OPEN POSSIBILITIES.

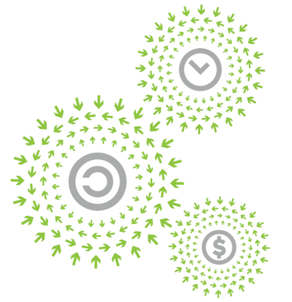
Enabling RAS on Xeon OCP Platforms Using Intel FSP and Coreboot

OCP GLOBAL SUMMIT

NOVEMBER 9-10, 2021

Enabling RAS on Xeon OCP Platforms Using Intel FSP and Coreboot

Ruth Li, Software Architect, Intel



OPEN POSSIBILITIES.

Agenda

- This talk is part of Intel® FSP exploration to support diversified boot frameworks along with Open system firmware initiative
- The topic depicted technical solution to enable Runtime RAS service under Coreboot framework via FSP API interface.

  - Open System Firmware and FSP Overview
  - Problem in Silicon specific SMM/Runtime modules working with FSP
  - Solution in Runtime RAS support with FSP
  - Demo of ‘RAS Enabling in FSP with Coreboot’

Open System Firmware and FSP Overview

OPEN SOURCE EDKII

EDKII:

https://github.com/tianocore/edk2

EDKII Minimum Platform Spec:

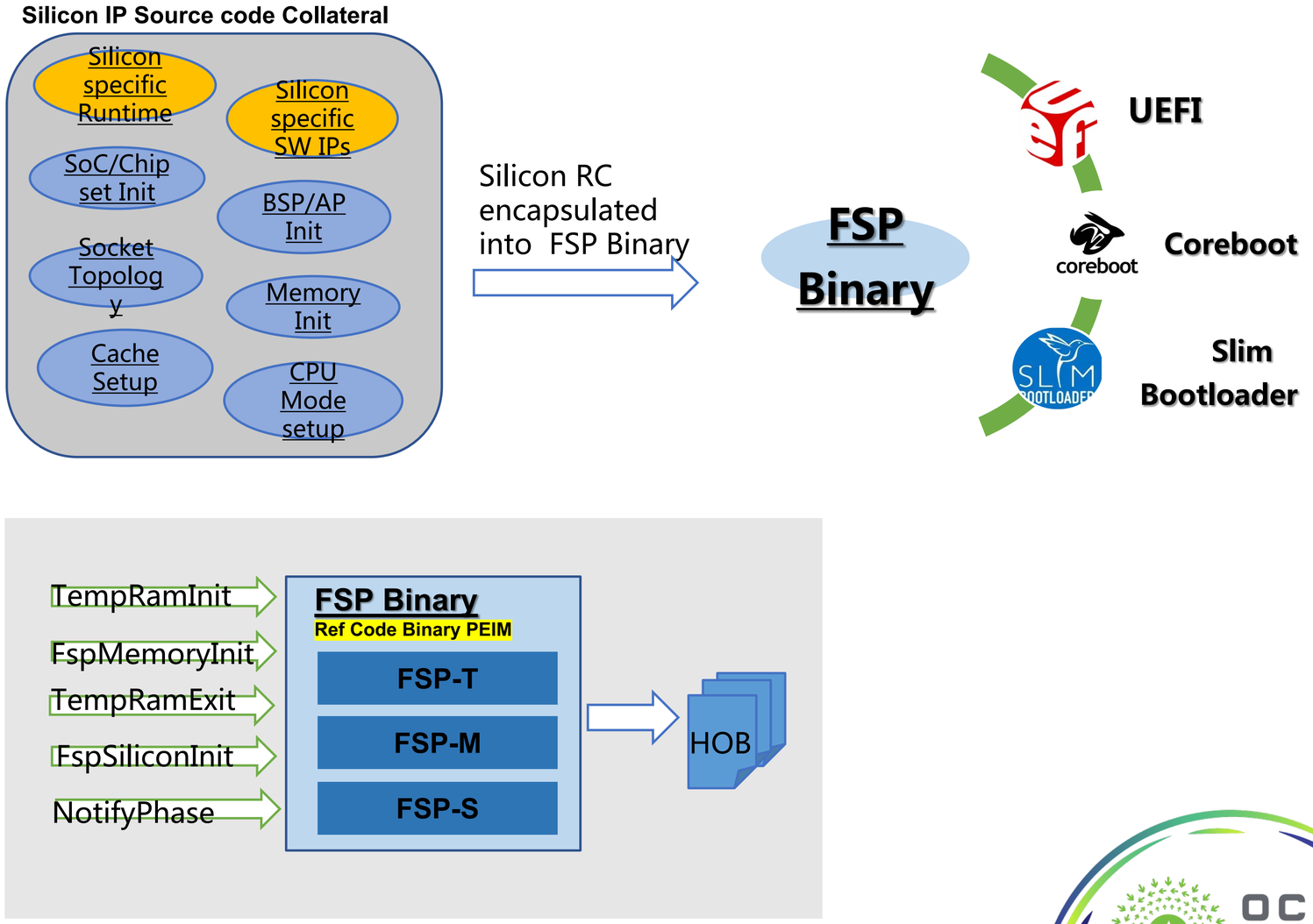
https://edk2-docs.gitbook.io/edk-ii-minimum-platform-specification

EDKII MinPlatform/OpenBoard:

https://github.com/tianocore/edk2-platforms/tree/master/Platform/Intel/MinPlatformPkg

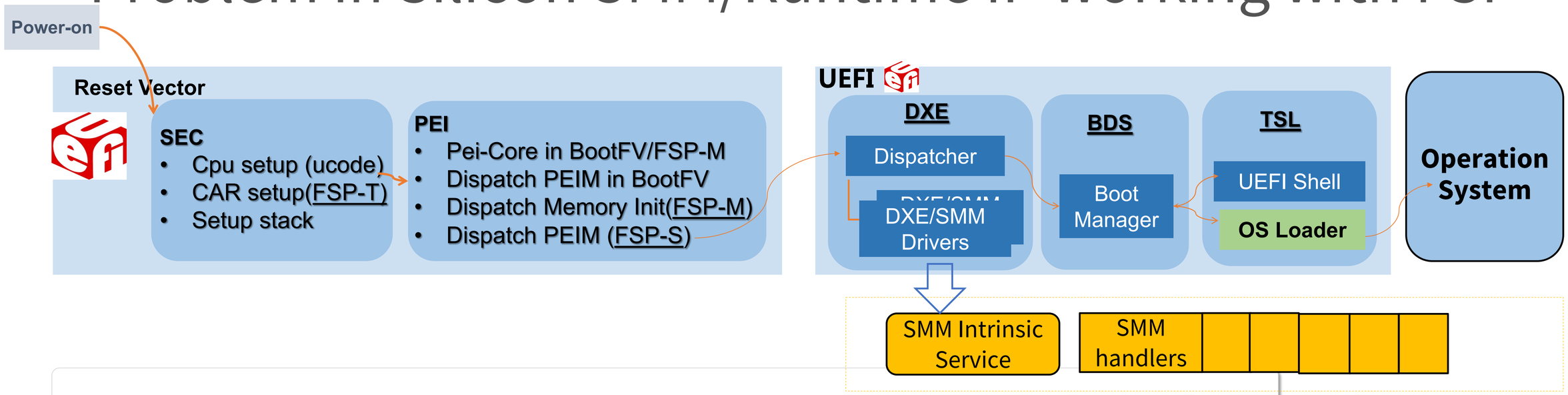
REDISTRIBUTED FIRMWARE BINARY

- **FSP(Firmware Support Package)**
  https://www.intel.com/content/www/us/en/intelligent-systems/intel-firmware-support-package/intel-fsp-overview.html
  - ✓ Latest spec: FSP 2.3
  - ✓ Supporting UEFI/Coreboot/Slim Bootloader
- **Microcode patch, Security ACM**
- **Mgmt.: SPS FW, Ignition FW**



OPEN POSSIBILITIES.

Problem in Silicon SMM/Runtime IP working with FSP



In UEFI Flavor, when working with Intel® FSP

- **CPU, Mem, Chipset initialization are executed in PEI phase, enclosed in Intel® FSP**
- **Silicon Specific SMM/Runtime modules are implemented in Close-source, Dispatched in DXE phase**

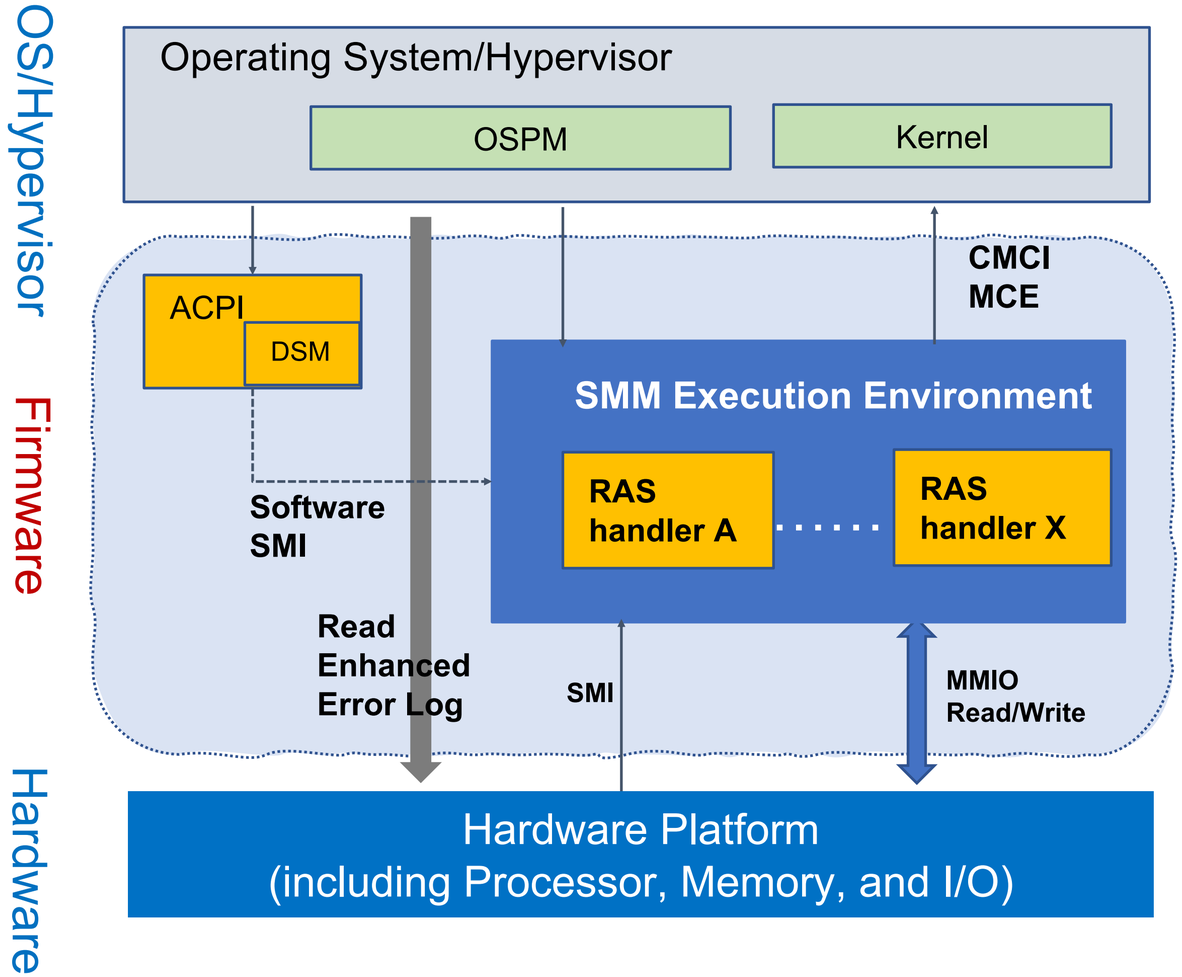
In non-UEFI flavor, if integrating with bootloaders (e.g., coreboot, Slimboot):

- **CPU, Mem, Chipset initialization are supported with Intel® FSP Binary**
- **UEFI DXE/SMM module CAN'T be Seamlessly integrated.**

Problem/Issue

- **No Unified Redistributable Silicon SMM/Runtime modules to support diversified bootloader for Open System Firmware Solution**

Reliability, Availability and Serviceability(RAS)



RAS Play important Role in modern Server/Cloud Service

- **Unexpected Downtime is Disruptive and Expensive - $5,000 for one-minute outage!!!**
- **Minimum Reliability Requirement is Rising – 4-nines, and 5-nines**

RAS Stack Topology

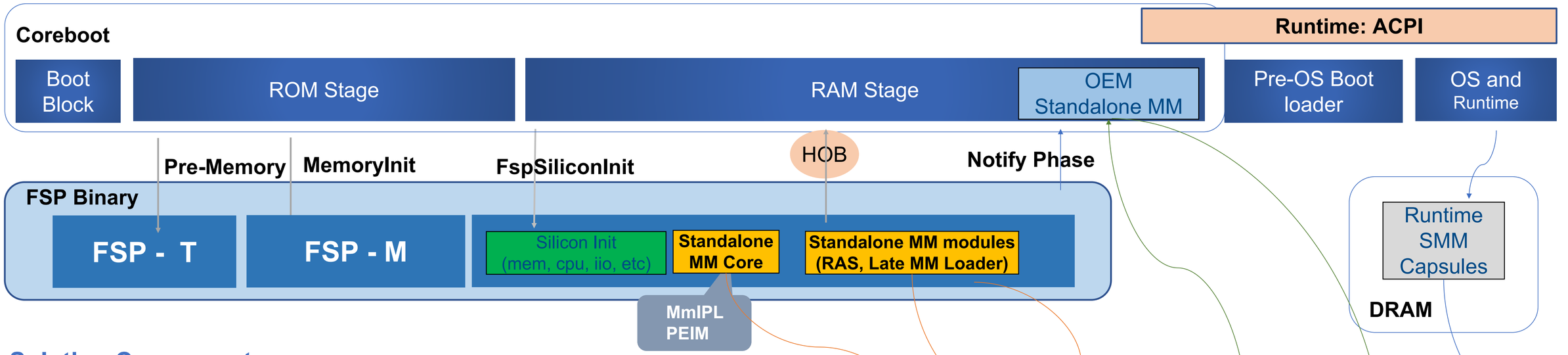
- **Boot-time: Initialize RAS driver and install Runtime/SMM RAS handler**
- **Run-time: When HW error happened**
  - **Firmware: RAS SMM handler process the error and Failure Telemetry**
  - **OS/Hypervisor: Further Error handling and Read error log etc.**

Gap with Intel® FSP Support

- **No Runtime/SMM Support**

Solution to enable RAS within Intel® FSP binary

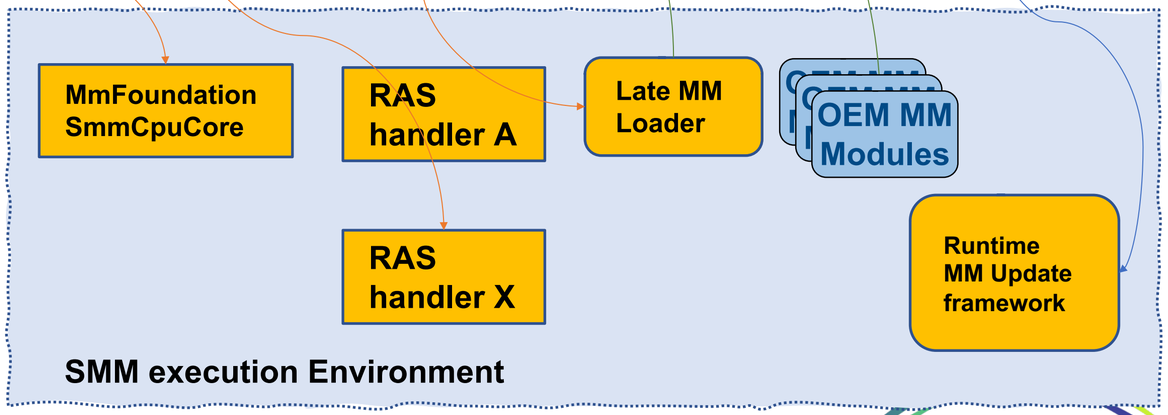
Solution to enable Standalone MM/RAS with Intel® FSP



Solution Components

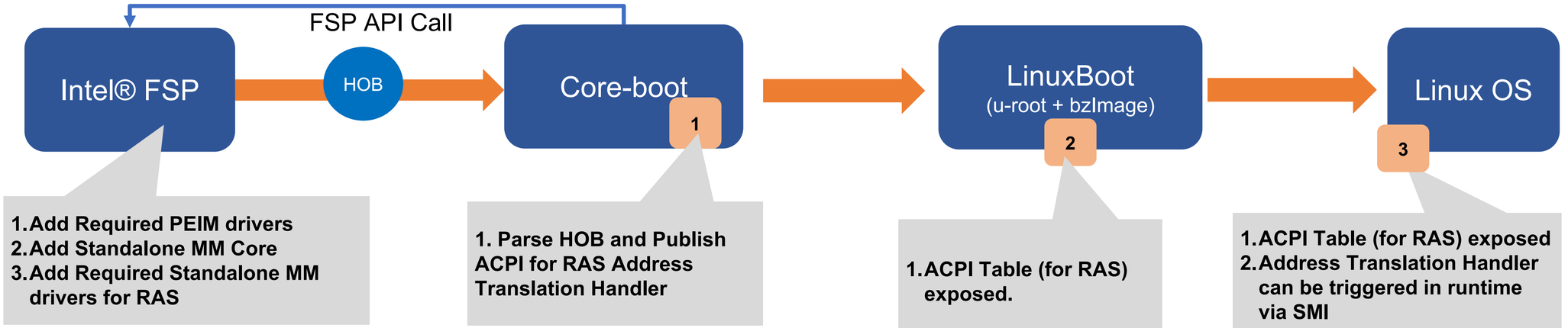
- RAS modules: Convert DXE/SMM RAS driver into PEIM and Standalone MM
- MM Core/Framework: support launching Standalone MM modules
- Late MM loader: dispatch OEM MM modules in later boot phase
- Runtime MM update: Introduce Runtime Update Framework to receive new MM module entered in OS time
- Encapsule Standalone MM Core, RAS module (PEIM, Standalone), MM Late loader, Runtime update framework into Intel® FSP binary

During Boot-time: Initialize MM core in early Silicon-Init phase, dispatch RAS driver from FSP



OPEN POSSIBILITIES.

RAS Demo Flow



Demo Configuration

- Demo use cases: RAS Address Translation Handler

Flash Image include

- FSP Binary (with API support) + Coreboot as boot loader
- Linux Boot used as pre-OS payload

Ubuntu as Target Production OS

Demo Flow: System Boot successfully, and doing checkpoint in Boot time and Runtime stage

During Boot Time

1 Confirm ACPI Table exposed as expectation in Coreboot

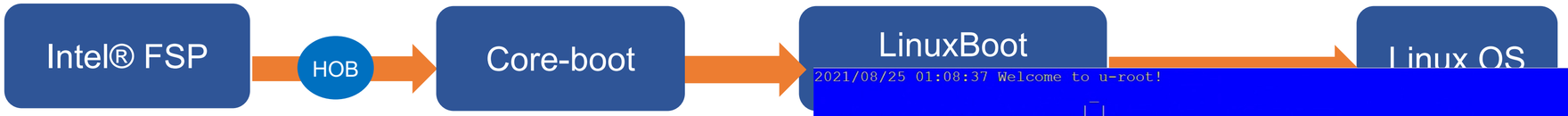
2 Confirm ACPI Table exposed as expectation in Linuxboot

Runtime Stage

3 Confirm ACPI Table is exposed correctly

- Triger Software SMI, Address Translation Handler was triggered as expectation

Demo




```
[AddressTranslationStandaloneSmmEntry] Enter AddressTranslationDsmEntry
[AddressTranslationStandaloneSmmEntry] RegisterSmi: Success[InstallSsdt] Enter Install Ssdt
[InstallSsdt] gMmCoreDataHobGuid Hob is Found!
[InstallSsdt] BFV Address: 5BA60000
[PatchSsdt] Enter Patch Ssdt
[PatchSsdt] Enter Patch Ssdt: Success
[InstallSsdt] PatchSsdt: Success
Dump data from 5BB19C68, size: 0x80
5BB19C68: 53 53 44 54 45 07 00 00 02 D4 49 4E 54 45 4C 00  | SSDTE.....INTEL.
5BB19C78: 41 44 44 52 58 4C 41 54 01 00 00 00 49 4E 54 4C  | ADDRXLAT....INTL
5BB19C88: 25 09 20 20 10 40 72 5C 5F 53 42 5F 5B 82 47 71  | %.  .@r\_SB_[.Gq
5BB19C98: 41 44 58 4C 08 5F 48 49 44 0D 49 4E 54 4C 30 30  | ADXL._HID.INTL00
5BB19CA8: 30 30 00 14 09 5F 53 54 41 00 A4 0A 0B 5B 80 42  | 00..._STA....[.B
5BB19CB8: 55 46 46 00 0C 10 88 57 7F 0C 34 01 00 00 5B 81  | UFF....W..4...[.
5BB19CC8: 44 0F 42 55 46 46 00 53 57 53 4D 20 43 4D 44 5F  | D.BUFF.SWSM CMD_
5BB19CD8: 20 53 54 53 5F 20 53 59 53 41 40 04 4E 4D 53 41  |  STS  SYSA@.NMSA
```

By FSP: 1). SMI handler is installed. 2). HOB is Exposed HOB to boot loader

```
2021/08/25 01:08:37 Welcome to u-root!

2021/08/25 01:08:37

2021/08/25 01:08:37 Failed to open ipmi device open /dev/ipmi0: no such file or directory, watchdog
[    0.000000] ACPI: Early table checksum verification disabled
[    0.000000] ACPI: RSDP 0x00000000000F0000 000024 (v02 COREv4)
[    0.000000] ACPI: XSDT 0x000000005BA200E0 000064 (v01 COREv4 COREBOOT 00000000 CORE 20200925)
[    0.000000] ACPI: FACP 0x000000005BA26740 000114 (v06 COREv4 COREBOOT 00000000 CORE 20200925)
[    0.000000] ACPI: DSDT 0x000000005BA20280 0064BA (v02 COREv4 COREBOOT 20110725 INTL 20200925)
[    0.000000] ACPI: FACS 0x000000005BA20240 000040
[    0.000000] ACPI: FACS 0x000000005BA20240 000040
[    0.000000] ACPI: SSDT 0x000000005BA26860 02414A (v02 COREv4 COREBOOT 0000002A CORE 20200925)
[    0.000000] ACPI: SSDT 0x000000005BA4A9B0 000745 (v02 INTEL  ADDRXLAT 00000001 INTL 20200925)
[    0.000000] ACPI: MCFG 0x000000005BA4B100 00003C (v01 COREv4 COREBOOT 00000000 CORE 20200925)
[    0.000000] ACPI: APIC 0x000000005BA4B140 000752 (v03 COREv4 COREBOOT 00000000 CORE 20200925)
[    0.000000] ACPI: SRAT 0x000000005BA4B8A0 000E80 (v01 COREv4 COREBOOT 00000000 CORE 20200925)
[    0.000000] ACPI: SLIT 0x000000005BA4C720 000030 (v01 COREv4 COREBOOT 00000000 CORE 20200925)
[    0.000000] ACPI: HPET 0x000000005BA4C750 000038 (v01 COREv4 COREBOOT 00000000 CORE 20200925)
```

```
ACPI: added table 2/32, length now 44
ACPI:     * SSDT(From Hob)
acpi_create_ssdt_from_hob start ...
Acpi hob at: 641a97d0, size: 10
hob_data->count: 1
        ->address: 77476000
get_next_acpi_table from 77476000: SSDT
index: 0; current: 77476000
current_table->signature: SSDT
find table at: 77476000
Found one ssdt at: 77476000
        length: 745
  revision: 2
0x77476000: 53 53 44 54 45 07 00 00 02 d4 49 4e 54 45 4c 00  SSDTE.....INTEL.
0x77476010: 41 44 44 52 58 4c 41 54 01 00 00 00 49 4e 54 4c  ADDRXLAT....INTL
0x77476020: 25 09 20 20 10 40 72 5c 5f 53 42 5f 5b 82 47 71  %.  .@r\_SB_[.Gq
```

In Coreboot: Received from HOB, and build ACPI record

```
kexec_core: Starting new kernel
[    0.000000] Linux version 4.15.0-112-generic (buildd@lcy01-amd64-027) (gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)) #113-Ubuntu SMP Thu Jul 9 23:41:39 UTC 2020 (Ubuntu 4.15.0-112.113-generic 4.15.18)
root@cb-test:/sys/firmware/acpi/tables# xxd -l 0x80 SSDT2
00000000: 5353 4454 4507 0000 02d4 494e 5445 4c00  SSDTE.....INTEL.
00000010: 4144 4452 584c 4154 0100 0000 494e 544c  ADDRXLAT....INTL
00000020: 2509 2020 1040 725c 5f53 425f 5b82 4771  %.  .@r\_SB_[.Gq
00000030: 4144 584c 085f 4849 440d 494e 544c 3030  ADXL._HID.INTL00
00000040: 3030 0014 095f 5354 4100 a40a 0b5b 8042  00..._STA....[.B
00000050: 5546 4600 0c10 8857 7f0c 3401 0000 5b81  UFF....W..4...[.
00000060: 440f 4255 4646 0053 5753 4d20 434d 445f  D.BUFF.SWSM CMD_
00000070: 2053 5453 5f20 5359 5341 4004 4e4d 5341   STS_ SYSA@.NMSA
root@cb-test:/sys/firmware/acpi/tables#
```

In Linuxboot: Acpi table is exposed

```
root@cb-test:/home/cb-test/IoAccess# ./IoAccess -w 0xB2 0x20
MmEntryPoint ...
gMmCoreMmst.NumberOfCpus=224
[AddressTranslation] SmiHandler: Start ...
[AddressTranslation] SmiHandler: End ...
MmEntryPoint Done
Port 0xb2 Output: 0x20( )
root@cb-test:/home/cb-test/IoAccess#
```

In Ubuntu: SMI handler is triggered, ACPI table is installed

OPEN POSSIBILITIES.

Call to Action

Get involved into Open Compute Project

Open System Firmware: https://www.opencompute.org/projects/open-system-firmware

Engage with Intel on FSP and Open Firmware Development

Intel FSP: https://www.intel.com/content/www/us/en/intelligent-systems/intel-firmware-support-package/intel-fsp-overview.html

Intel EDKII: https://github.com/tianocore/edk2

Thank you!

NOVEMBER 9-10, 2021