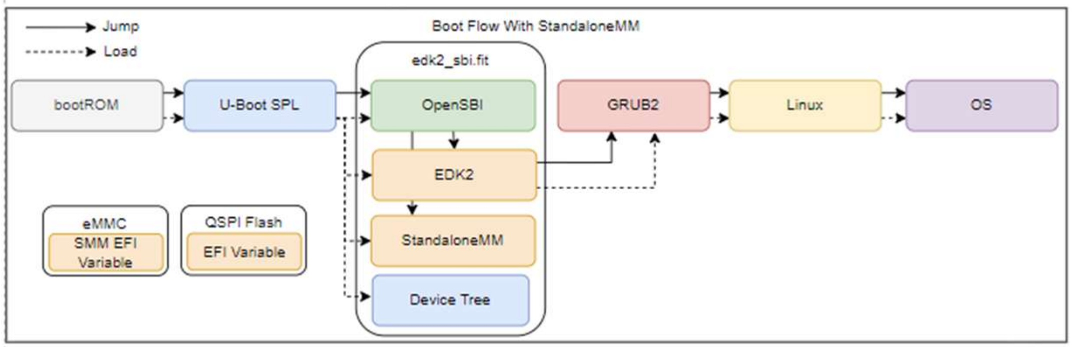


```
FS0:\> EmFSOpen: Open '.' Success
EmptyApplication-Riscv.efi.signed
FSOpen: Open '\EmptyApplication-Riscv.efi.signed' Success
FSOpen: Open '\EmptyApplication-Riscv.efi.signed' Success
FSOpen: Open '\EmptyApplication-Riscv.efi.signed' Success
FSOpen: Open '\EmptyApplication-Riscv.efi.signed' Success
[Security] 3rd party image[0] can be loaded after EndOfDxe: VenHw(837DCA9E-E874-4D82-B29A-23FE0E23D1E2,0000001000000000)/HD(1,M
ication-Riscv.efi.signed.
DxeImageVerification: MeasureVariable (Pcr - 7, EventType - 800000E0, VariableName - db, VendorGuid - D719B2CB-3D3A-4596-A3BC-D
MeasureBootPolicyVariable - Success
InstallProtocolInterface: 5B1B31A1-9562-11D2-8E3F-00A0C969723B FE5DC440
CpuSetMemoryAttributes: Set memory attributes not supported yet
InvalidateInstructionCacheRange:RISC-V unsupported function.
Invalidating the whole instruction cache instead.
Loading driver at 0x000FDCC2000 EntryPoint=0x000FDCC2240 EmptyApplication.efi
InstallProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF FEBC6118
ProtectUefiImageCommon - 0xFE5DC440
  - 0x00000000FDCC2000 - 0x00000000000083C0
InstallProtocolInterface: 752F3136-4E16-4FDC-A22A-E5F46812F4CA 83FFF720
[Penglai Monitor] sm_smm_communicate invoked 81515da8
[Penglai Monitor] sm_smm_communicate return: 0d
[Penglai@Monitor] host interface SBI_SMM_COMMUNICATE (funcid:129d)

    Maximum Variable Storage Size: 0x0003FF9C [262044]
  Remaining Variable Storage Size: 0x0003EBAC [256940]
            Maximum Variable Size: 0x00001FC4 [8132]

ShellExecute - PassCpuSetMemoryAttributes: Set memory attributes not supported yet
[Penglai Monitor] sm_smm_communicate invoked 81515da8
[Penglai Monitor] sm_smm_communicate return: 0d
[Penglai@Monitor] host interface SBI_SMM_COMMUNICATE (funcid:129d)
[Penglai Monitor] sm_smm_communicate invoked 81515da8
[Penglai Monitor] sm_smm_communicate return: 0d
[Penglai@Monitor] host interface SBI_SMM_COMMUNICATE (funcid:129d)
[Penglai Monitor] sm_smm_communicate invoked 81515da8
[Penglai Monitor] sm_smm_communicate return: 0d
[Penglai@Monitor] host interface SBI_SMM_COMMUNICATE (funcid:129d)
[Penglai Monitor] sm_smm_communicate invoked 81515da8
[Penglai Monitor] sm_smm_communicate return: 0d
[Penglai@Monitor] host interface SBI_SMM_COMMUNICATE (funcid:129d)
FSOpen: Open '\' Success
```

EFI application signature verification

VisionFiveV2 Platform (By StarFive)

- Used Key store in OTP, enabled U-Boot SPL Secure Boot Flow
- Ported StandaloneMM and enabled u-boot spl to load both edk2 and StandaloneMm with dynamic mode
- Enabled OpenSBI domain to isolate the memory resource
- Developed RPMB DXE driver
- Ported MMC Core Module to StandaloneMmPkg
- Ported RPMB DXE Driver to StandaloneMmPkg
- Tested and verified EDK2 UEFI Secure boot in VF2



* Limitation: iommu is not supported on VF2, so the domain isolation is

Call for action and future work

- Upstream the domain context switch/manager feature to OpenSBI in first

- Invovle us in PRXY and RPMI Spec work and PoC, so that we can merge the RPMI MM code to the PoC.

- As to the Hypervisor base platform design, CoVE will be a good secure monitor in such case, need more detail design for this, this probably the collaboration work in 2024
  - How MM runs in a TVM
  - How to Isolate/protect the resources in CoVE TSM
  - How to use CoVE API to communicate with the MM instance
  - How to use share memory in TVM to pass the MM message like RPMI

Thanks