

Linux Plumbers Conference

Vienna, Austria | September 18-20, 2024



State of CXL Error Handling

2024 LINUX PLUMBERS CONFERENCE
CXL MICROCONFERENCE

Robert Richter <rrichter@amd.com>
Terry Bowman <terry.bowman@amd.com>



RAS: Reliability, Availability and Serviceability

CXL RAS features include:

- Error Handling
 - Link and Protocol Errors
 - Device Errors
- CXL Error Injection
- Poison/Viral Handling
- Maintenance
 - Event Records for system notifications
 - Commands available for Maintenance operations
- CXL Error Isolation



CXL Error Handling

Facilities used to report and handle CXL related errors:

- Link and Protocol Errors
 - PCIe AER
 - Restricted mode: RCH DP, RCH UP, RCD RCiEP
 - VH mode: All components except CXL Host Bridge
- Device errors
 - Event logging
- Firmware Error Reporting



Link and Protocol Errors

- Modes available:
 - Restricted mode: RCH DP, RCH UP, RCD RCiEP
 - VH mode: CXL components in PCIe hierarchy: RP, DSP, USP, EP
- Protocol types:
 - CXL.io
 - CXL.cache and CXL.mem, referred to as CXL.cachemem
- Registers:
 - PCIe AER errors (PCIe config space)
 - PCIe DVSEC for CXL Devices and Ports (PCIe config space)
 - CXL RAS capability (PCIe memory space, CXL.cachemem block, all components except host bridge)
- Uses PCIe AER
- Kernel generates Tracepoints for an error



AER: Advanced Error Handling and Recovery

- Standard PCIe error reporting mechanisms over CXL.io as AER
- AER is supported by all CXL components (is a PCIe facility)
- Reporting via PCIe AER:
 - CXL.io: Errors logged in in their respective AER Extended Capability
 - CXL.cachemem: Components report errors using Uncorrectable and/or Correctable Internal PCIe AER errors (UIEs and/or CIEs), error status is in the CXL RAS capability
 - Linux challenge: portdrv implements AER service for ports (root port or switch ports), but implementation did not allow a custom CXL port driver to handle CXL specifics
- Linux kernel support exists for PCIe AER



PCIe AER and CXL Restricted Mode (RCH/RCD)

- In restricted mode RCH DP and RCD UP do not show up on the PCI bus
- Registers are memory mapped and requires CXL specific access to:
 - AER registers (in RCRB)
 - CXL RAS caps (in Component Registers, MEMBAR0 of RCRB)
- Notified through a RCEC
- RCH: Root complex event collectors (RCEC) are used to report AER in CXL restricted host (RCH) mode



Linux Kernel Support for Link and Protocol Errors

- CXL port device support (currently in development)
 - <https://lore.kernel.org/all/20240617200411.1426554-1-terry.bowman@amd.com/>
- RCH DP error handling (6.5/6.7)
 - 7f946e6d830fbdf411cd0641314edf11831efc88
 - 0c0df63177e37ae826d803280eb2c5b6b6a7a9a4
- CXL RAS cap error unmasking (6.3)
 - 5a6fe61facdb7f830895712b31fb39f544ffc165
- CXL AER handling and correctable error extensions, CXL RAS cap and tracepoint support (6.2)
 - e0f6fa0d425f745a887e640be66e22b45451e169



Device Errors

- Device may report:
 - Poison/Viral
 - Non-memory errors:
 - PCIe AER/CXL RAS capability of the endpoint, same as for ports
 - Memory Errors:
 - Memory error logging and signaling mechanisms defined by the CXL specification
 - Errors are logged as events using the same general-event logging facility as for general device events
 - Notification may use mailbox MSI/MSI-X device interrupt



Event Logging

- CXL endpoints and switches use the CXL mailbox with the component command interface (CCI) protocol to communicate events
- Events are used to report errors, make requests, and provide responses between CXL components
- Mailbox commands exist to read and clear the Event Log
- Driver status:
 - Event logging exists upstream for media, DRAM, and memory module events
 - DCD events currently in development
 - Physical switch events are TODO



Linux Kernel Support for Device Errors and Event Logging

- UAPI support of CXL log related mailbox commands (6.10)
 - [db4fdb73f9835cab1e21c901e59d17fad32a0369](#)
- CXL background command (6.5)
 - [dcfb70610d40704d929d824db36b1444c8f37f7a](#)
- Poison list and injection infrastructure (6.4)
 - [856ef55e7e1fb411cd42b917bac2a7aaf75344ae](#)
- CXL event log and interrupt support (6.3)
 - [Dbe9f7d1e155b97a42f7da81e22acc98fe0a9072](#)
- Mailbox support (5.16)
 - [dd72945c43d34bee496b847e021069dc31f7398f](#)



CXL Error Reporting using Firmware (APEI)

ACPI Platform Error Interfaces (APEI):

- Errors reported via Common Platform Error Record (CPER) and Generic Hardware Error Source (GHES)
- EINJ error types added for CXL.cachemem protocol errors

CXL CPER Records introduced:

- CXL protocol errors:
 - CXL agents:
 - Restricted device and downstream port (RCD/RCH) (UEFI 2.9)
 - VH: devices and ports (RP, USP, DSP) (UEFI 2.10)
 - Record includes:
 - PCI Express Capability Structure (PCIe config space)
 - CXL Device or Port DVSEC (PCIe config space)
 - PCIe DVSEC for CXL Device
 - Flex Bus Port DVSEC
 - CXL RAS Capability Structure (PCIe memory space, mmio)
- CXL Component Events:
 - CXL Event Record of the component



Firmware Error Reporting (Continued)

ACPI _OSC (Operating System Capabilities)

- Used to pass error handling from FW First mode to OS First
- OS negotiates control with Firmware
- Introduced _OSC interface for a CXL Host Bridge with support and control fields for:
 - CXL Protocol Error Reporting Supported
 - CXL Memory Error Reporting Control
 - Event log of Component errors

Kernel Tracepoints

- Used to receive error records for further handling (RAS daemon etc.)
- Same for both, FW First and OS First



Firmware Error Reporting - Linux Kernel Support

- CXL CPER Component Event support, CXL driver (6.8/6.10)
 - 3601311593eb44d34f142b993cb6f38f9a7863b3
 - df2a8f4b444f92152a9e981d9b0eb0776130892a
- CXL Error INJection support (6.9)
 - 75f4d93ee8faf08546f3cc4c3d96c866b24358c8
- CXL CPER event decoding, ACPI driver (6.2)
 - fc4c9f450493daef1c996c9d4b3c647ec3121509
- CXL _OSC support (5.19/6.6/6.7)
 - 9d004b2f4fea97cde123e7f1939b80e77bf2e695
 - 2ad78f8cee9ae6cd99c685e217e89fa99cc222ef
 - b3741ac86c8e648709506102f7ab51905d50df43



Poison and Viral

- 'If demand accesses to memory result in an uncorrected data error, the CXL device must return data with poison. The requester (processor core or a peer device) is responsible for dealing with the poison indication.' [1]
- Media and Poison Management: Mailbox CCI commands support poison operations: list, inject, clear [2]
- Endpoint communicates poison state using CXL.mem/CXL.cache M2S and S2M [3]
- Data is tagged as poisoned in the endpoint device hardware and handled by host (part of memory management)
- When the device communicates Viral, data shall be considered suspect
- Viral control and status is in the PCIe DVSEC for CXL Device registers
- An uncorrected fatal error generates a Viral indication
- Device can send or receive a Viral indication to or from the host
- Driver support for Poison Management

[1] CXL3.1 - 12.2.3.1 CXL.cache and CXL.mem Errors

[2] CXL3.1 – 8.2.9.9 Memory Device Command Sets

[3] CXL3.1 – 3.3.5 M2S Request



CXL Isolation

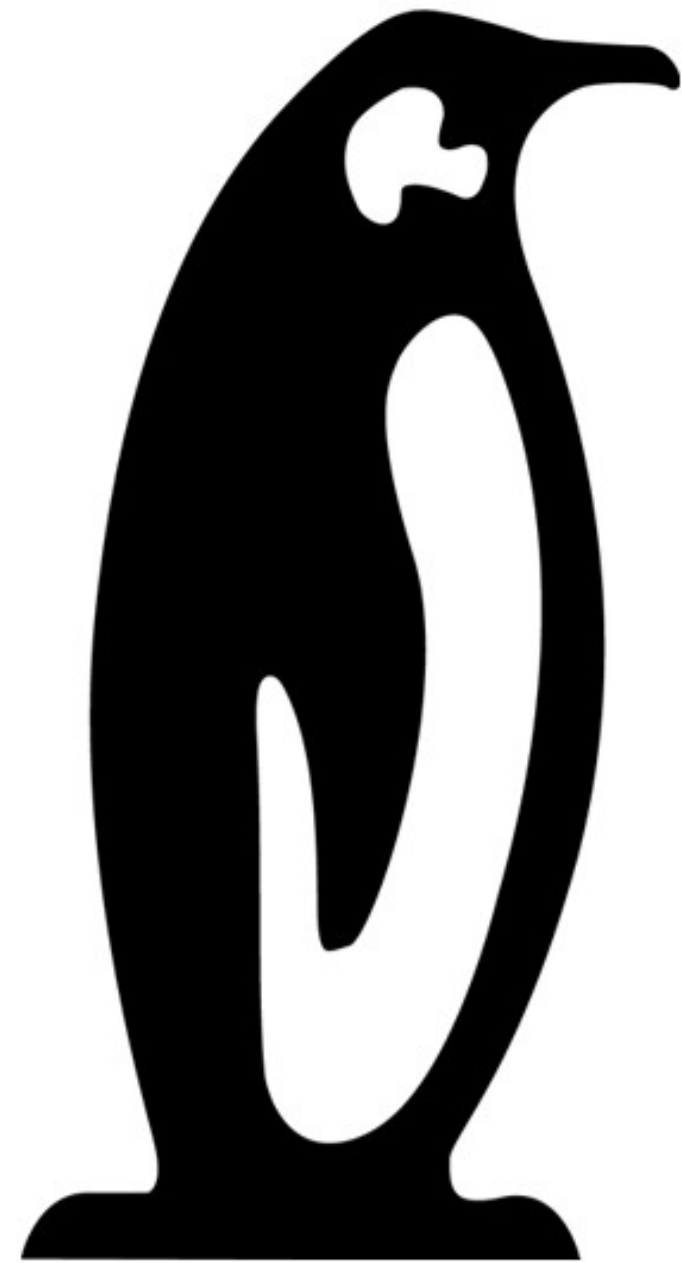
- ‘CXL isolation is the mechanism that is used for graceful handling of Surprise Hot-Remove of CXL adapters.’ [1]
- Designed to isolate the CXL memory device in 2 cases:
 - Link down or
 - Protocol response timeout
- Challenges: If CXL memory is added by default at boot time then a kernel reference is added to the CXL memory making it 'unmovable'. CXL isolation takes memory offline, but requires the memory is 'movable'.
- Solution: Use special purpose memory (SPM) with default offline kernel commandline parameter (`memhp_default_state=offline`).
 - Memory is not onlined when using SPM with default offline commandline parameter.
- Driver status:
 - Patchset upstreamed, but is not currently accepted
 - Need users before accepting upstream
 - <https://lore.kernel.org/all/20240215194048.141411-1-Benjamin.Cheatham@amd.com/>

[1] CXL 3.1, 9.9 Hotplug



Discussion, Feedback, and Q&A





Linux Plumbers Conference

Vienna, Austria | September 18-20, 2024



CXL RAS Page offline for Corrected Errors

Srinivasulu Thanneeru(Micron)

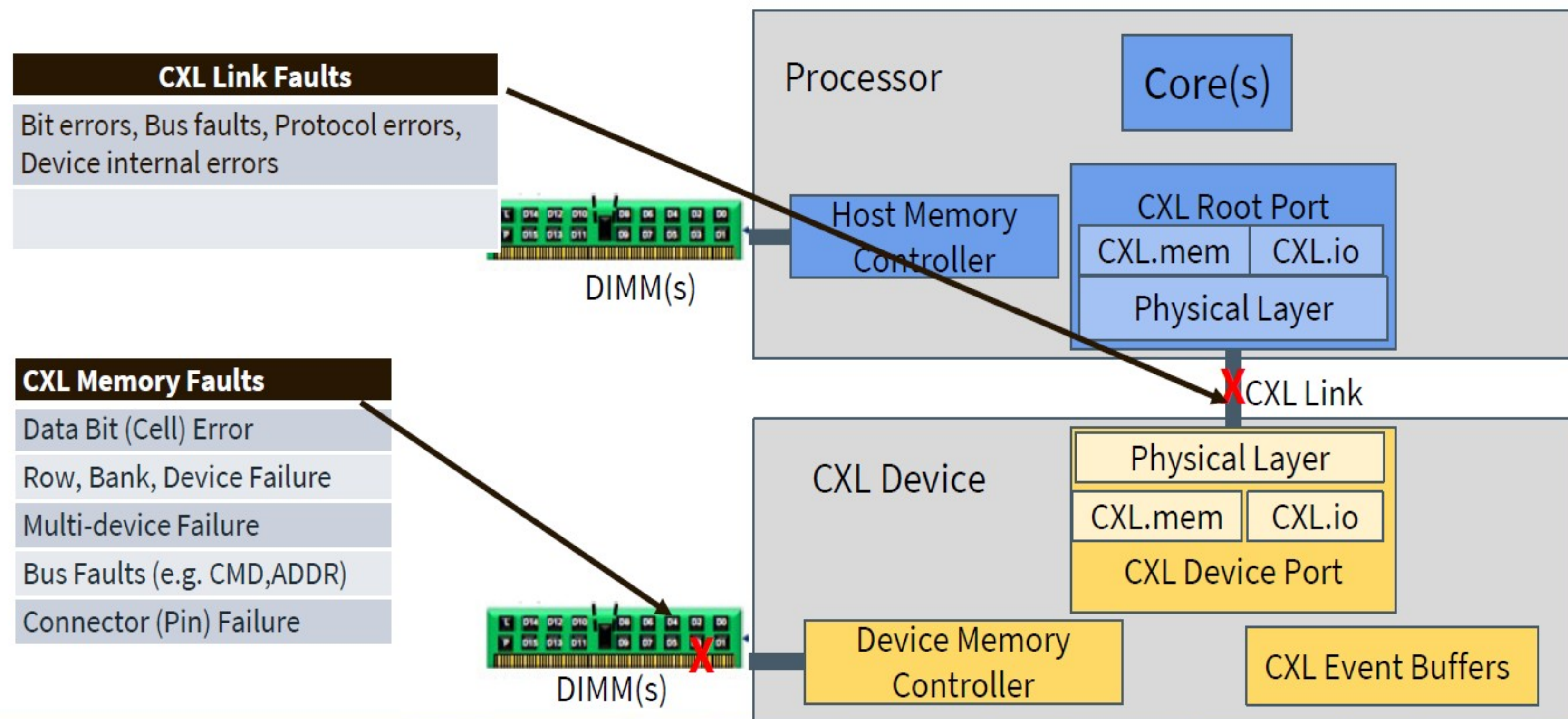


Agenda

- Current Status
 - Type-3 Device Fault Domains
 - Error Coverage
 - Error Reporting
 - DRAM Event Records Logs
- CXL Page offline support
 - Predictive Failure Algorithm (PFA)



CXL Type 3 - Fault Domains



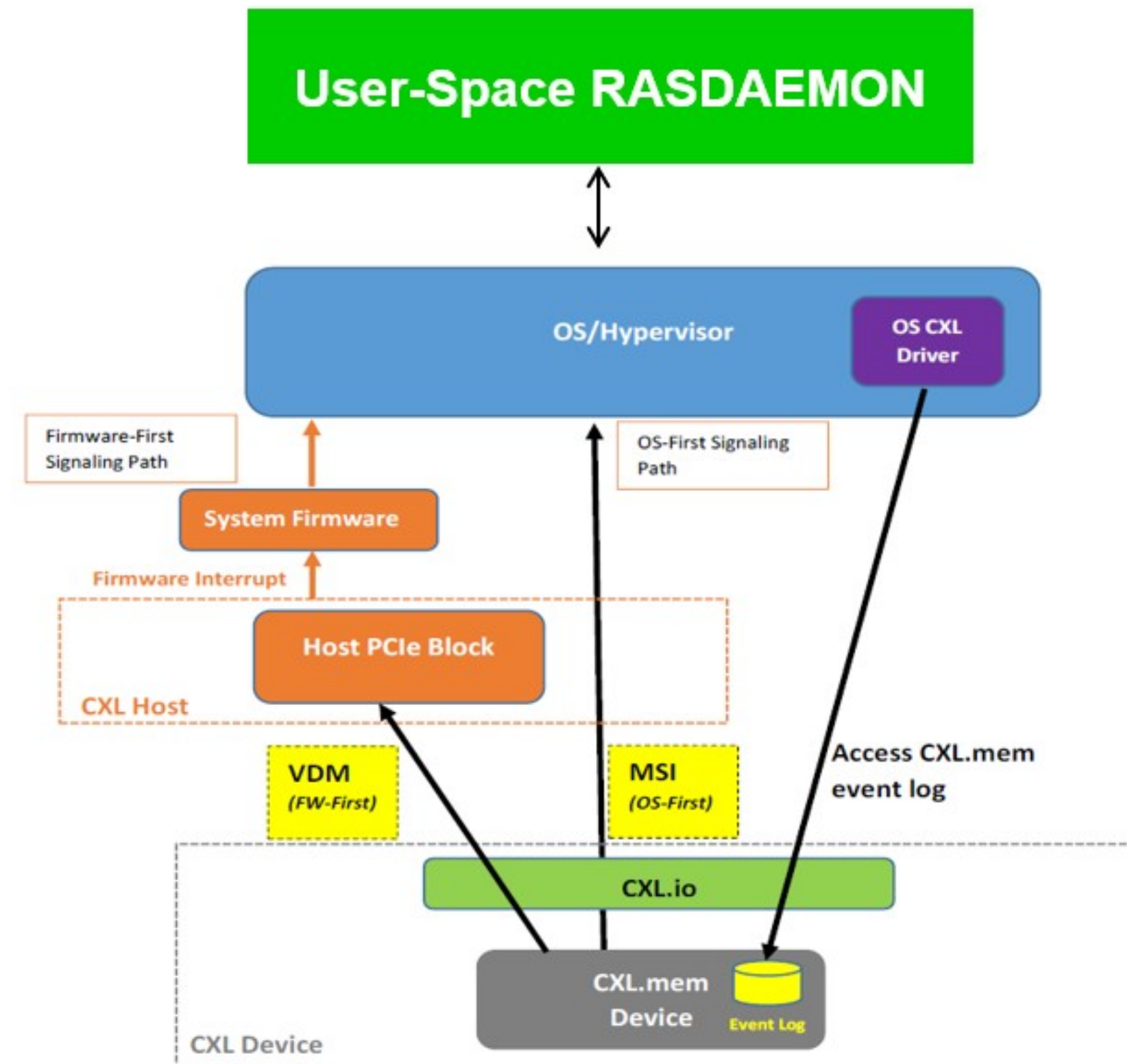
CXL Type 3 - Memory Faults Coverage

Possible Fault Type	Fault Causes (Examples)	Coverage (RAS Feature)
Data Bit (Cell) Error	High energy particle strike. Soft Error (SE). Transient error.	ECC, Demand Scrub, Patrol Scrub
Row Failure	Marginality. Persistent error	OS soft page offline, PPR, Chipkill
Bank Failure	Marginality. Persistent fault	Chipkill, OS soft page offline
Device Failure	Marginality. Persistent fault	Chipkill, OS soft page offline
Multi-device Failure	Persistent device failures	Contained by Poisoning (UCR) , MCA Recovery, OS hard page offline
ECC: Error Check and Correction RAS: Reliability, Availability, Serviceability		PPR: Post Package Repair UCR: Uncorrected Recoverable Error



CXL Type 3 - Memory CE Error Reporting Example

- **Type-3 Device Logs the CE event.**
- **Type-3 Device Triggers interrupt (OS-First) on Corrected Error when PFA threshold triggers.**
- **OS gets the event records and populate traces.**
- **RAS-Daemon captures DRAM event records.**



CXL Type 3 - Predictive Failure Algorithm (PFA)

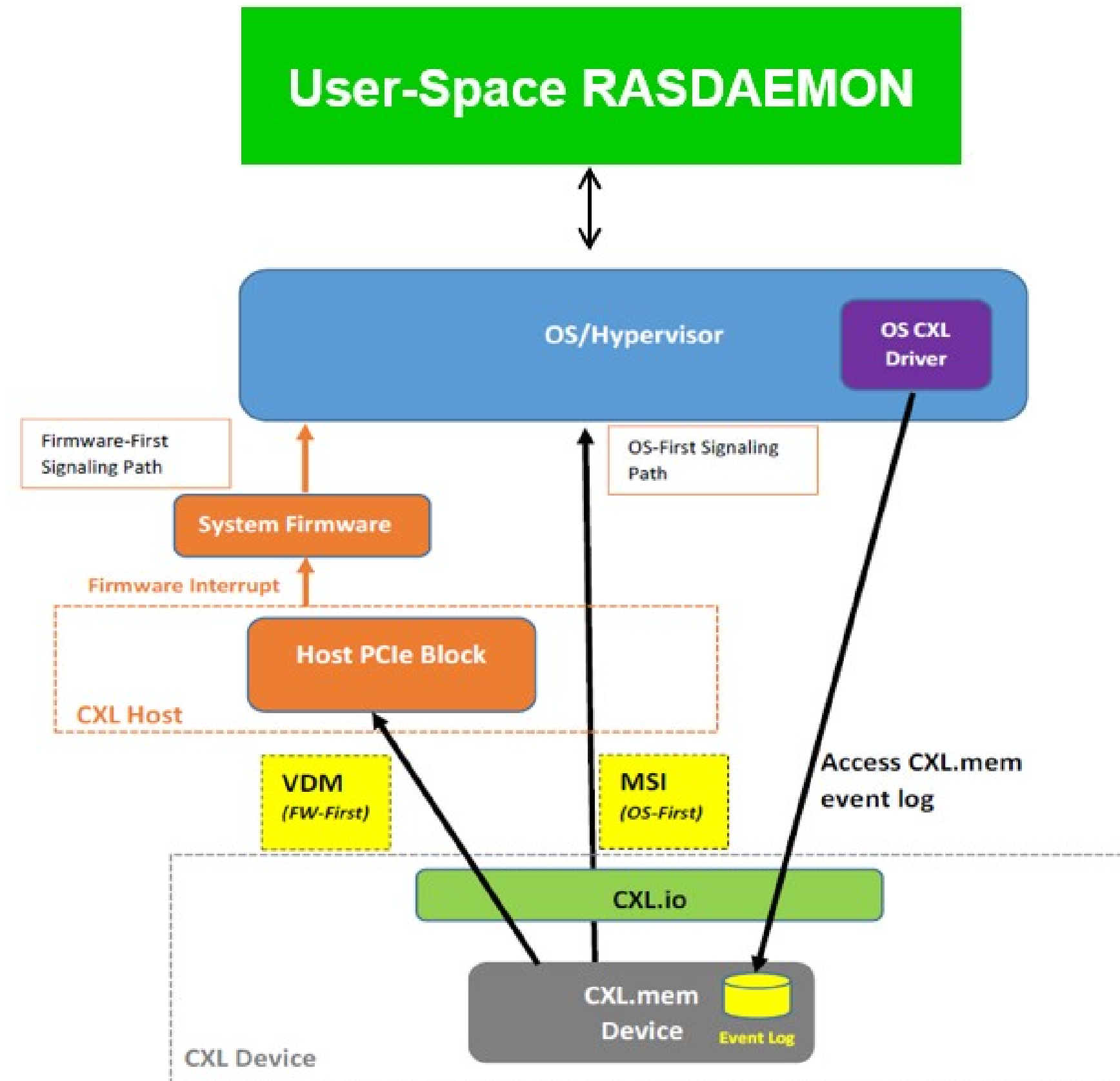
- **Devices implements an internal Correctable Error threshold or Predictive Failure Algorithm (PFA).**
- **Designed to assist the host in avoiding usage of memory locations that may degrade into an uncorrectable error.**
- **Device generates a DRAM Event Record with a Threshold Event descriptor and DPA pointing to the questionable memory location.**



CXL Type 3 - Page offline support

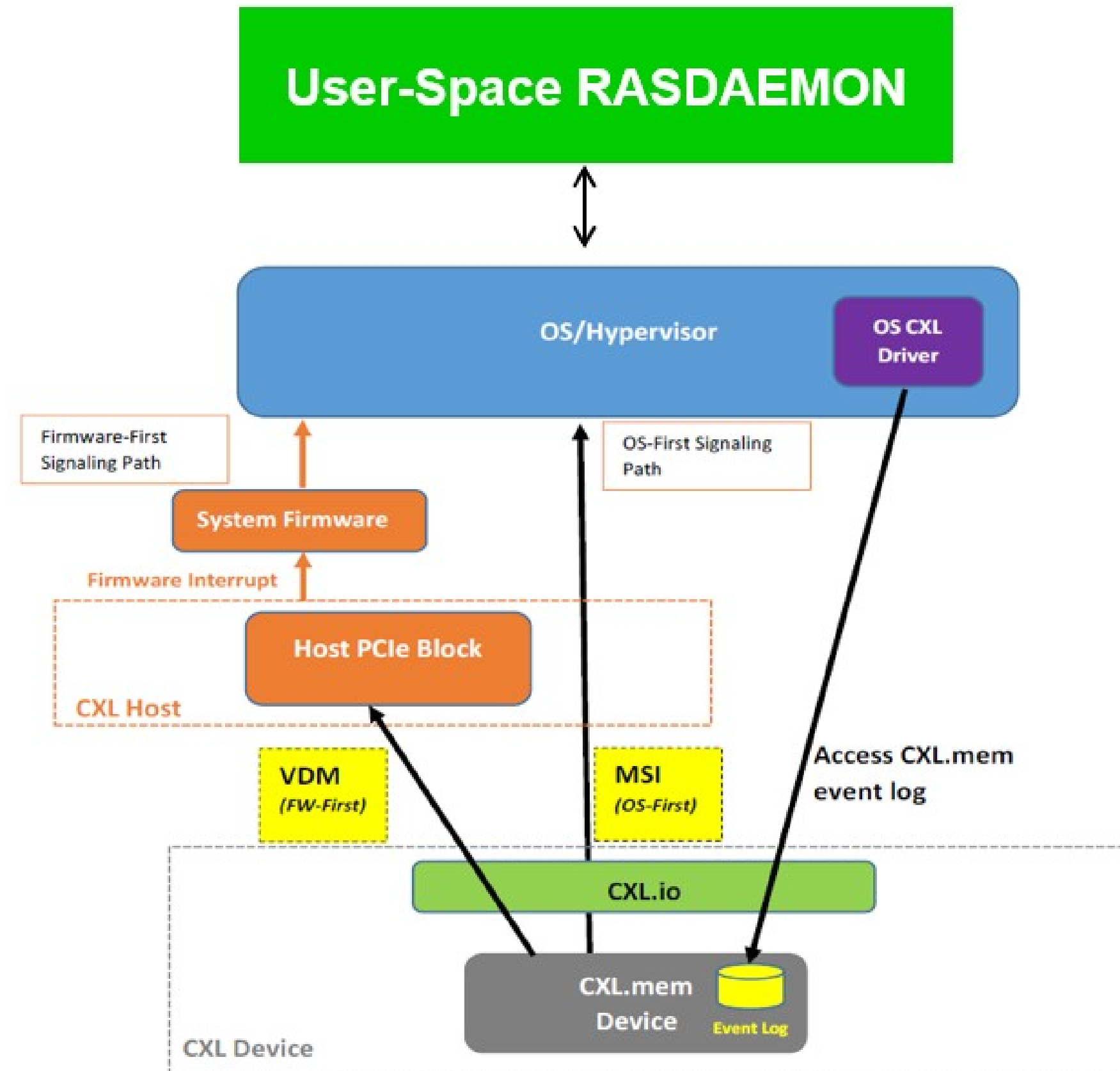
Example Event Record:

Event Record Name: **DRAM Event Record**
 Event Record Identifier: 601dccb3-9c06-4eab-b8af-4e9bfb5c9624
 Event Record Length: 128
 Event Record Flags: 0
 Event Record Handle: 0x8001
 Related Event Record Handle: 0x0
 Event Record Timestamp: 1717485377838306151 2024-06-04 07:16:17.838306
 Maintenance Operation Class: 0
 Event Record Reserved_2: 0
 Event Record Data:
Physical Address: 0x41
 Memory Event Descriptor: 0x0
Uncorrectable Event: 0
Threshold Event: 1
 Poison List Overflow Event: 0
 Memory Event Type: Media ECC Error
 Transaction Type: Internal Media Management
 Validity Flags: 0xff
 Memory Event Location:
 Channel: 2 (Valid: 1)
 Rank: 0 (Valid: 1)
 Bank Group: 0 (Valid: 1)
 Bank: 0 (Valid: 1)
 Row: 0x0 (Valid: 1)
 Column: 0x0 (Valid: 1)
 Nibble Mask: 0x4000 (Valid: 1)
 Correction Mask (Valid: 1)



CXL Type 3 - Page offline support

- Type-3 Device logs the CE event
- Type-3 Device Triggers interrupt (OS-First) on Corrected Error when PFA threshold triggers.
- OS gets the event records and populate traces
- RAS-Daemon captures DRAM event records
- **CE with threshold set, then do page-offline for DPA/HPA.**

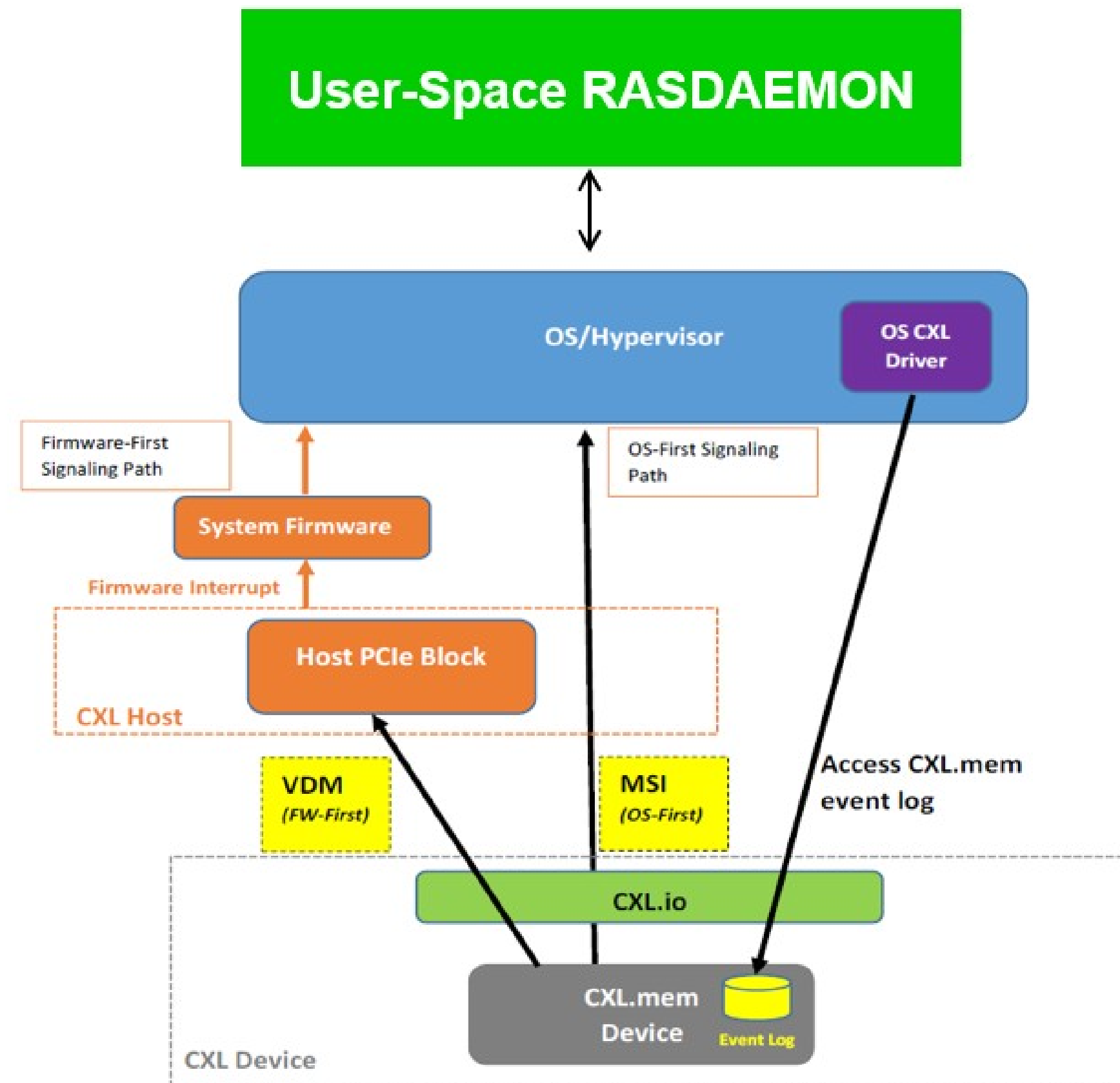


CXL Type 3 - Page offline support

```
diff --git a/ras-cxl-handler.c b/ras-cxl-handler.c
index 037c19c..04be770 100644
--- a/ras-cxl-handler.c
+++ b/ras-cxl-handler.c

+ if (tep_get_field_val(s, event, "hpa", record, &val, 1) < 0)
+     return -1;
+ ev.hpa = val;
+ if (trace_seq_printf(s, "hpa:0x%llx ", (unsigned long long)ev.hpa) <= 0)
+     return -1;
+
+ if (tep_get_field_val(s, event, "dpa_flags", record, &val, 1) < 0)
+     return -1;
+ ev.dpa_flags = val;
@@ -1005,6 +1017,12 @@ int ras_cxl_dram_event_handler(struct trace_seq *s,
    }
}

#ifdef HAVE_MEMORY_CE_PFA
+ /* Account page corrected errors */
+ if (!ev.descriptor.uncorreted && ev.descriptor.threshold )
+     ras_record_page_error(ev.hpa, PAGE_CE_THRESHOLD + 1, now);
#endif
+
```



CXL RAS Page offline for Corrected Errors

