

# CXL: Control Plane vs Data Plane

## *Software For Memory*

Presenter: Adam Manzanares | Director @ Samsung Electronics

# CXL Device Type and Protocols

CXL Device Type	Protocols	Example Use Case
Type 1	.io + .cache	Smart NIC
Type 2	.io + .cache +.mem	GPU/Accelerator
Type 3	.io + .mem	Memory expansion

.io protocol covers all use cases  
.io protocol is control plane

- .mem request routed through PCIe/CXL hierarchy via HW
- .io protocol used to establish routing
- Rest of presentation will focus on type3 device



# Tiering is interesting ... but!

- RAS is handled through .io
  - Poison handling
  - Scrubbing
  - Viral
- CXL vs DDR
  - Supports higher latencies
  - Can demand more software awareness of errors
- CXL Performance Monitoring
  - Controlled through .io
  - Hotness tracking proposed<sup>[1]</sup>

[1] <https://www.opencompute.org/documents/ocp-cms-hotness-tracking-requirements-white-paper-pdf-1>



# RAS Implications for Linux

- EDAC system in place
  - Error Detection and Correction
  - Targets memory controllers
- CXL moves memory controller functionality to the device
  - CXL vs EDAC for handling RAS
    - Layering implications interesting
      - Scrub<sup>[1]</sup>
      - Poison<sup>[2]</sup>
- CXL errors may come from multiple sources <sup>[3]</sup>
  - Devices
  - Path to devices

[1] <https://lore.kernel.org/linux-cxl/20240716150336.2042-1-shiju.jose@huawei.com/>

[2] <https://lore.kernel.org/linux-cxl/20240417075053.3273543-1-ruansy.fnst@fujitsu.com/>

[3] <https://lore.kernel.org/linux-cxl/20240617200411.1426554-1-terry.bowman@amd.com/>



# Summary

- CXL .mem protocol behaves like memory as we know it  
Hardware handles access, no software necessary
- CXL .io protocol is the control plane to CXL Devices  
Memory devices can leverage control plane for RAS, performance monitoring, ...
- CXL capabilities challenge existing software  
Layering and responsibility of control evolving  
*HW interaction with FW and OS*  
*OS subsystem interactions*

