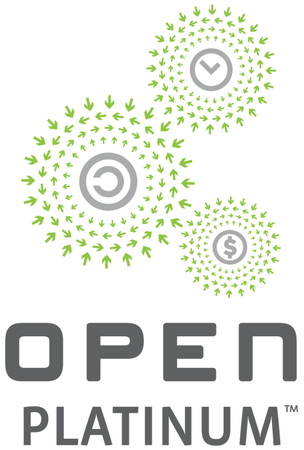

Open. Together.
OCP
SUMMIT

OPEN SYSTEM FIRMWARE

Open System Config Data: Firmware to Kernel

Sarathy Jayakumar, Principal Engineer, Intel Corp
Natarajan Sivagar, Software Engineer, Google Inc



What are we trying to solve

OPEN SYSTEM FIRMWARE

- A rich set of system data that is generated by the BIOS. Examples..
  - DDR Rank Margining Information
  - DDR Post Package Repair information
  - PCIe Link/Port Margining, Topology etc.
  - eMMC Bus Margining etc
  - Memory Map Topology
- No standard mechanism to expose this data
- Highly vendor specific. Proprietary mechanisms today

Specifications

How does OS use this data: Examples

OPEN SYSTEM FIRMWARE

- Holistic view aids Data Center operations in efficient decision making

Examples:

- Failure prediction based on Margining data and error rate
  - Assists in identifying systems that could become unreliable

- FRU Identification to enable faster servicing
  - E.g, Identify which DIMM might need to be replaced

- Margin Drift over time
  - Health of the FRU over time for planned servicing

Specifications

Mechanisms to Surface Platform Info

OPEN SYSTEM FIRMWARE

- Standard Table based mechanisms
  - SMBIOS, ACPI

- Standard Runtime Mechanisms
  - Platform Runtime Mechanism (Replaces certain categories of SMIs)
  - UEFI Runtime Services

- Proprietary Mechanisms (Focus of this Presentation)
  - **BDAT (BIOS Data ACPI Table)**
  - Exposes Platform Info in a Proprietary table and associated Schema

Specifications

Challenges with ACPI Tables today

OPEN SYSTEM FIRMWARE

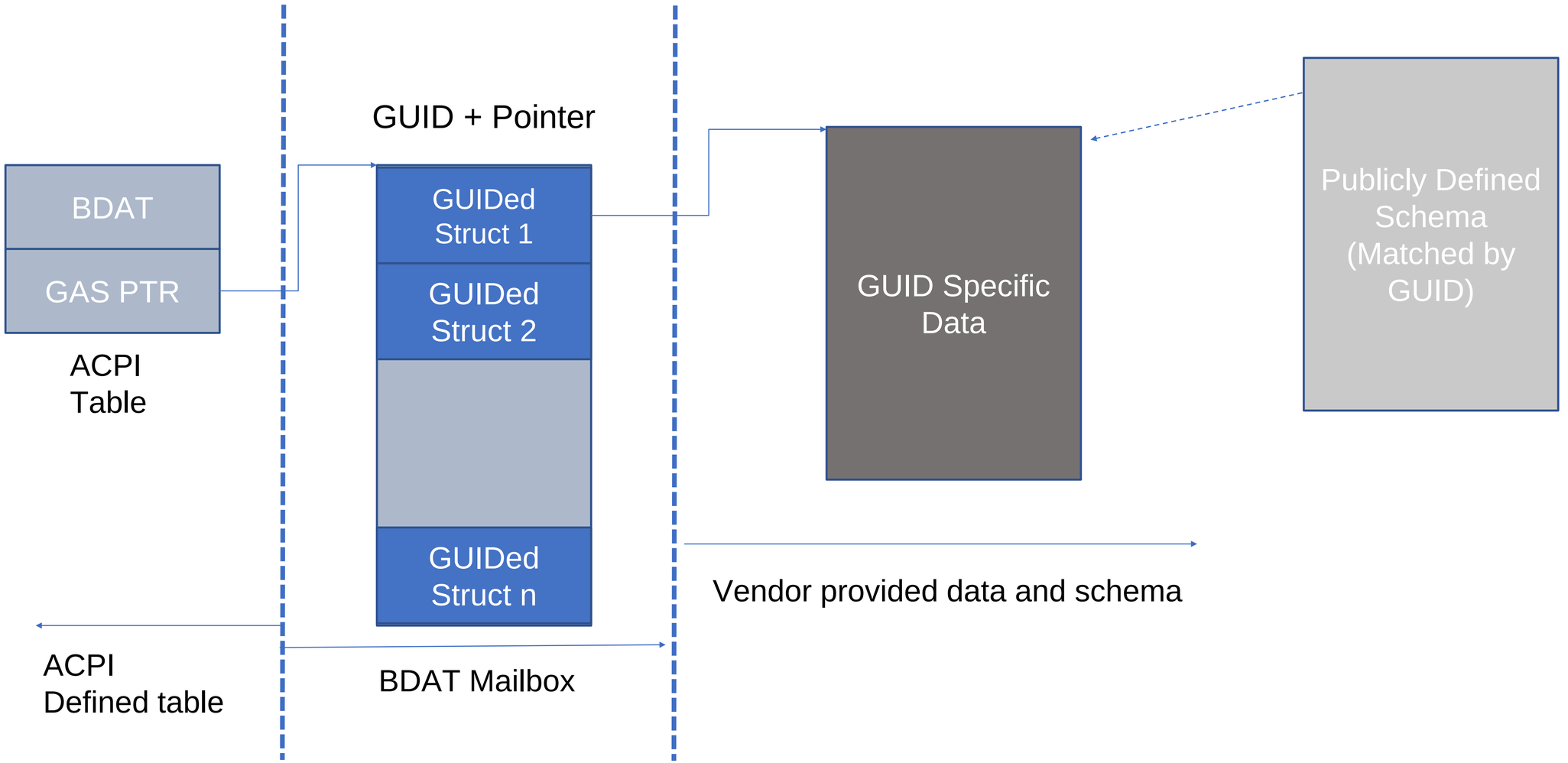
- **Addresses the lengthy and arduous process of deploying new technologies via ACPI**
  - Defined by a standards body
  - Updates or changes to this requires a lengthy process and needs to be signed off by the WG
  - Hence this affects deploying new technologies and features.

- **Seamless update ACPI tables/ data in runtime is not possible.**
  - OS restrictions and page table mapping and relocation of the ACPI table

- **Even if the above is addressed, still require a kernel soft reset for a new table to take effect**
  - This is highly disruptive to cloud workloads and affects 99.999% uptime for servers

Specifications

Making ACPI Tables Flexible and Extensible



OPEN SYSTEM FIRMWARE

Specifications

Schema brings flexibility and extensibility

OPEN SYSTEM FIRMWARE

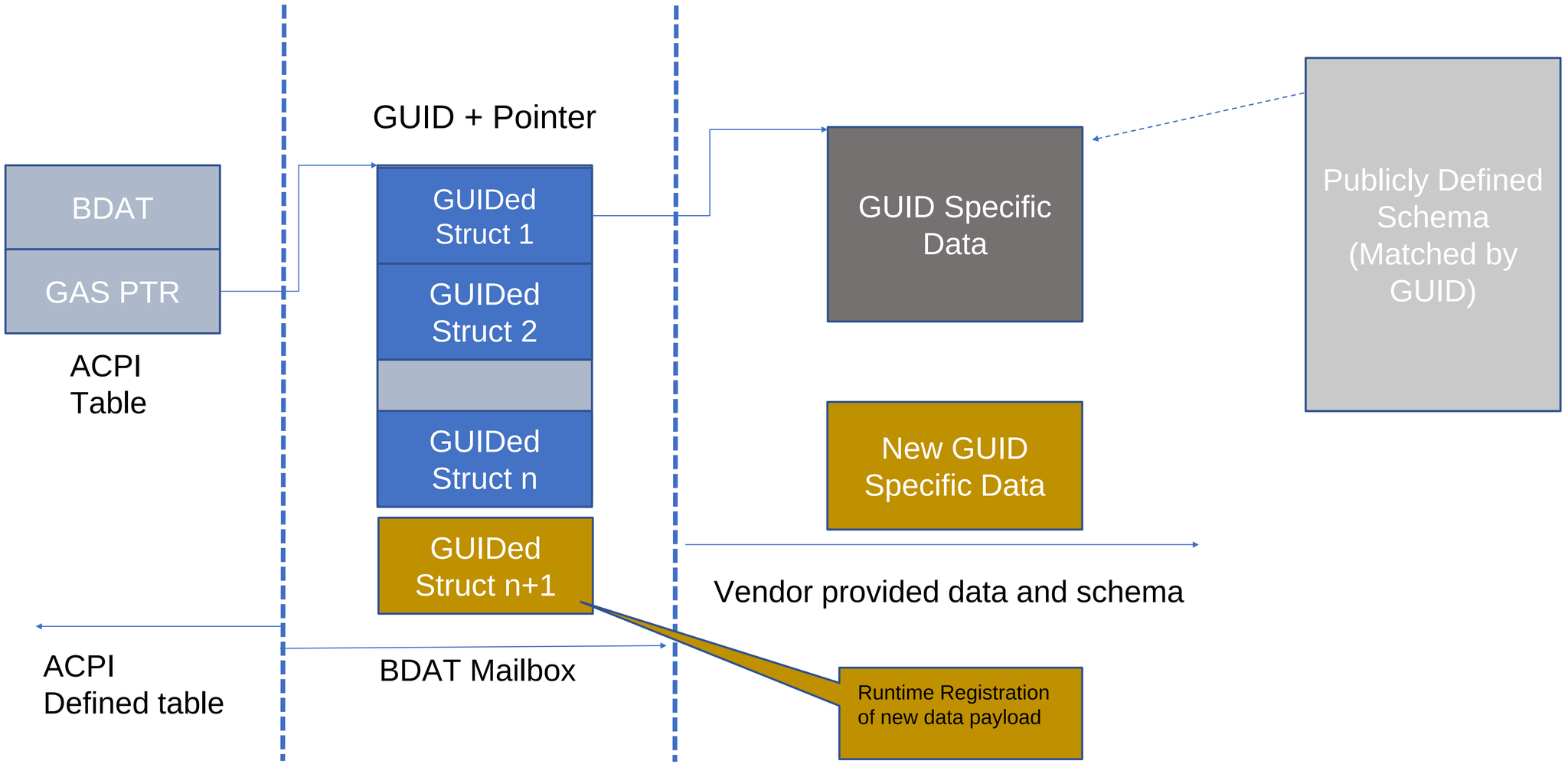
- Use richer Redfish / JSON Based schema
  - Describes the published data for interpretation
  - Flexibility – Vendor data in an Industry standard format

- Disconnects from ACPI Spec revisions
  - Allows for extensibility
  - New Schemas can be added without Spec updates

- Provides Richer Alternative for ACPI Name Space
  - E.g Can describe System Hierarchy

Specifications

Runtime Update



OPEN SYSTEM FIRMWARE

Specifications

OPEN SYSTEM FIRMWARE

Example Usage Case - Memory Rank Margining

Specifications

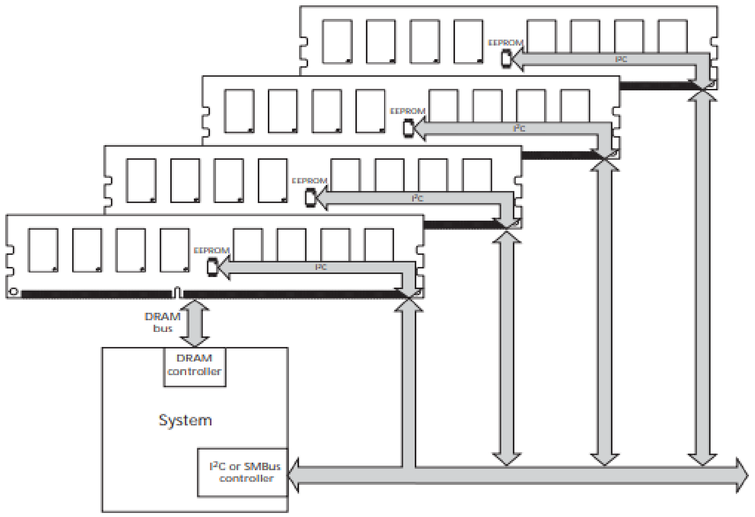
How BDAT Fills the gap with DIMM Schema

OPEN SYSTEM FIRMWARE

- Industry standard mechanisms expose platform config to the OS
  - SMBIOS Structures defined by DMTF
  - Various ACPI Tables and Methods
  - JEDEC Specifications on DIMM SPD Data
- The Gap
  - There is no standard mechanism to expose SPD Data, DRAM Timing and Training data
- BDAT
  - Publish vendor schemas to parse the info provided in the BDAT Table

Specifications

DIMM Serial Peripheral Detect (SPD) Info



- DIMM SPD data in non-volatile storage
- SPD data is accessed through SMBus
- SPD provides relevant info about the DIMM
  - Organization, supported timings, serial & product info, manufacturer, etc.
  - Proprietary maintenance data: Post Package Repair info, fused cell info etc.

- Data in SPD is useful during DIMM repair flow

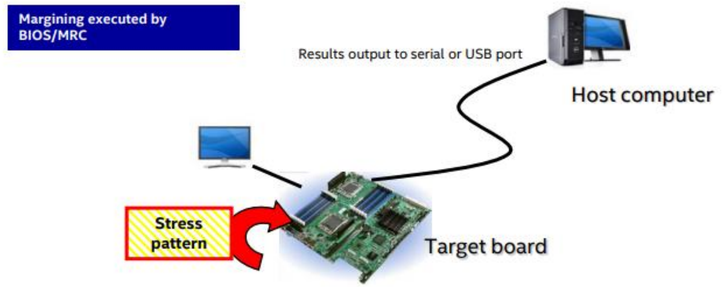
- Reading thru SMBus during runtime is not always feasible
  - Should be exposed via standard mechanisms

OPEN SYSTEM FIRMWARE

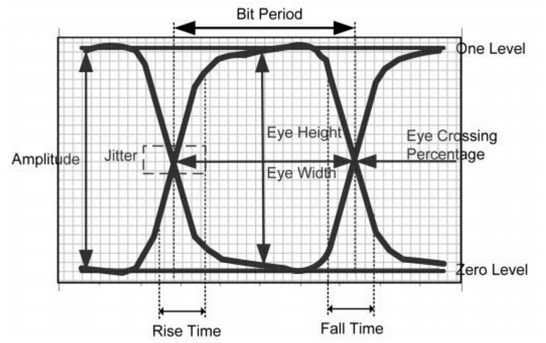
Specifications

DIMM Rank Margin Info

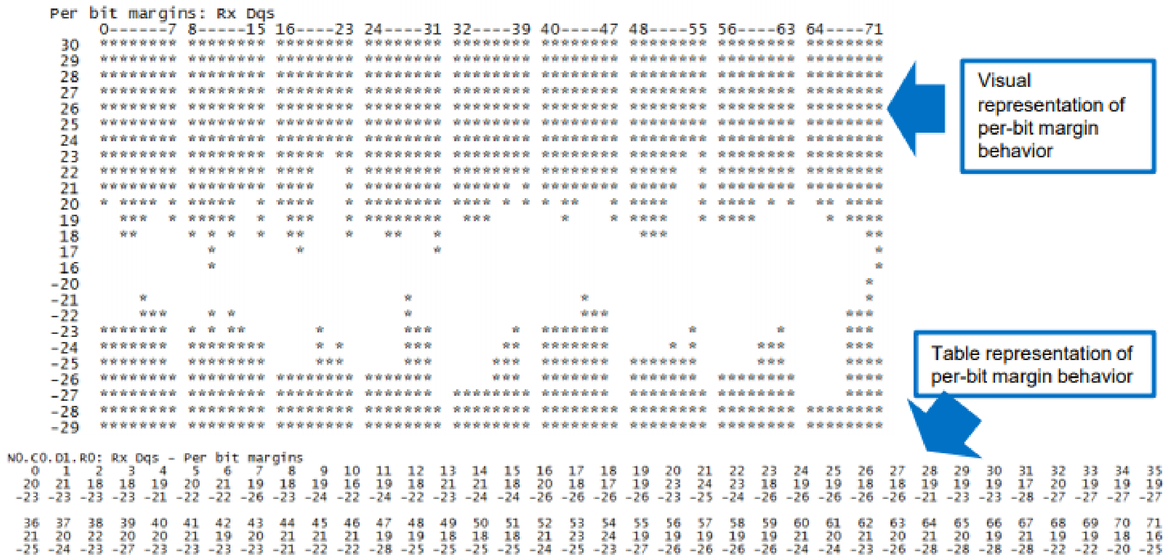
Rank Margining Tool



Rank Margining Tool provides automated Read/Write DQ and CMD margining



Per-Bit Margin Interpretation



- MRC trains the DRAM timings based on SPD data, OEM/Board configuration and margining algorithms
- MRC programs the memory controller with right timing values
- MRC disables DIMM/memory channels as required
- Should be exposed via standard mechanisms

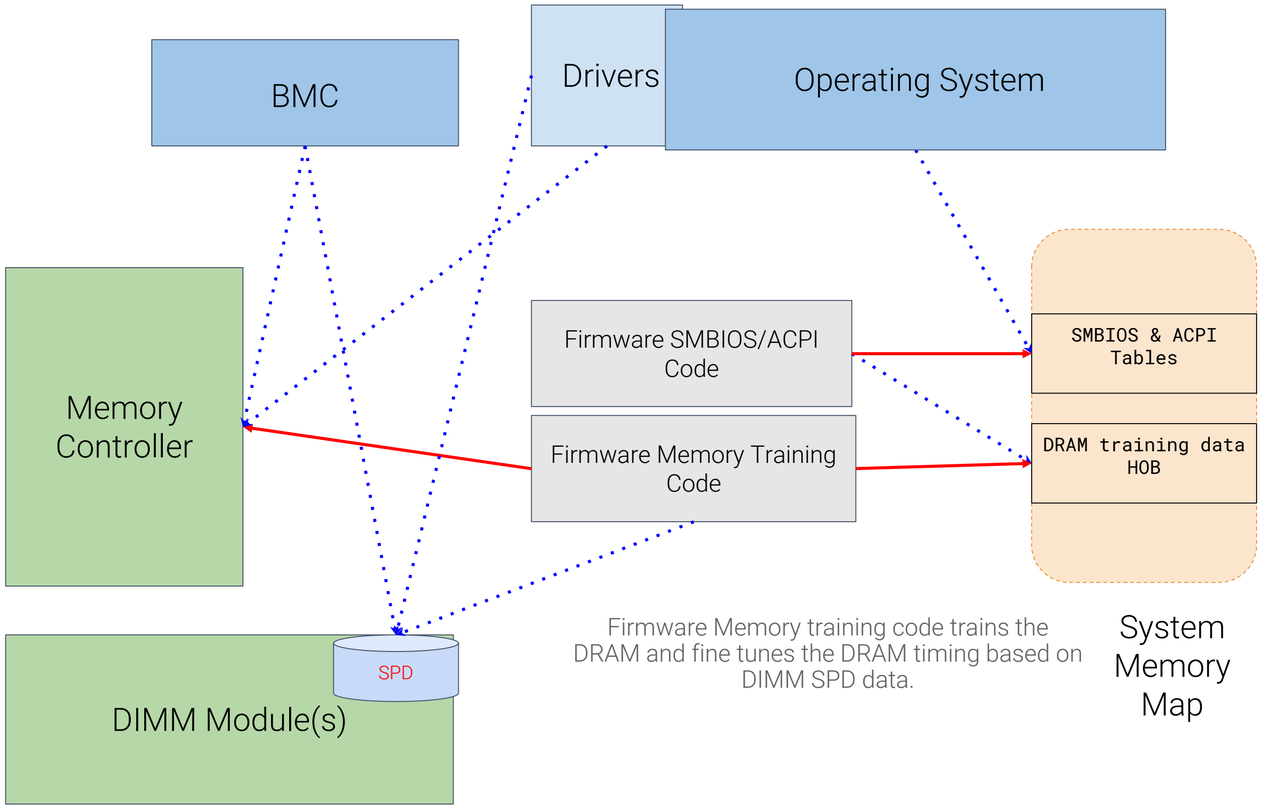
OPEN SYSTEM FIRMWARE

Specifications

DIMM Software Stack: OS & BMC

OPEN SYSTEM FIRMWARE

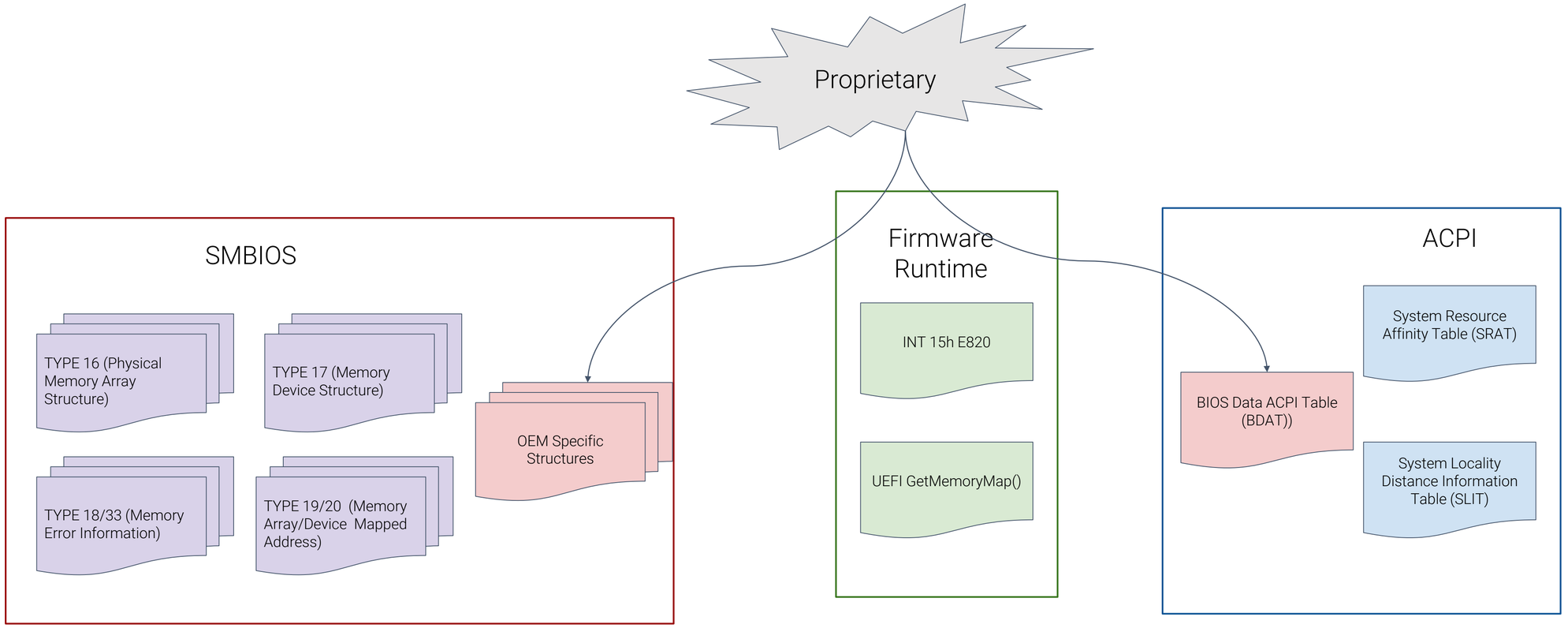
BMC thru inband or outband, queries DRAM data for FRU and sensor management.



- For system memory map the OS relies on standard, Firmware exported data like ACPI tables
- For repair workflow (for DIMM) and other timing info OS/BMC has to use proprietary methods even though underlying technology (DDR3/DDR4 etc) is standard

Specifications

Firmware Memory Data Interfaces



OPEN SYSTEM FIRMWARE

Specifications

Next Steps

OPEN SYSTEM FIRMWARE

Specifications

ACPI BDAT & Schemas

OPEN SYSTEM FIRMWARE

- Intel's BDAT (BIOS Data ACPI Table) provides a starting point to an open standard
  - Provides Standard ACPI Headers
  - Consists of vendor specific Schemas
- Schemas are basically UUID (GUID) based data structure
- The UUID identifies the type of data associated with the schema
- Multiple schemas will be arranged together to form a coherent data structure
- BDAT schemas provide flexibility in defining new data formats

Specifications

Proposed Open Firmware Data

OPEN SYSTEM FIRMWARE

- Intel is publishing the BDAT Table and associated schemas in public (link TBD)
- ACPI Initiatives in place to make it part of the ACPI Specifications
- Schemas are UUID identified and published in the open (not part of ACPI Spec)
- Develop a kernel driver (bdat) to expose BDAT schema
- Create necessary /sys nodes to access BDAT data in userspace

Specifications

Call to Action

OPEN SYSTEM FIRMWARE

- Work as a community to make it available in all platforms
- Provide feedback and add missing data
- Create relevant support in Open Source community like standard Linux drivers and tools
- Create additional interfaces like RedFish/IPMI to uniformly identify and utilize this data
- Contacts:
  - OSF Site (https://www.opencompute.org/projects/open-system-firmware)
  - OSF Mailing List (**OCP-OSF@OCP-All.groups.io**)
  - https://github.com/opencomputeproject/OSF

Specifications


Open. Together.
OCP
SUMMIT

SMBIOS Structures

OPEN SYSTEM FIRMWARE

- Memory device (Type 17) structure provide info on individual DIMM such as
  - DIMM dimensions size, width, speed, voltage etc
  - DIMM identifier such as location string, asset tag, serial number etc
  - Handle to error information structure (Type 18 or 33)
  - Handle to the physical memory array group
  - Usually not filled in in the DIMM is disabled due to training error
- Physical memory array (Type 16) structure groups memory devices to logical unit say a channel or memory riser etc.
- Memory array mapped address (Type 19) & memory device mapped address (Type 20) structures provide ability to translate memory address to DIMM
- As SMBIOS allows variable structure size, vendors use the extra data for storing proprietary info about the DIMM

Specifications

ACPI Tables

OPEN SYSTEM FIRMWARE

- Static Resource Affinity Table (SRAT)
  - Describes the resource affinity domains in the system
  - Critical structure to understand the Non-Unified Memory Architecture (NUMA) of the system
- System Locality Distance Information Table (SLIT)
  - Describes the locality of NUMA (Non-Unified Memory Architecture) nodes
  - The distance is given in terms of Memory Latency