HMAT Table Intro

NICOLE TRAPPE

Latency

Bandwidth

Cache

Main Memory

Disk

Project Stages

Stage I

✍️ Parse the HMAT table for **System Locality Latency & Bandwidth** data.

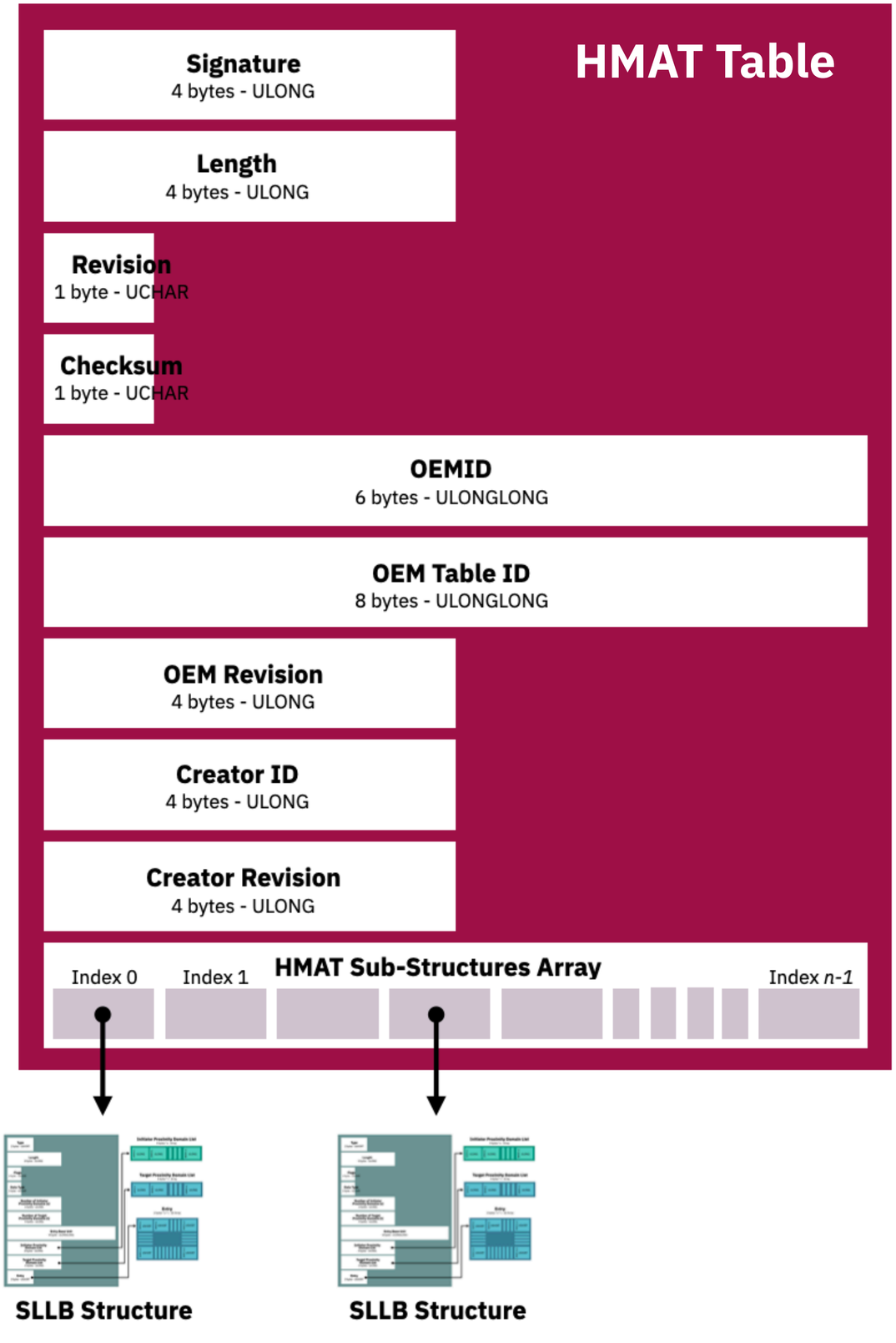
Stage II

🔨 Construct an API using I/O controls to expose HMAT data to users.

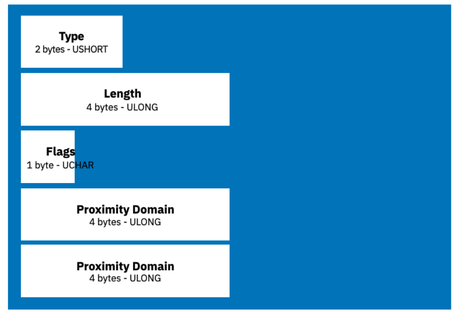
Stage III

📈 Write TAEF tests to fully flush out parsing code and I/O controls.

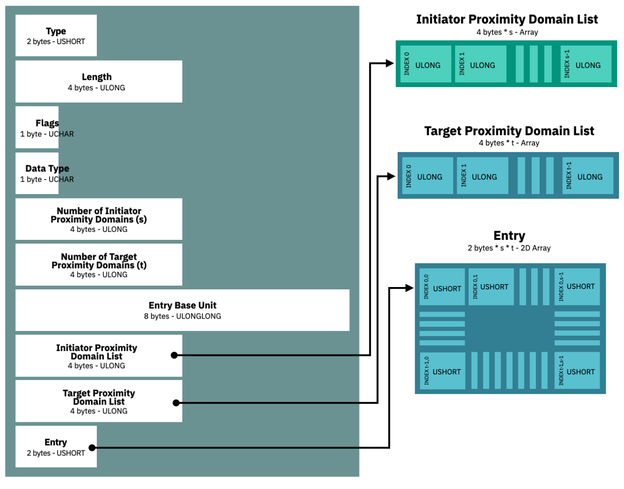
Stage I: Parsing the HMAT Table



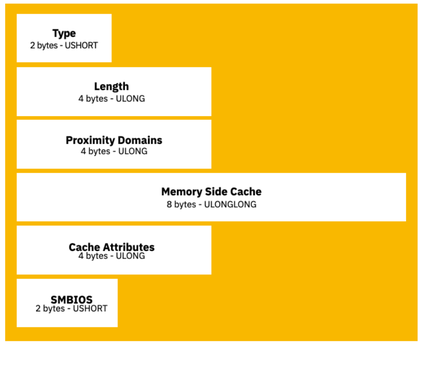
Memory Proximity Domain Attribute Structure (MPDA)



System Locality Latency & Bandwidth Structure (SLLB)

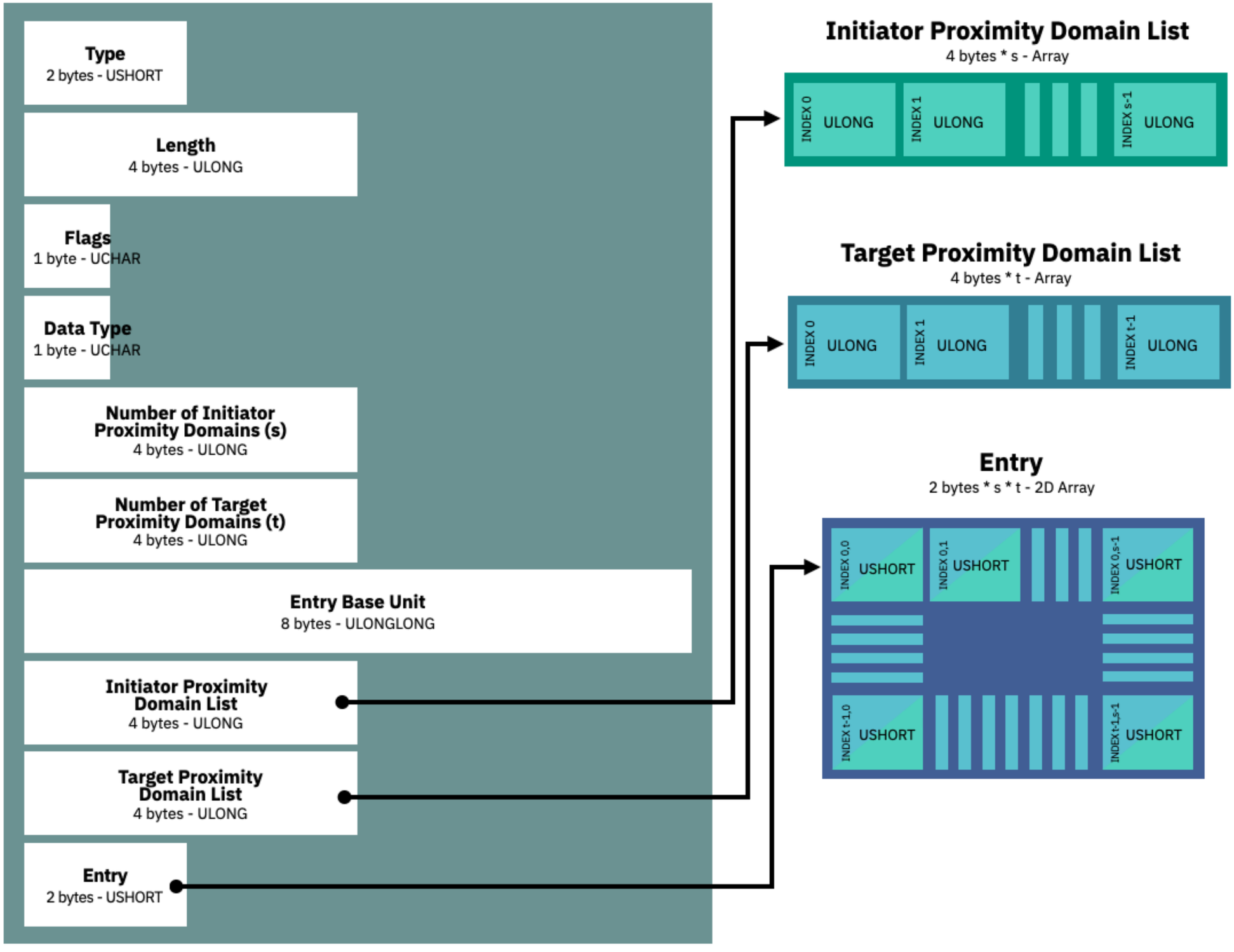


Memory Side Cache Information Structure (MSCI)



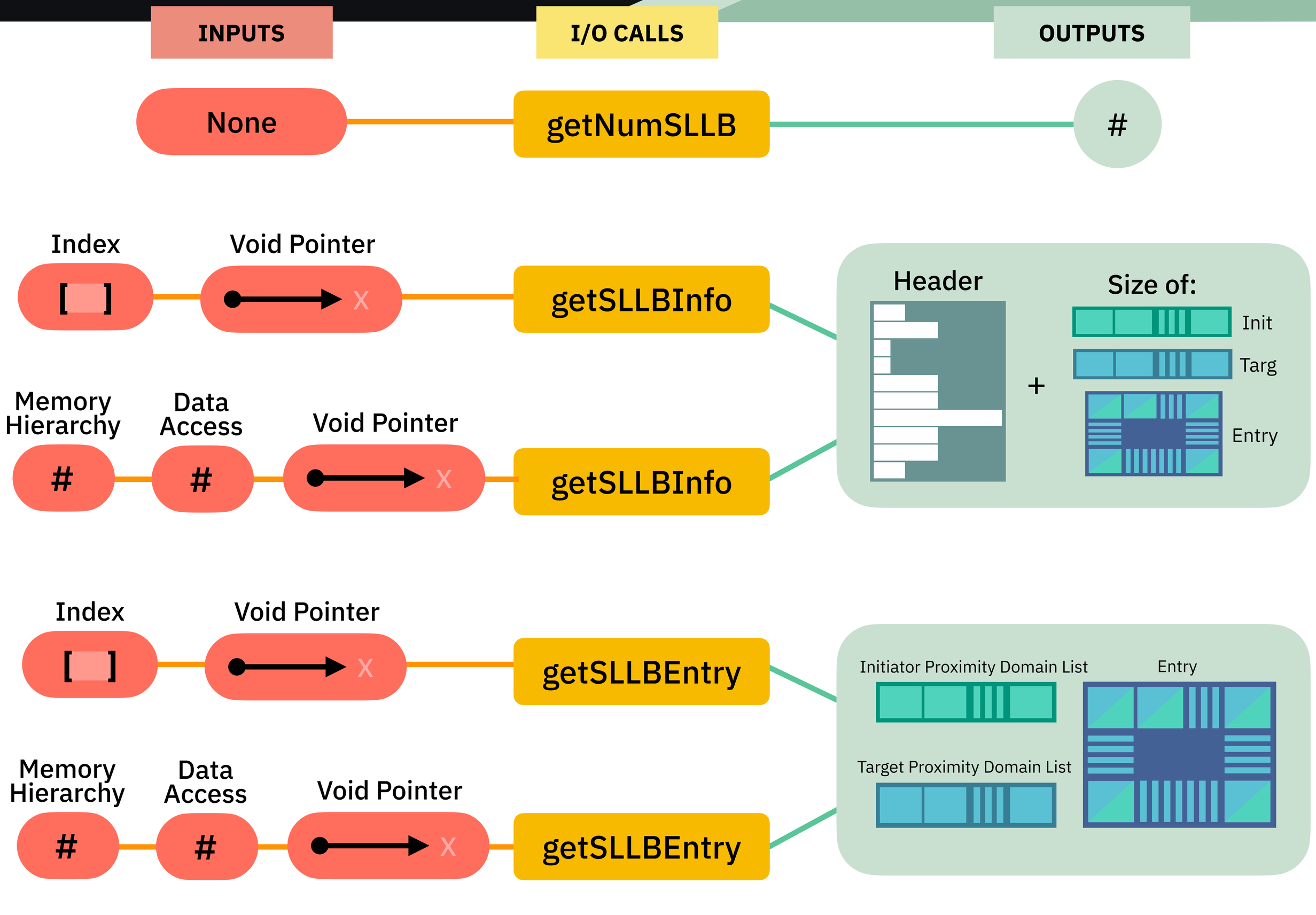
Stage I: Parsing the HMAT Table

System Locality Latency & Bandwidth Structure



NICOLE TRAPPE

Stage II: Constructing the API



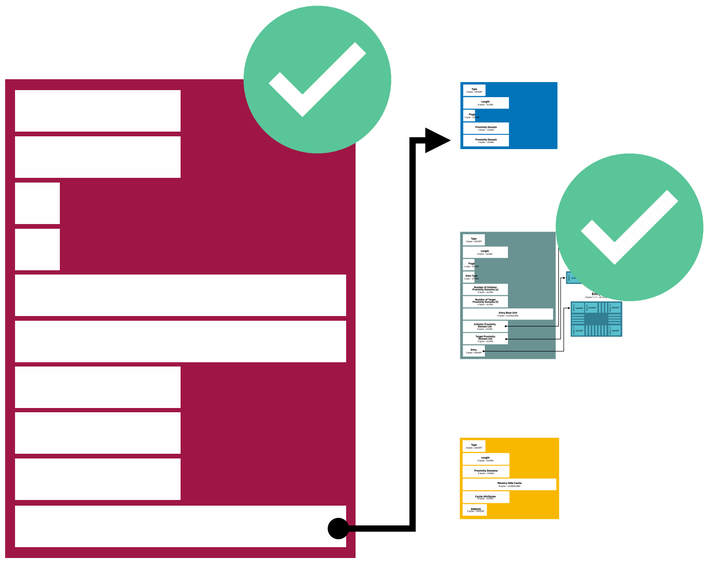
NICOLE TRAPPE

Stage III: Testing

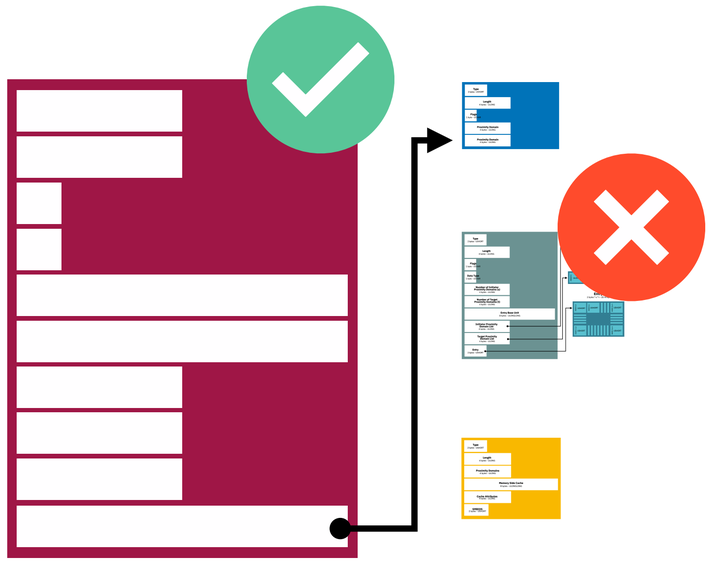
TAEF TESTS

Parsing Examples

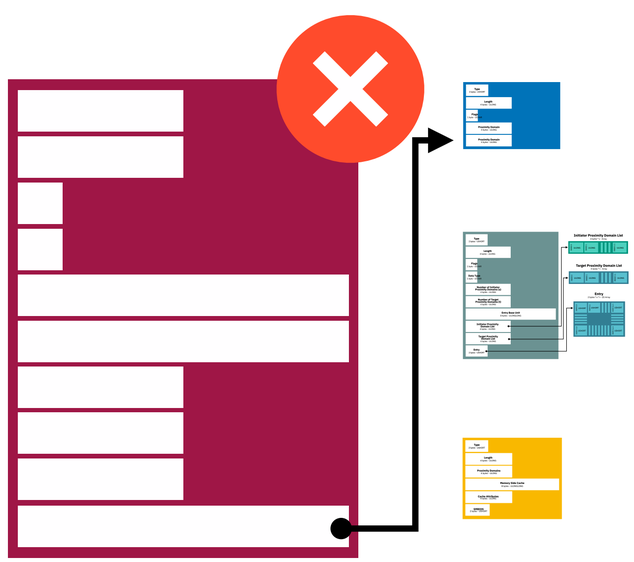
Valid Header & SLLB



Valid Header & no SLLB

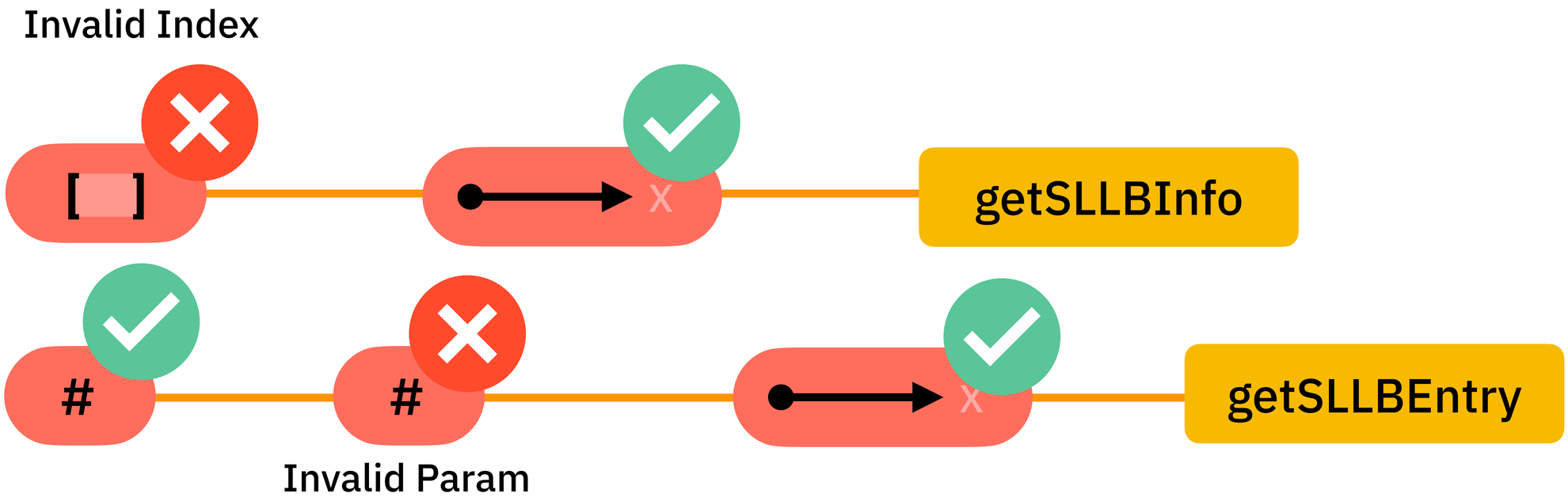


Invalid Header



Invalid data will be injected with **JSON** files

I/O Examples



Questions or comments before we move on to a few discussion questions?

Discussion Qs

Parsing HMAT

- Should we include the MPDA since that provides info needed for the Initiator/Target Domain?
- Data type to hold list of SLLB structures
  - Array of pointers: takes up more memory because some dead spaces BUT pro of O(1) access
  - Linked list: no wasted memory but search is O(n)
  - Should we dynamically allocate the entire HMAT table all at once?
    - Don’t want static (even header) because will take up memory even if code never used
    - Don’t think we want to malloc then malloc then malloc again as we read
    - Better: pass through and find out size & # of SLLBs then allocate a block of memory for whole HMAT with SLLBs
  - **Any questions about parsing the HMAT table?**

Discussion Qs

I/O Controls

- Should we include the MPDA since that provides info needed for the Initiator/Target Domain?
- What would a user of our API want?
  - “Read” information across the SLLBs
  - Specific memory hierarchy data
  - All data?
    - Do they just want the Initiator/Target/Entry info or also the header?
    - Should we provide the header data?
    - Should we provide a count for the # of SLLBs, etc.
- Do we expect the user to know the memory hierarchy and data operation (e.g. read)?
- Entry base unit (in header) is a multiplier for the unit of latency/bandwidth init/targ combinator
  - Do we provide the entry base unit OR do we do the calculations to then provide bandwidth/latency in Mbs or picoseconds?
- **Any questions about the API?**