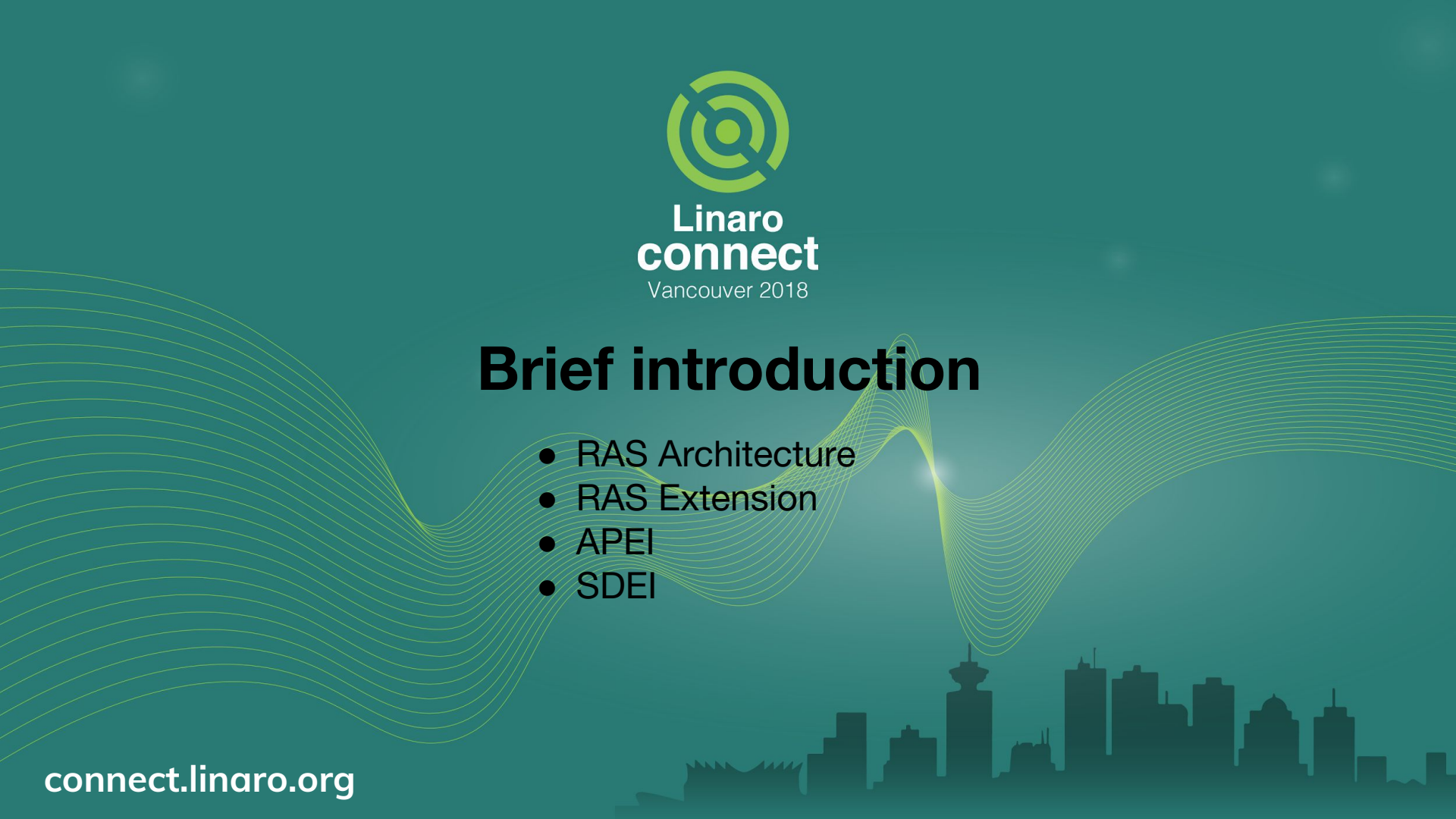
YVR18-103:Reliability, Availability, and Serviceability (RAS) on ARM64 status

Fu Wei <wefu@redhat.com>

AGENDA

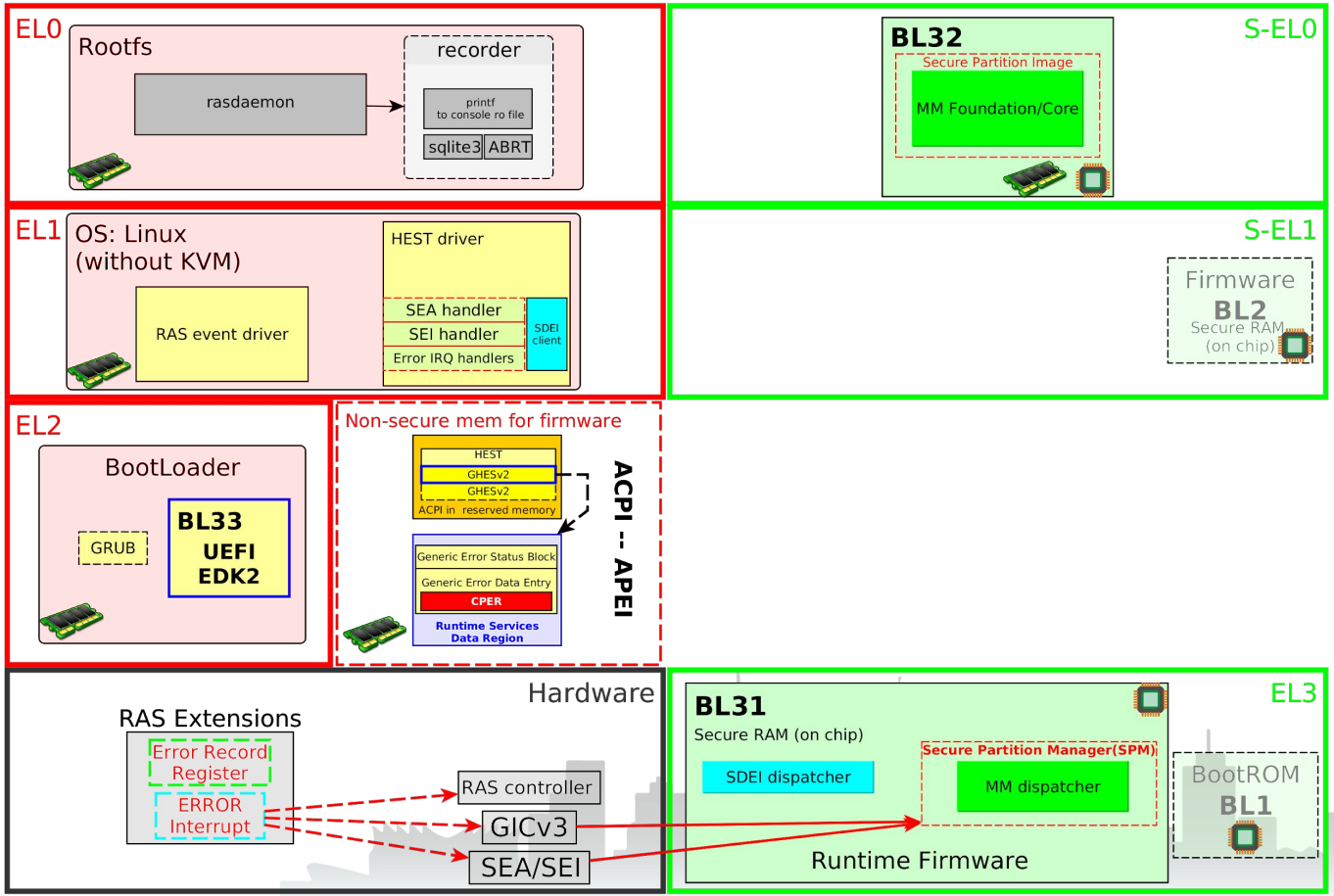
1. Brief introduction
   - Architecture, RAS Extension, APEI, SDEI
2. Prototype Solution
   - **Firmware First Error Handling**
     - **Overview**
     - **APEI protocol (boot time)**
     - **CperLib (run time)**
   - BERT
     - When catastrophic errors occur
     - After emergency reboot
3. Future Plans



Brief introduction

- RAS Architecture
- RAS Extension
- APEI
- SDEI

RAS Architecture



LEG
Server

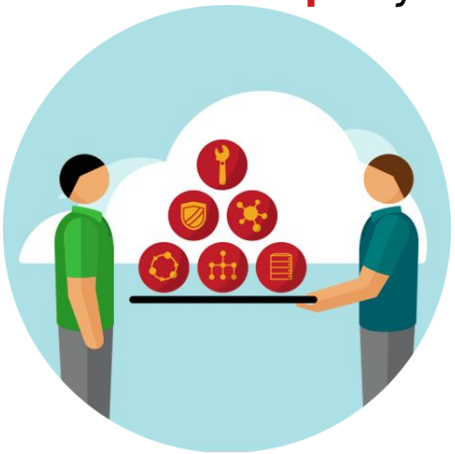
RAS Extension

Gather HW error info for FW

ESB instruction
Help to **locate** Error

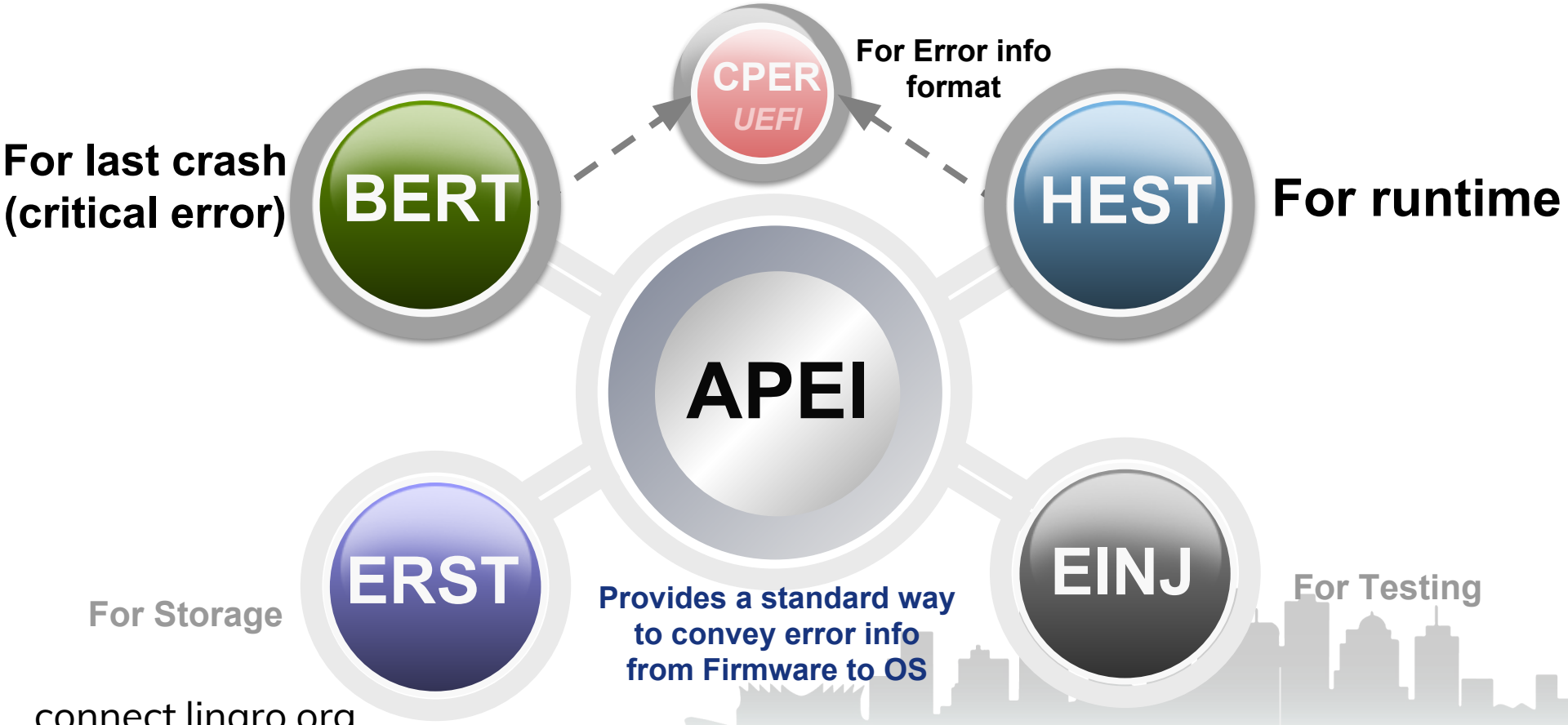
RAS Extension registers

- Provide the error **info** to FW
- Control the **Interrupt** by FW



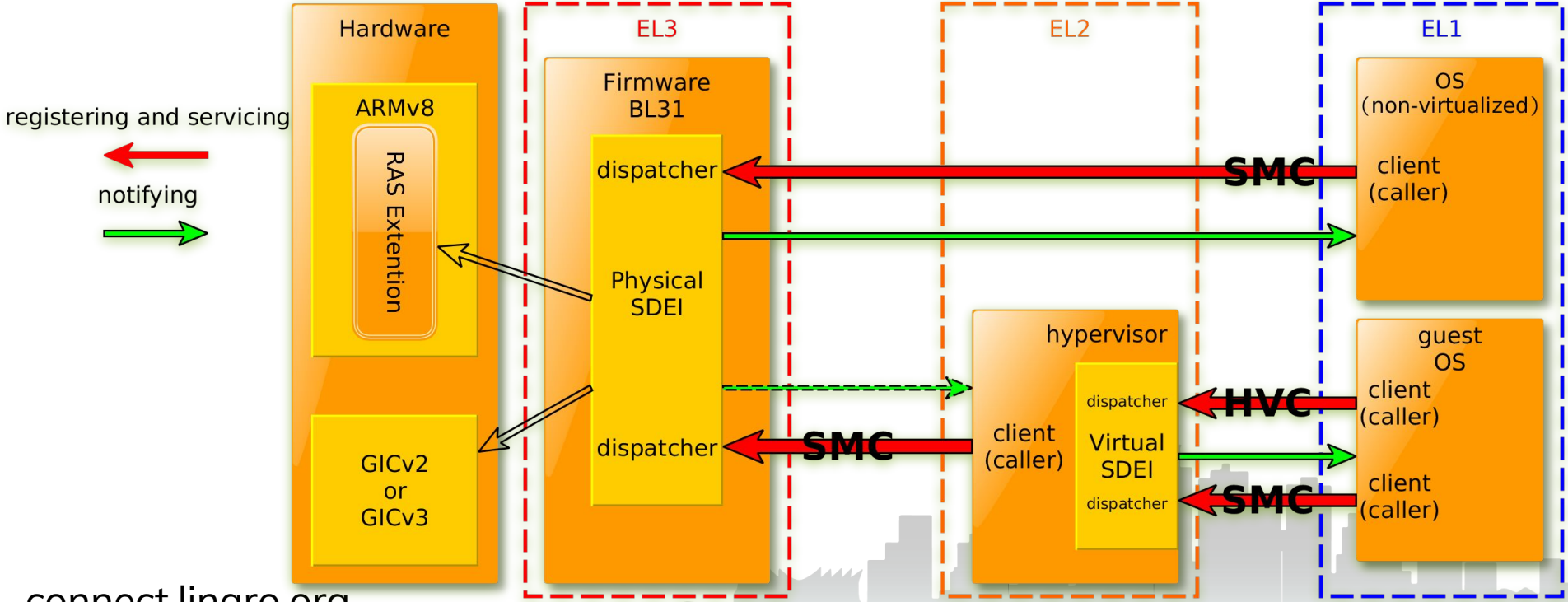
ARMv8-A RAS extensions standardize
the interface between HW and FW

APEI (ACPI Platform Error Interfaces)



SDEI usage in RAS

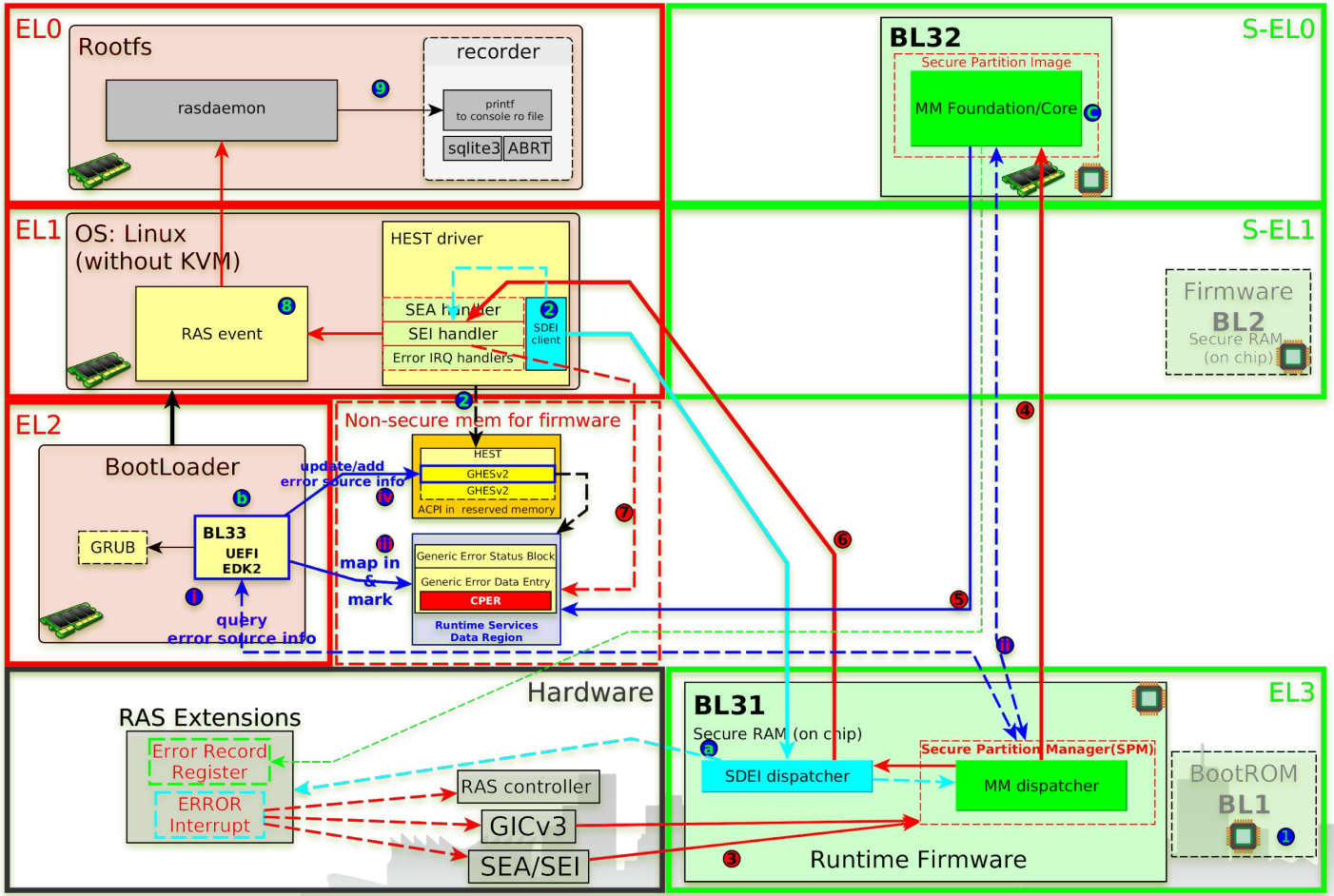
Software Delegated Exception Interface : An interface between FW & OS, for registering, notifying and servicing system events using SMC/HVC. [SDEI Specification (ARM DEN0054A)]



Prototype Solution

- Firmware First Error Handling
- BERT

Firmware First Error Handling Overview



LEG
Server

Uncorrected Error
-- HEST & MM

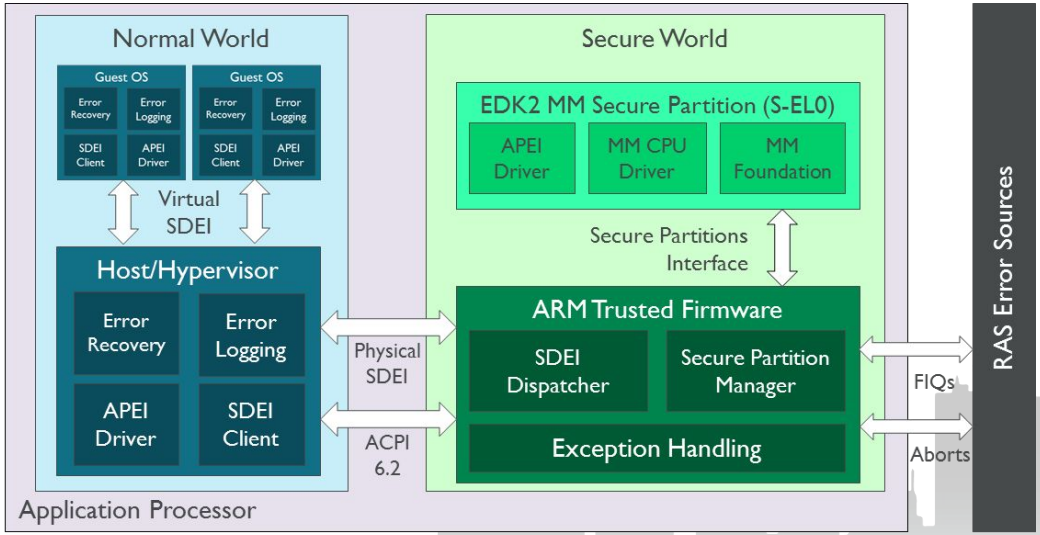
1. System boot: BootROM-->BL2-->BL3x
   a. **BL31** initializes SPM, SDEI dispatcher and BL32(MM dispatcher)
   b. **UEFI** (BL33), DXE, UEFI Platform Driver:
      i. query MM partitions by **APEI protocol** for error source info
      ii. MM partitions return error source info back to **UEFI**
      iii. **UEFI** map in and mark error record region as Runtime Services Data Region
      iv. Update/add error source info in **HEST**
2. OS starts running: **HEST driver** scan HEST table and register error handlers by **SDEI**
3. UE occurred, the event will be routed to EL3 (SPM)
4. SPM routes the event to RAS error handler in S-EL0 (MM partitions)
5. MM Foundation **(CperLib)** creates the CPER blobs by the info from RAS Extension
6. SPM notifies **SDEI** to call the corresponding OS registered handler
7. OS gets the CPER blobs by Error Status Address block, process the error, try to recovery.
8. report the error event by **RAS event**
9. **rasdaemon** log error info from RAS event to recorder

Boot Time **RunTime**

MM Secure Partition & Secure Partition Manager (SPM)

- **MM Secure Partition** implements management functions , runs in **S-EL0** to achieve isolation from S-EL1 & EL3
- Leverages existing firmware code based on EDK2: **StandaloneMm**
- Minimise code in EL3 and delegate **RAS error handling**

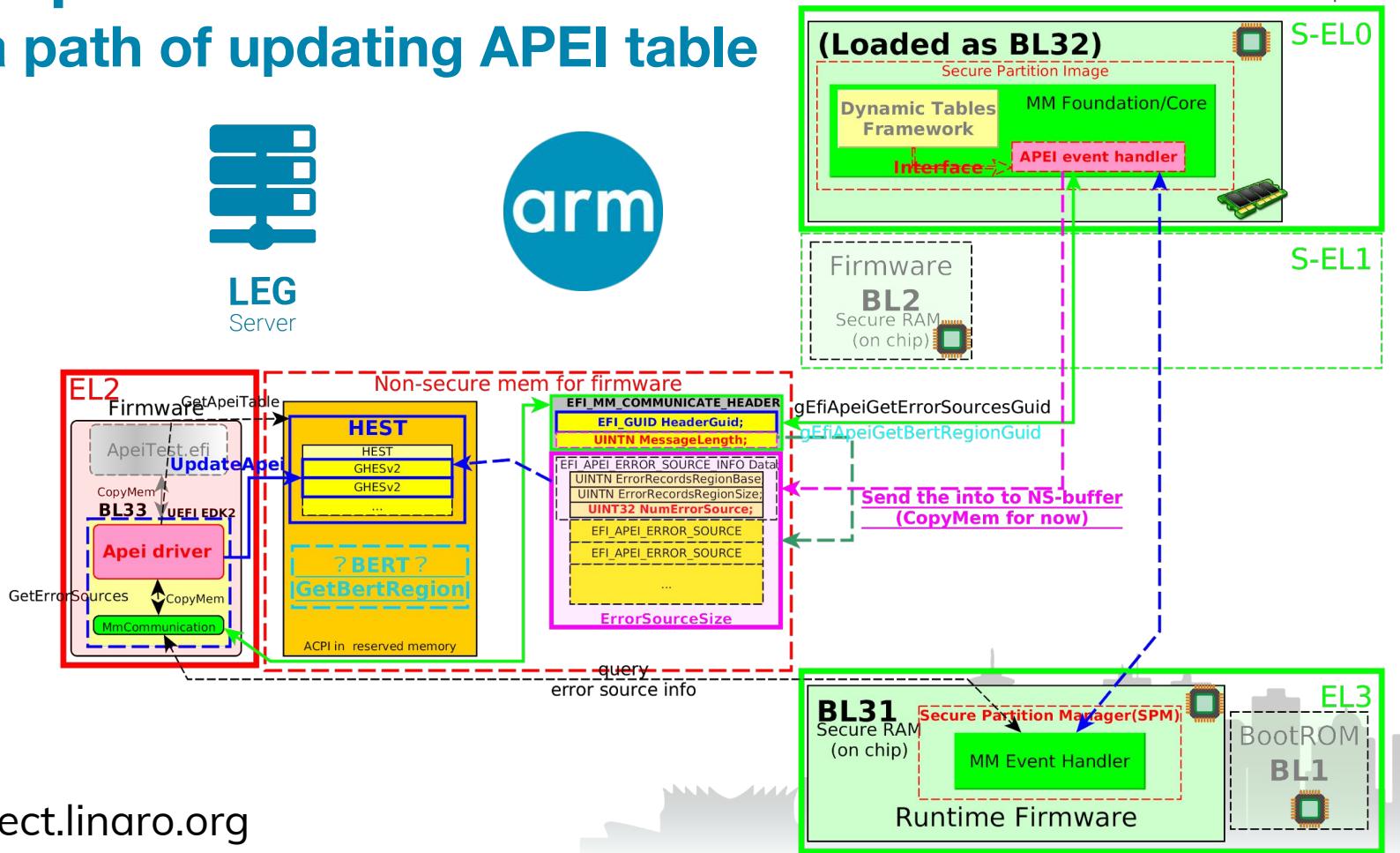
- **Secure Partition Manager** in **BL31** exports standard ABI to
  - Initialize the partition
  - Delegate SMC requests to the partition



Firmware First Handling of AP RAS Errors

APEI protocol:

Data path of updating APEI table



APEI protocol:

update APEI tables by the info from StandaloneMm at boot time

Main

- **Apei.c**
  - **UpdateApei** updates error source information to the specified APEI(HEST/BERT) Table

```
typedef
EFI_STATUS
(EFIAPI *EFI_APEI_UPDATE) (
  IN CONST EFI_APEI_PROTOCOL    *This,
  IN UINT32                     Signature
 );
```

- **ApeiCommon.c**
  - **GetApeiTable**
  - **AcpiTableChecksum**

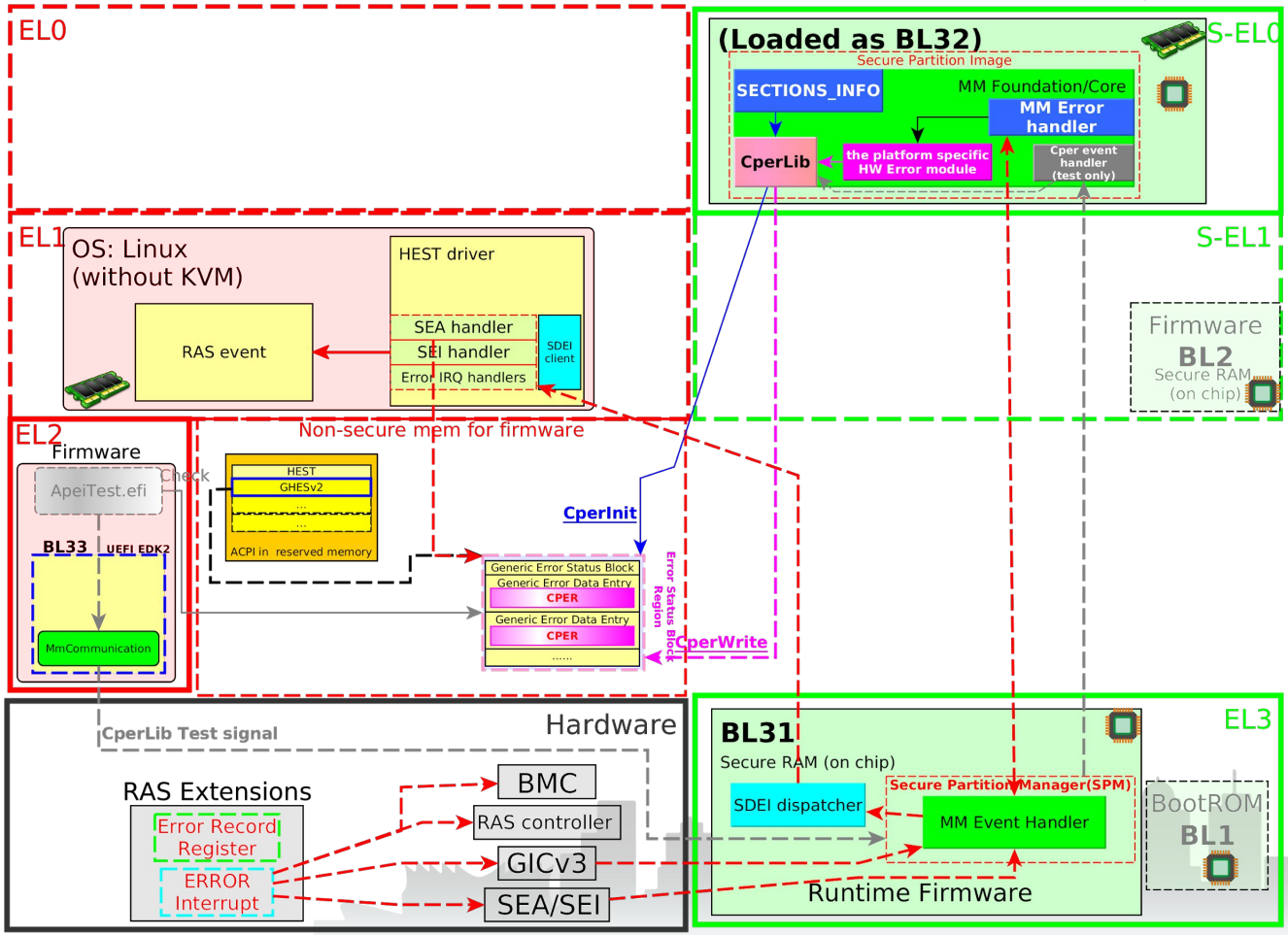
Call

For HEST table: ApeiHest.c

- **UpdateHest**
  - **GetErrorSources**
    - get ErrorSourceInfo by MmCommunication from SPI with **gEfiApeiGetErrorSourcesGuid**
    - **BufferSize**
  - **UpdateGhes**
    - AllocateZeroPool for GHESv2, do a basic init
    - **ImportErrorSourceInfo**
      - **ReserveRuntimeMemory**

For BERT table: ApeiBert.c(TBD):

CperLib : Data path of generating the CPER blob



LEG Server

CperLib:

Wrap “section” data into CPER blob, send to NS-buffer for OS

- **CperInit: parse the error source info, and return the error record address info**

```
EFI_STATUS
EFIAPI
CperInit (
  IN EFI_APEI_ERROR_SOURCE  *ErrorSource,
  IN OUT UINTN              *ErrorRecordAddress
);
```

- **CperWrite: creates a CPER blob at the defined memory region from Section info data**

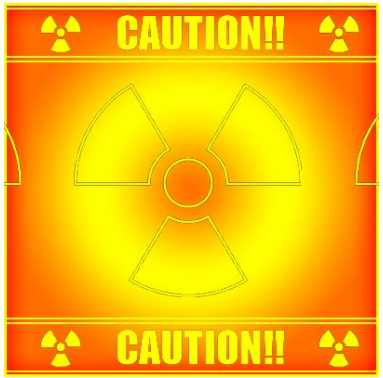
```
EFI_STATUS
EFIAPI
CperWrite (
  IN SECTIONS_INFO          *SectionInfo,
  IN UINTN                  ErrorRecordAddress
);
```

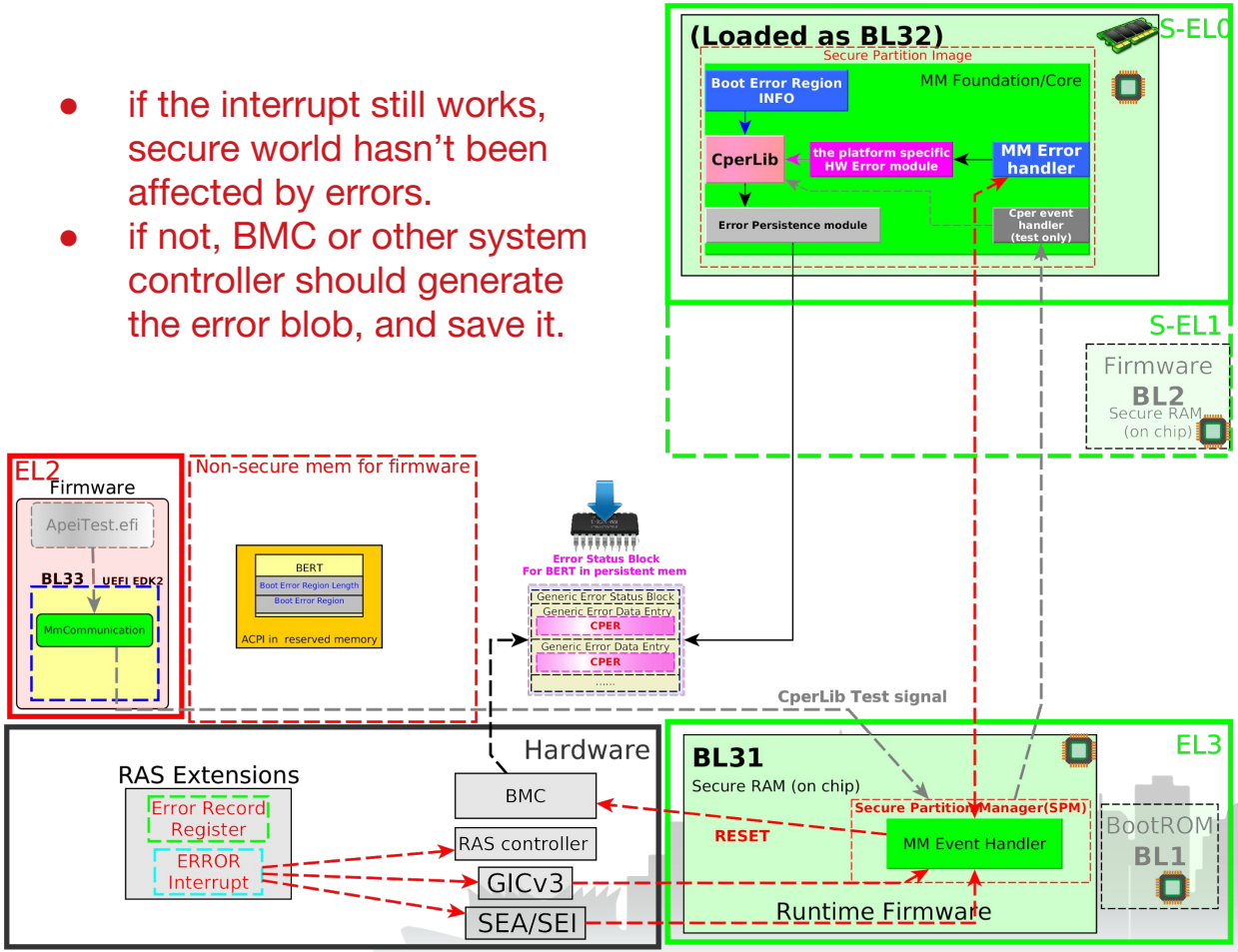
- **CperInit: the Dynamic Tables module passes the ErrorSource to help CperLib get the memory region info for other APIs, like CperWrite**

- **CperWrite: wrap the given section data into CPER blob and put it into the specific memory region**

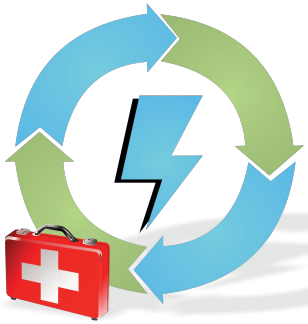
BERT : When catastrophic errors occur

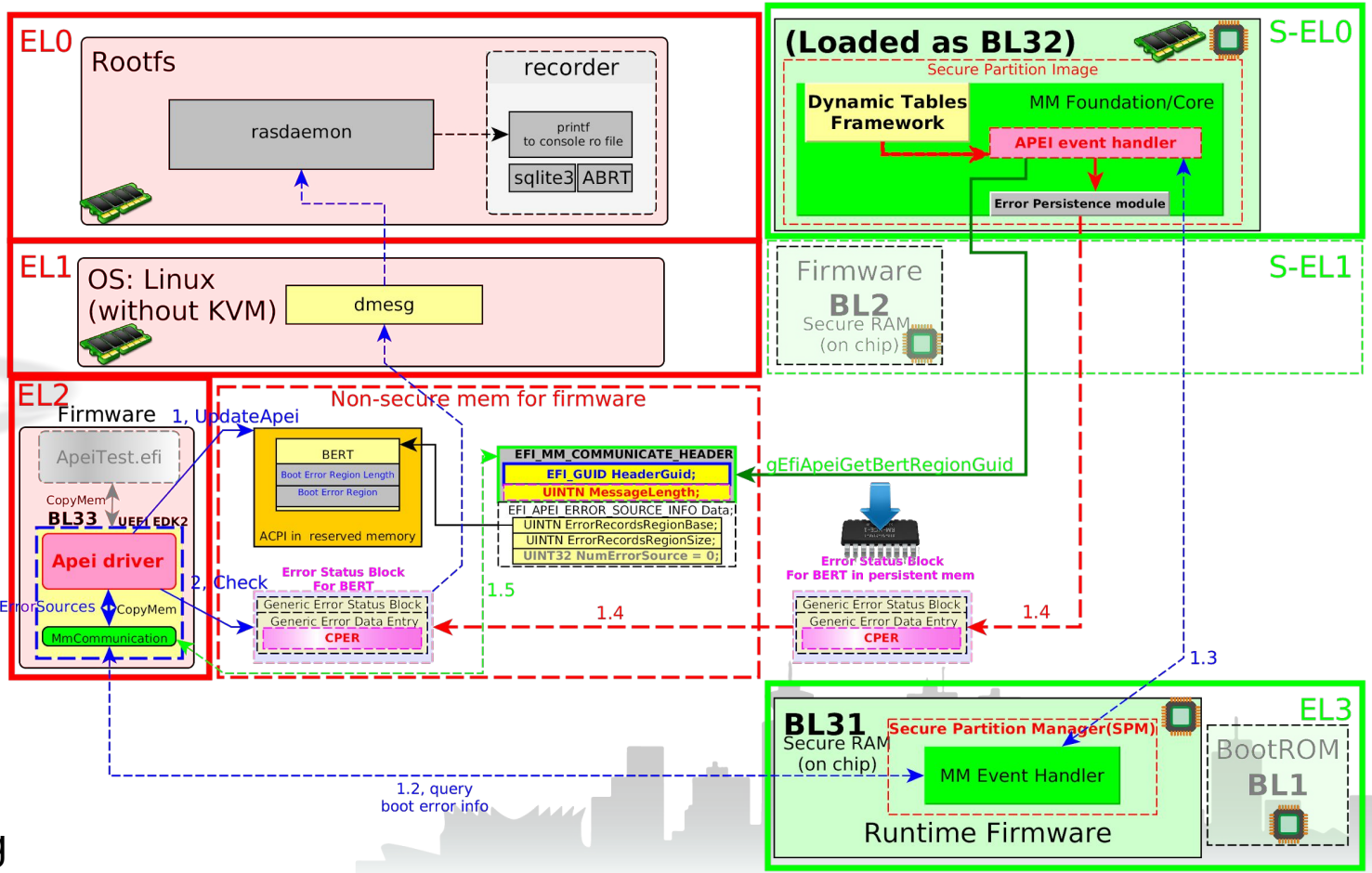
- if the interrupt still works, secure world hasn't been affected by errors.
- if not, BMC or other system controller should generate the error blob, and save it.





BERT: After emergency reboot





Future Plans

- Ongoing development
- TODO list for Reference Solution

Ongoing development

- QEMU
  - upstream sec-uart support for qemu
  - Move to platform
- ARM-TF
  - upstream qemu support for StandaloneMm
- EDK2
  - Improve and upstream APEI protocol and CperLib code

TODO list for Reference Solution

- Hardware
  - Test on a real hardware (ARMv8.2, including RAS extension)
- Firmware
  - EDK2:
    - prototype solution of BERT Support in StandaloneMm
    - ERST and EINJ implementation

Acknowledgments

- Achin Gupta (ARM)
- Al Stone (Red Hat)
- Arvind Chauhan (ARM)
- Daniil Egranov (ARM)
- John Feeney (Red Hat)
- Leif Lindholm (Linaro)
- Supreeth Venkatesh (ARM)
- Thomas Abraham (ARM)

Thank You

#YVR18

YVR18 keynotes and videos on: connect.linaro.org

For further information: www.linaro.org