NVMe Telemetry & Open Source Readiness

Presenter:

Ross Stenfort, Hardware System Engineer, Meta

Michael Allison, Senior Director, Samsung Electronics

Karthik Balan, Associate Director, Samsung Electronics

Agenda

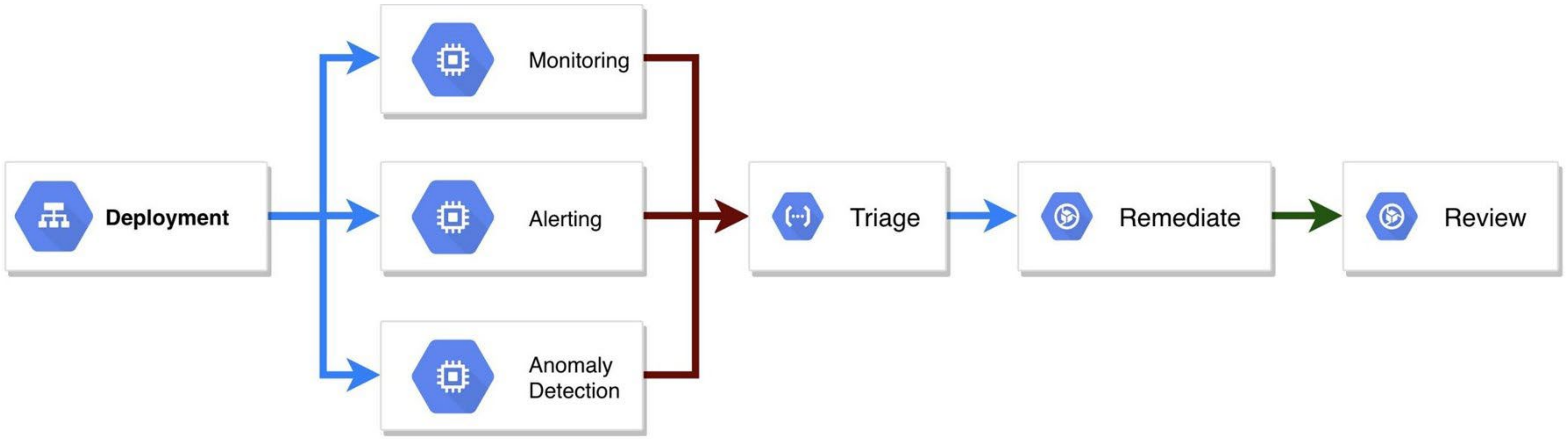
- History of SSD Telemetry and Debug
- Improved Telemetry Methodology
- Specification Design & Validation
- Enablement of Open-Source Tools
- Concurrent Readiness of Telemetry Feature

History of Telemetry and Debug

- ❖ Traditional Debug Methods:
  - SMART Logs or Vendor Unique Logs
  - Send Encrypted Telemetry Logs to supplier
    - Note: Customers with security concerns do not allow this

- ❖ Improved Methods: (Defined in OCP Datacenter NVMe® Spec):
  - Health Information Log (V1.0)
    - o SSD Statistics for monitoring based on deployment at scale
  - Latency Monitor (V2.0)
    - o Isolates performance spikes and enables debug at scale with live traffic
  - Formatted Telemetry (V2.5)
    - o Enables Flexible Human Readable Telemetry at Scale

How Debug is done today



Data Collection at Scale

Periodic Collection - At Scale (Automated)

SMART Attributes

DMESG Logs

SMART Cloud Health Log (0xCO)
(Based on NVMe DC SSD V1.0)

- NAND Statistics
- PCIe Statistics
- Health Statistics

Latency Monitoring Log
(based on OCP NVMe DC SSD V2.0)

Vendor & model independent logs can be captured efficiently by automation

Non- Periodic – Individual Drive (Manual)

Telemetry Logs

Drive Event Logs

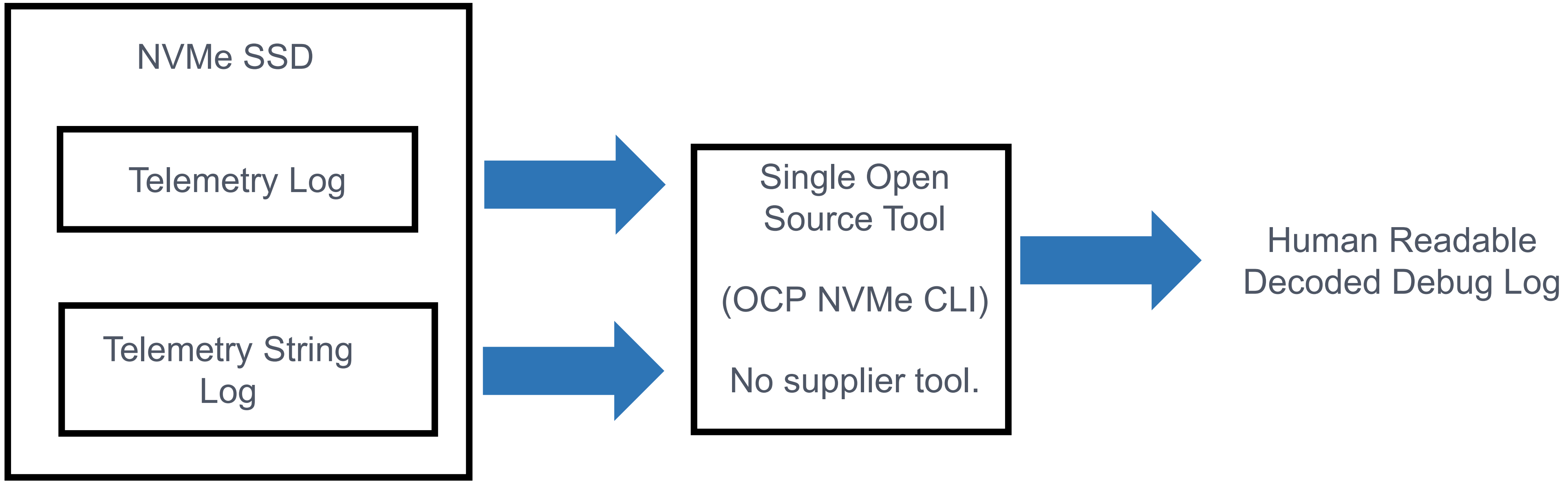
More Drives => Requires More Resources

Telemetry and Drive Event Logs do not Scale

<u>More Drives Require More Resources</u>

What is the solution?
OCP Structured Telemetry/Debug Log

- ❖ Standardized Structured Telemetry/Debug Logs
- ❖ All suppliers use same format for telemetry/debug information
  - Enables Human Readable Logs
  - Enables Open-Source Tooling



Problem

❖ **Theory:**

Specification Draft:

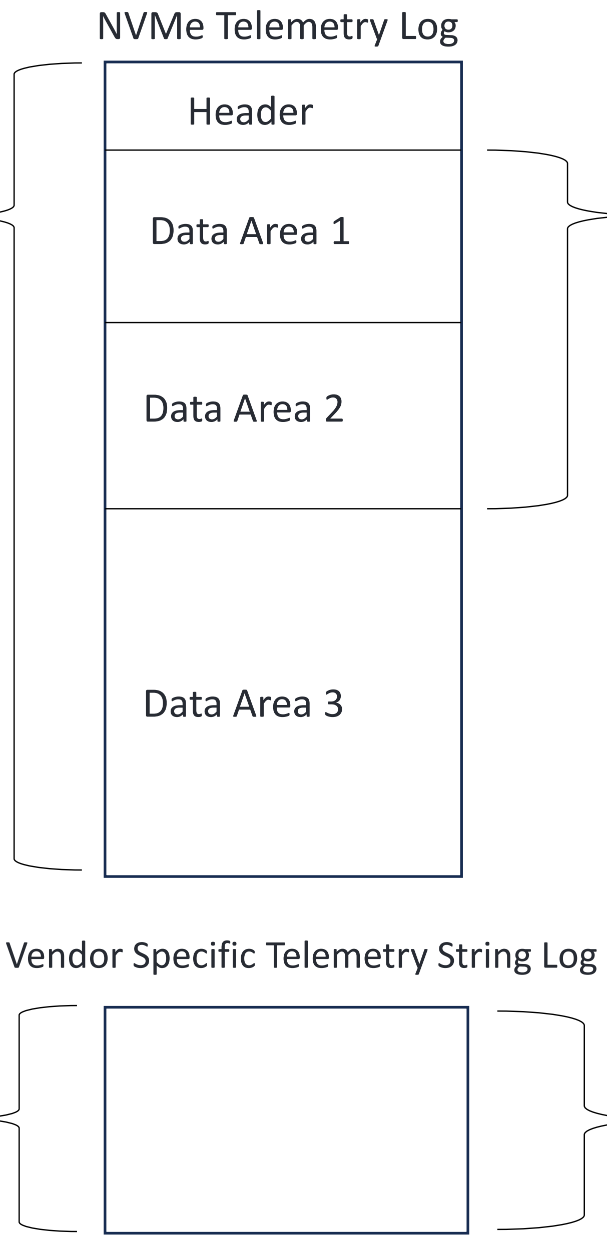
~38 pages of detailed data structures

Question:

How do I know the specification functionality works as expected without POC Hardware?

Defined in

NVM Express Based Specification 2.0



Defined by [OCP Datacenter NVMe SSD Specification V2.5]

Contains:

- FIFOs with Events
  - Specification Defined
  - Vendor Defined
- Counters
  - Specification Defined
  - Vendor Defined
- Overlayed Debug Logs
  - SMART / Heath Log
  - SMART / Heath Information Extended Log

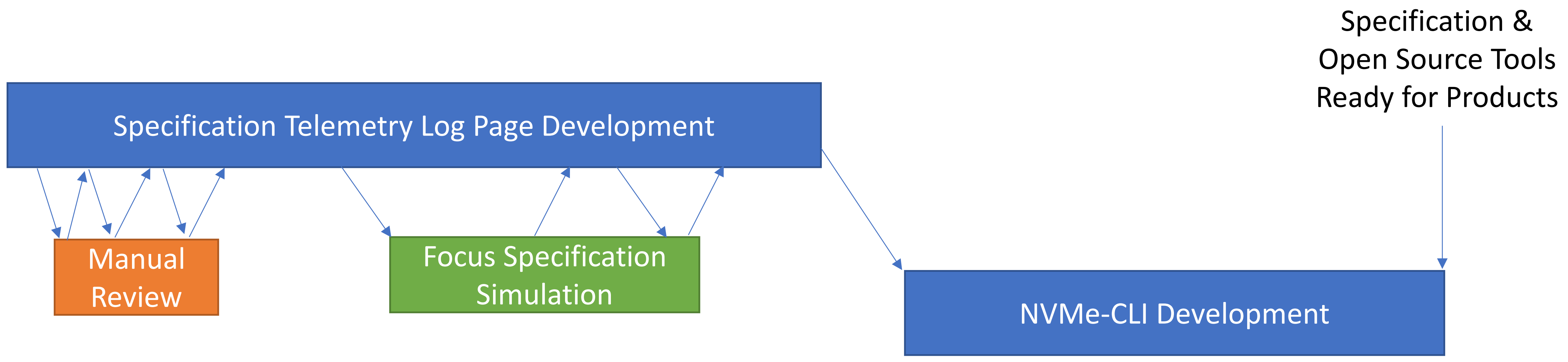
Defined in

OCP NVMe Datacenter SSD Specification V2.5

Contains:

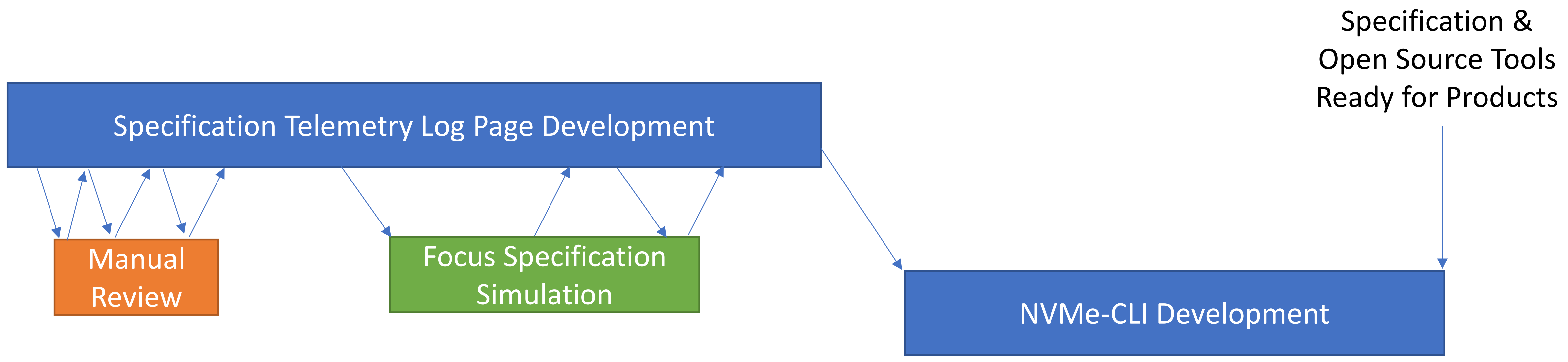
- Data Structures to map Data Area 1 and Data Area 2 to ASCII Strings

Answer: Focused Simulation



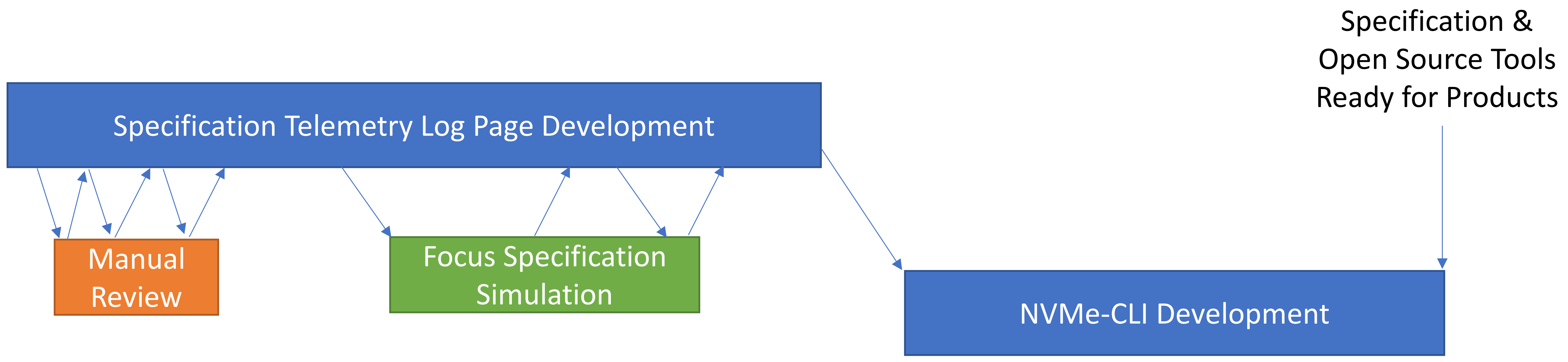
- Developed scripts dedicated to validate the OCP Datacenter NVMe™ SSD Specification Telemetry log page
  - 1st focus created a script to generate:
    - Telemetry log pages as defined by NVM Express® Base Specification
      - Controller-Initiated and Host-Initiated
      - Data Areas 1 & 2 as specified by the OCP Datacenter NVMe™ SSD Specification
        - Statistics
        - FIFOs
      - Data Areas 3 & 4
    - Create a Strings Log page with strings that identify itself

Answer: Focused Simulation



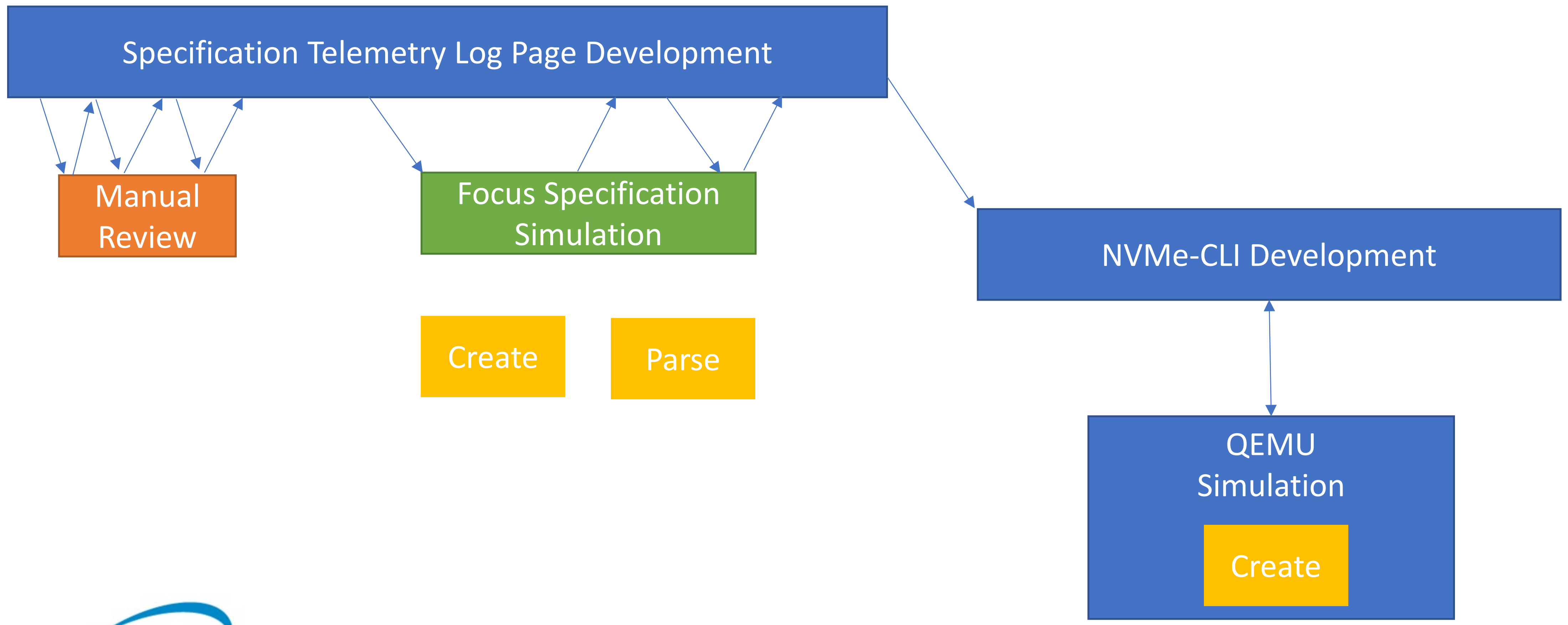
- Developed scripts dedicated to validate the OCP Datacenter NVMe™ SSD Specification Telemetry log page
  - 1st focus created a script to generate:
  - 2nd focus was to deal with the flexibility of the OCP Specification
    - Need to produce fix data
    - Needed to produce random data
    - Required to be repeatable
    - Needed to allow user to specify sizes, fixed data, types of random data

Answer: Focused Simulation

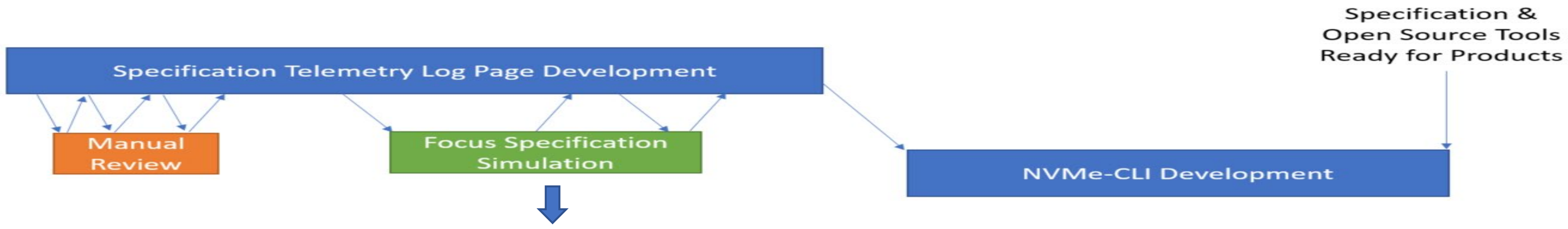


- Developed scripts dedicated to validate the OCP Datacenter NVMe™ SSD Specification Telemetry log page
  - 1st focus created a script to generate:
  - 2nd focus was to deal with the flexibility of the OCP Specification
  - 3rd focus parsed a Telemetry log page as defined by NVM Express® Base Specification

Answer: Focused Simulation


Specification Telemetry Log Page Development
Manual Review
Focus Specification Simulation
Create
Parse
NVMe-CLI Development
QEMU Simulation
Create

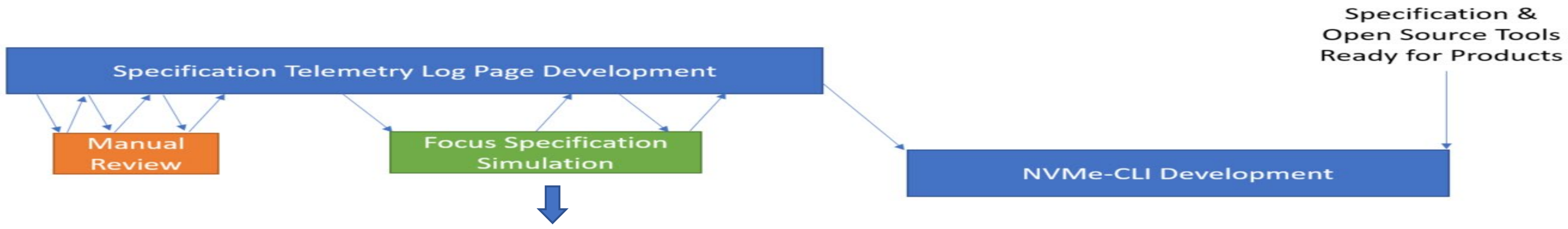
Enablement of Open Source Tools


Specification & Open Source Tools Ready for Products
Specification Telemetry Log Page Development
Manual Review
Focus Specification Simulation
NVMe-CLI Development

Samsung contributed to the OCP GitHub (link)

1. The script: **ocp_generate_nvme_telemetry_log.py**
    - Generates a NVMe™ Telemetry log page and a OCP Strings log page
        - Data Area 1 and Data Area 2 conforming to the OCP definition
        - Formatted data field with randomly generated values
        - Vendor defined Strings

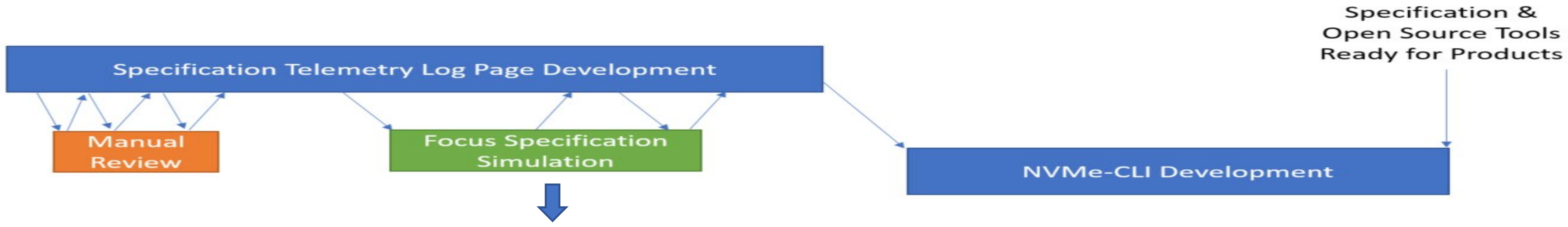
Enablement of Open Source Tools



Samsung contributed to the OCP GitHub (link)

1. The script: **ocp_generate_nvme_telemetry_log.py**
2. The script: **ocp_dump_nvme_telemetry_log.py**
   - Prints an NVMe Telemetry log page with the vendor defined

Enablement of Open Source Tools


Specification & Open Source Tools Ready for Products
Specification Telemetry Log Page Development
Manual Review
Focus Specification Simulation
NVMe-CLI Development

Samsung contributed to the OCP GitHub (link)

1. The script: **ocp_generate_nvme_telemetry_log.py**

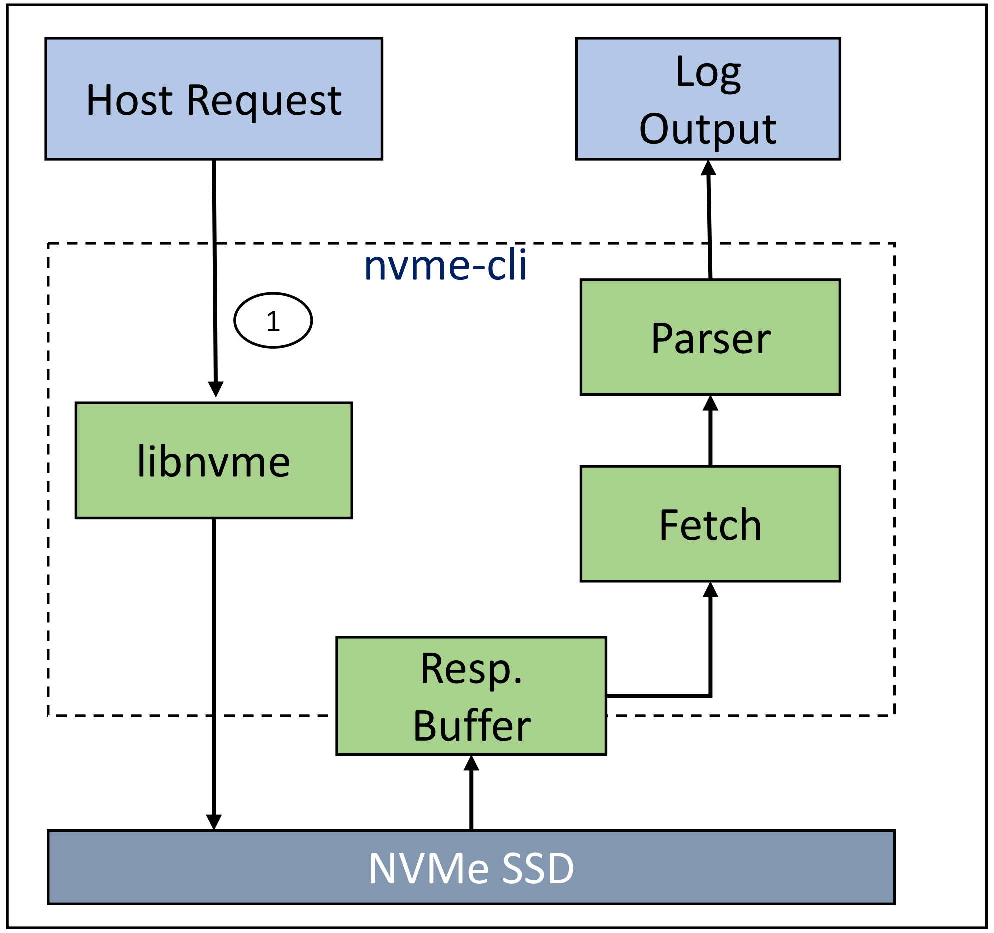
2. The script: **ocp_dump_nvme_telemetry_log.py**

Samsung updated nvme-cli to support the OCP Telemetry log page formats and is validated using this contribution

OCP Feature List in NVMe-CLI

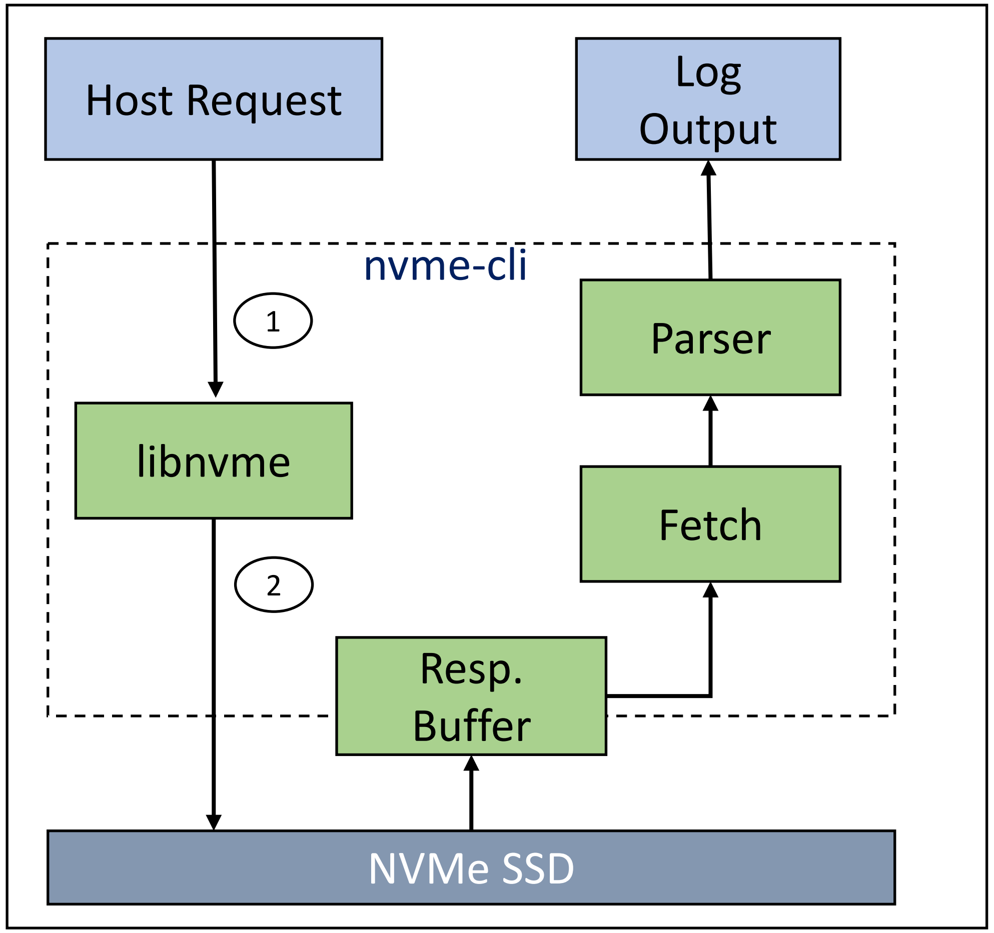
OCP Feature and Description	NVME-CLI Command Usage
SMART / Health Information Extended (Log Identifier C0h) • Retrieve extended SMART Information	'nvme ocp smart-add-log' <device> [--output-format=<fmt> | -o <fmt>]
Latency Monitor (Log Identifier C3h) • Get Latency Monitor Log Page	'nvme ocp latency-monitor-log' <device> [--output-format=<fmt> | -o <fmt>]
Latency Monitor (Feature Identifier C5h) • Set Latency Monitor feature	'nvme ocp set-latency-monitor-feature ' <device> [--active_threshold_a | -a][--save | -s]
Telemetry String Log (Log Identifier C9h) , Telemetry Log Page • Retrieve Telemetry string Log Page, telemetry log page	**Implemented by Samsung** and Command usage given in following slides
Clear Firmware Update History Feature (Feature Identifier C1h) • Clear firmware update history FID	'nvme ocp clear-fw-activate-history' <device> [--no-uuid | -n]
Unsupported Requirements (Log Identifier C5h) • Get Unsupported Requirements Log Page	'nvme ocp unsupported-reqs-log' <device> [--output-format=<fmt> | -o <fmt>]
DSSD Power State Requirements (Feature Identifier C7h) Get Device capabilities Requirements Log Page	'nvme ocp set-dssd-power-state-feature' <device> [--power-state=<fmt> | -p <fmt>] [--no-uuid | -n] [--save | -s]
PLP Health Check Interval (Feature Identifier C6) • Get/set PLP Health Check Interval	'nvme ocp get-plp-health-check-interval' <device> [--sel=<select> | -s <select>]
Device Capabilities (Log Identifier C4h) • Get Device capabilities Requirements Log Page	'nvme ocp device-capability-log' <device> [--output-format=<fmt> | -o <fmt>]
Error Recovery (Log Identifier C1h) • Get Device capabilities Requirements Log Page	'nvme ocp error-recovery-log' <device> [--output-format=<fmt> | -o <fmt>]
Clear PCIe Correctable Errors • Clear PCIe correctable error counters	'nvme ocp clear-pcie-correctable-error-counters' <device> [--no-uuid | -n]
EOL or PLP circuitry failure Mode • Define EOL or PLP circuitry failure mode.	'nvme ocp eol-plp-failure-mode' <device> [--mode=<mode> | -m <mode>] [--no-uuid | -n] [--save | -s] [--sel=<select > | -s <select>]

nvme-cli : DUT Response Buffer Parsing



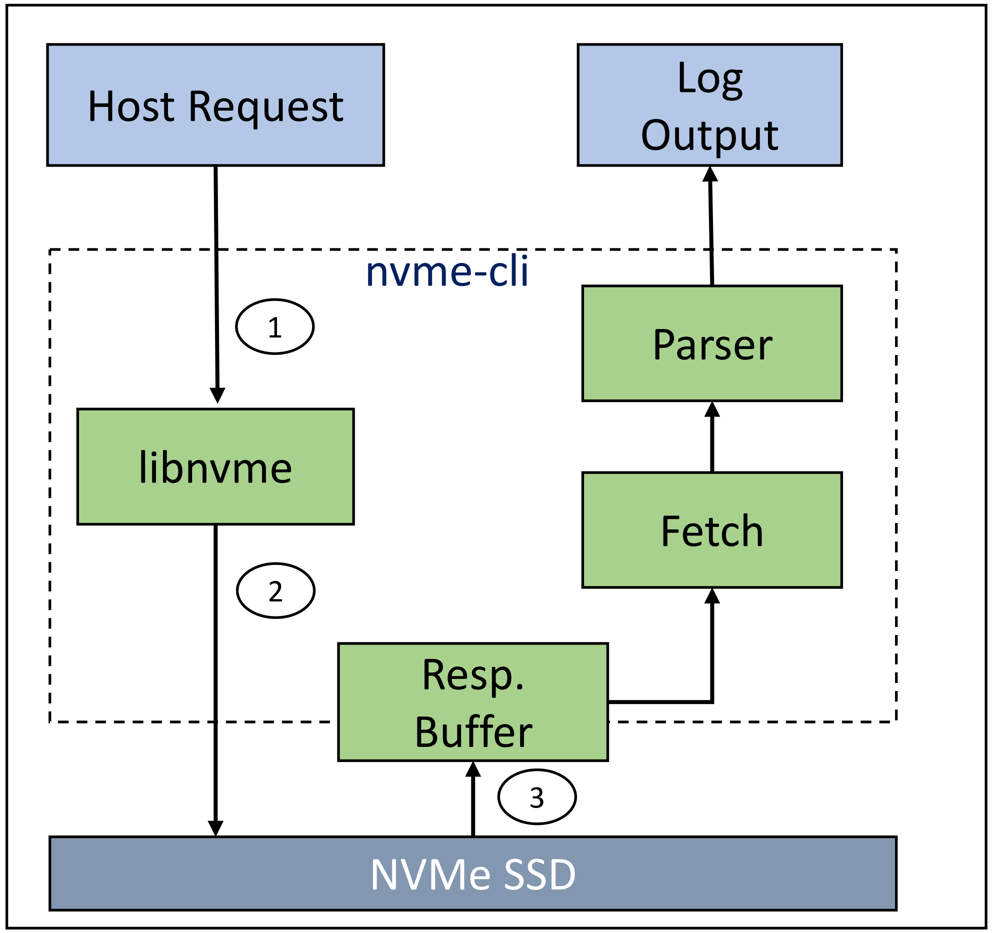
1. **Host Request via nvme-cli**
   - Telemetry Controller Initiated Log Page (LID : 08h)
   - Telemetry Host Initiated Log Page (LID : 07h)
   - Telemetry String log page (LID : C9h)

nvme-cli : DUT Response Buffer Parsing



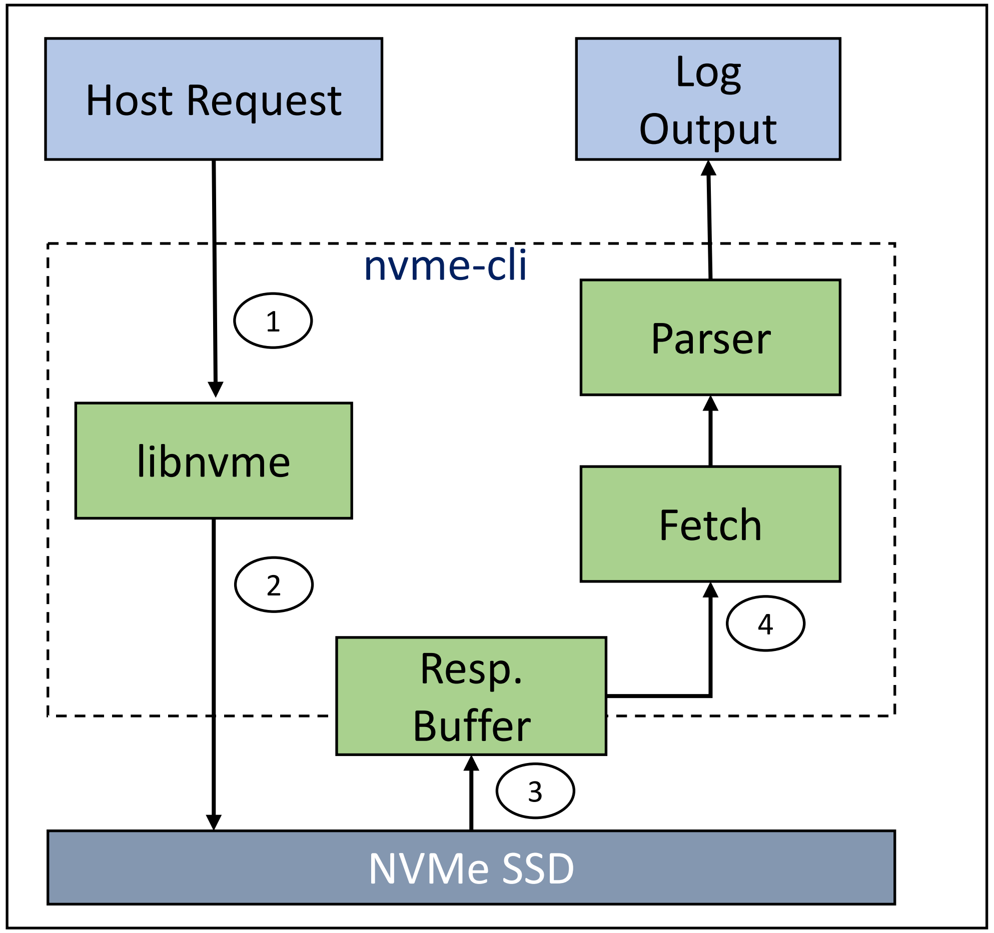
1. Host Request via nvme-cli
2. **libnvme module to communicate to SSD**
    - Get Log Page Command formation
    - Dispatch to SSD

nvme-cli : DUT Response Buffer Parsing



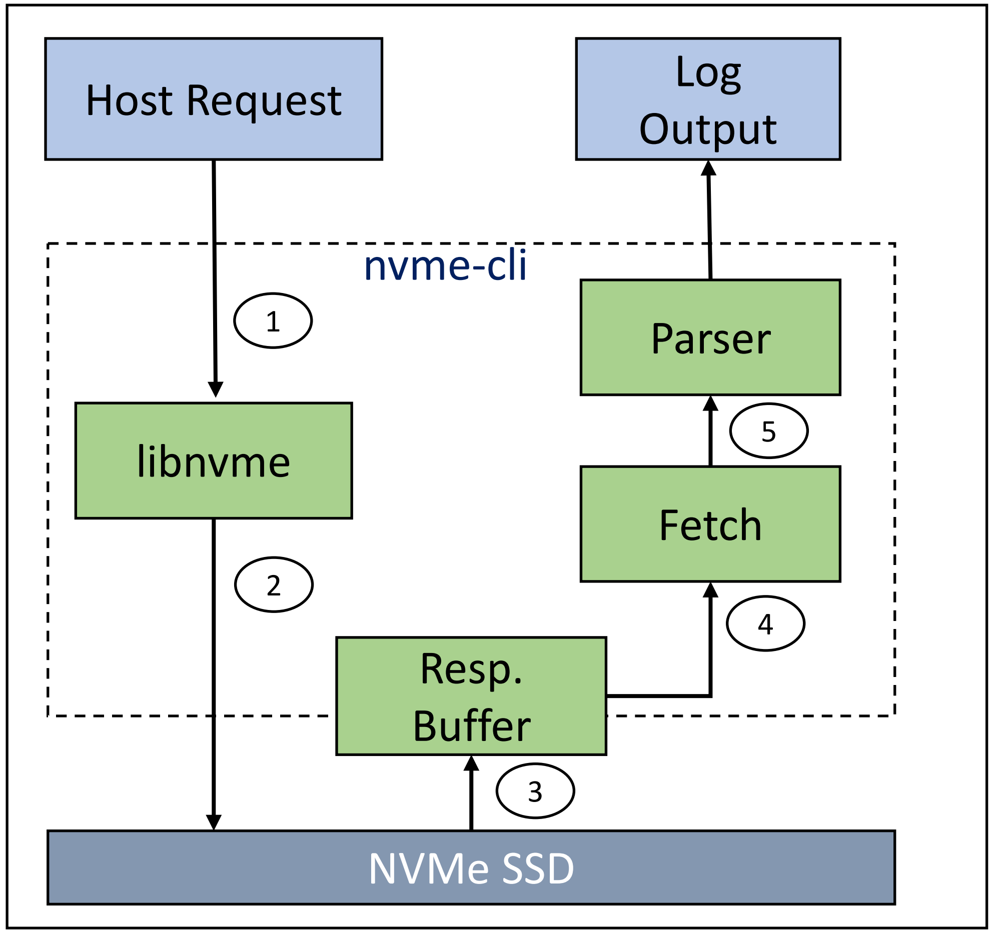
1. Host Request via nvme-cli
2. libnvme module to communicate to SSD
3. **SSD DMA data to resp. buffer based on host request type**
    - Telemetry Controller Initiated Log Page (LID : 08h)
    - Telemetry Host Initiated Log Page (LID : 07h)
    - Telemetry String log page (LID : C9h)

nvme-cli : DUT Response Buffer Parsing



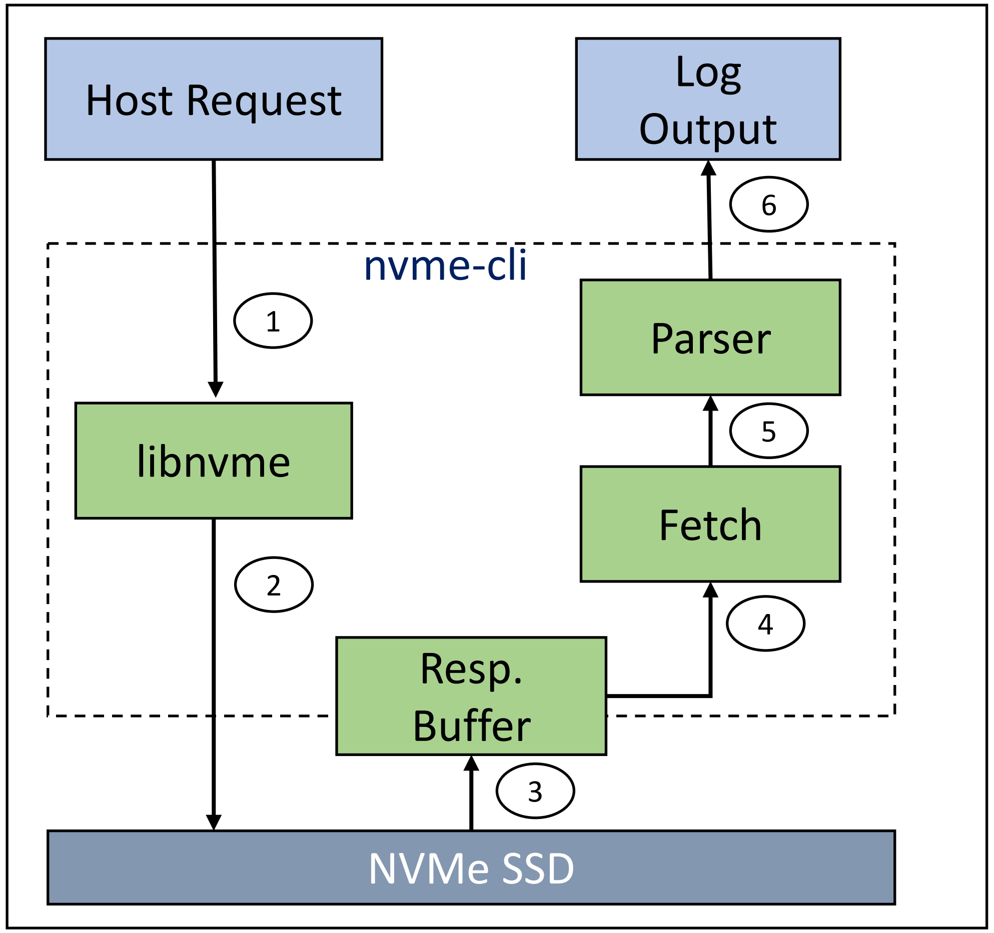
1. Host Request via nvme-cli
2. libnvme module to communicate to SSD
3. SSD DMA data to resp. buffer based on host request type
4. **Fetching response buffer using nvme-cli tool fetch mechanism**

nvme-cli : DUT Response Buffer Parsing



1. Host Request via nvme-cli
2. libnvme module to communicate to SSD
3. SSD DMA data to resp. buffer based on host request type
4. Fetching response buffer using nvme-cli tool fetch mechanism
5. **Parse the response buffer and generate output**
    - Parsing of Telemetry data and mapping to Spec Structures

nvme-cli : DUT Response Buffer Parsing



1. Host Request via nvme-cli
2. libnvme module to communicate to SSD
3. SSD DMA data to resp. buffer based on host request type
4. Fetching response buffer using nvme-cli tool fetch mechanism
5. Parse the response buffer and generate output
6. **Output Log File Generation**
   - User desired output
     - Human Readable
     - RAW Buffer
     - JSON

nvme-cli: telemetry command usage

- Telemetry Sting Log page Command usage with arguments

Usage:

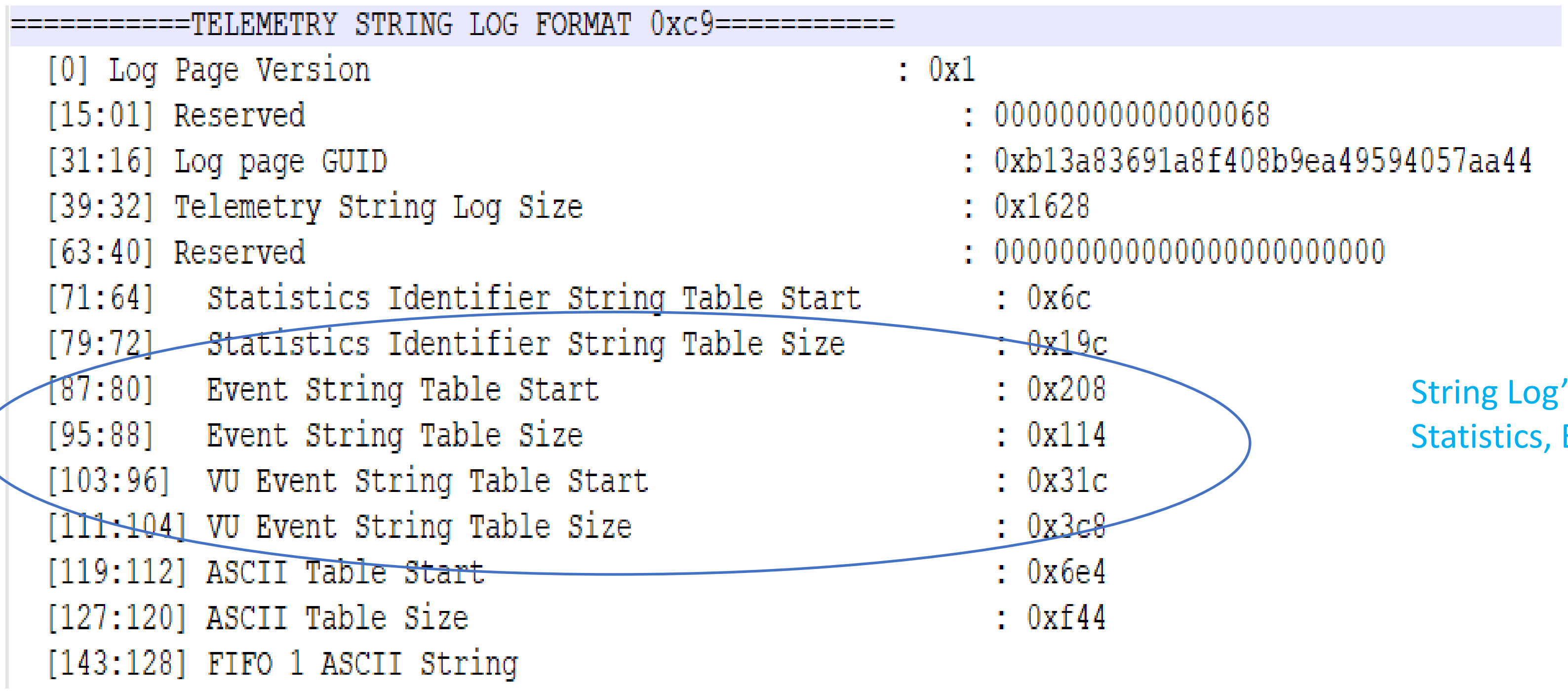
```
nvme ocp telemetry-str-log <device> [--output-format=<fmt> | -o <fmt>]
Define Parsing the telemetry string log format

Options:
-o <fmt>::
--output-format=<fmt>::
        This option will set the reporting format to normal, json, or binary.
        Only one output format can be used at a time.
```

e.g: nvme ocp telemetry-string-log /dev/nvme0n1

Output Snippet

- Telemetry String Log Page Snippet
- Human Readable format of Statistics data

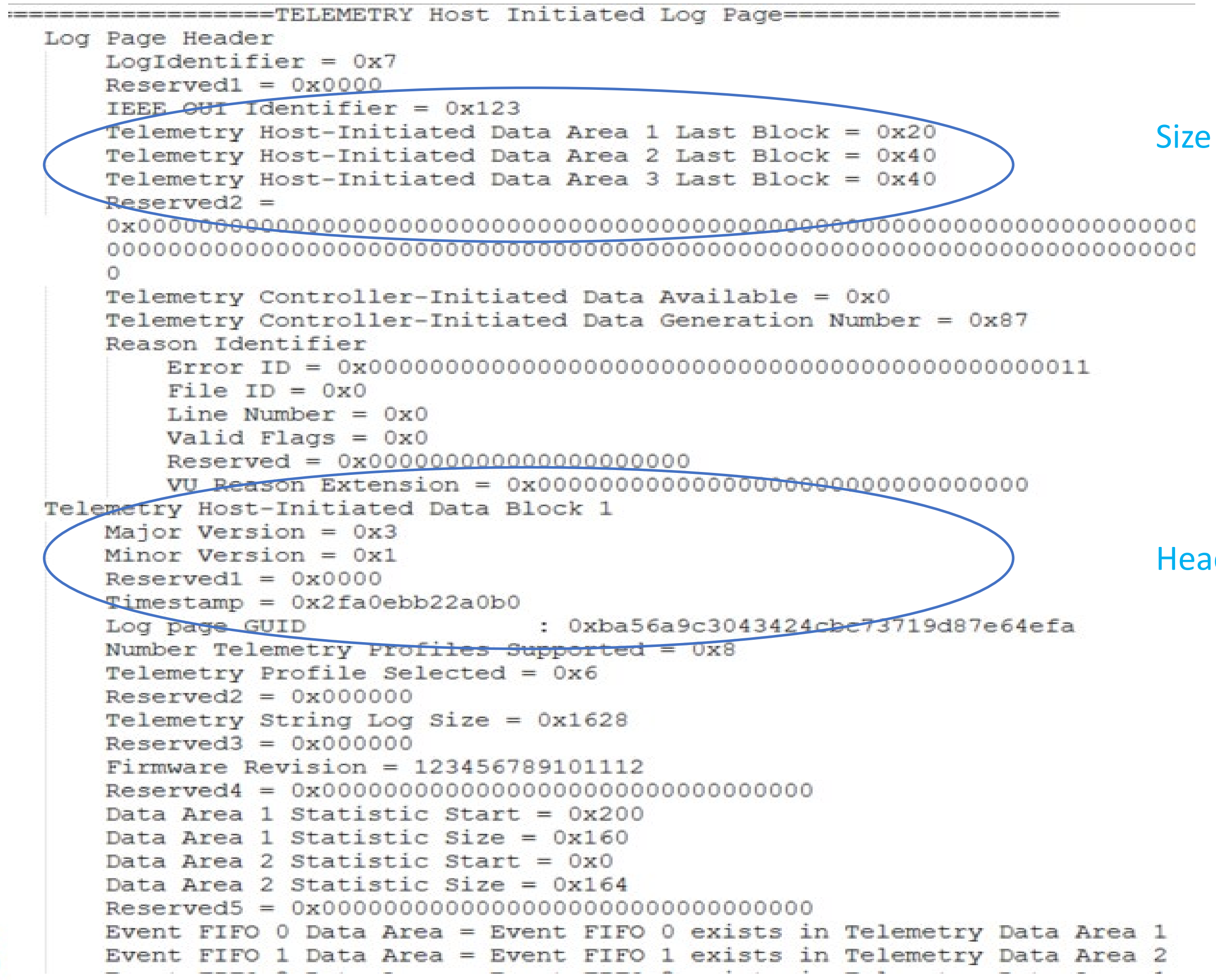


```
===========TELEMETRY STRING LOG FORMAT 0xc9===========
  [0] Log Page Version                                : 0x1
  [15:01] Reserved                                        : 0000000000000068
  [31:16] Log page GUID                                   : 0xb13a83691a8f408b9ea49594057aa44
  [39:32] Telemetry String Log Size                       : 0x1628
  [63:40] Reserved                                        : 000000000000000000000000
  [71:64]   Statistics Identifier String Table Start        : 0x6c
  [79:72]   Statistics Identifier String Table Size         : 0x19c
  [87:80]   Event String Table Start                        : 0x208
  [95:88]   Event String Table Size                         : 0x114
  [103:96]  VU Event String Table Start                     : 0x31c
  [111:104] VU Event String Table Size                      : 0x3c8
  [119:112] ASCII Table Start                               : 0x6e4
  [127:120] ASCII Table Size                                : 0xf44
  [143:128] FIFO 1 ASCII String
```

String Log's Start & Size for various Statistics, Events,etc

Output Snippet (cont.)

- Telemetry Controller/Host Initiated Log Page Snippet
- Human Readable format of Data Areas

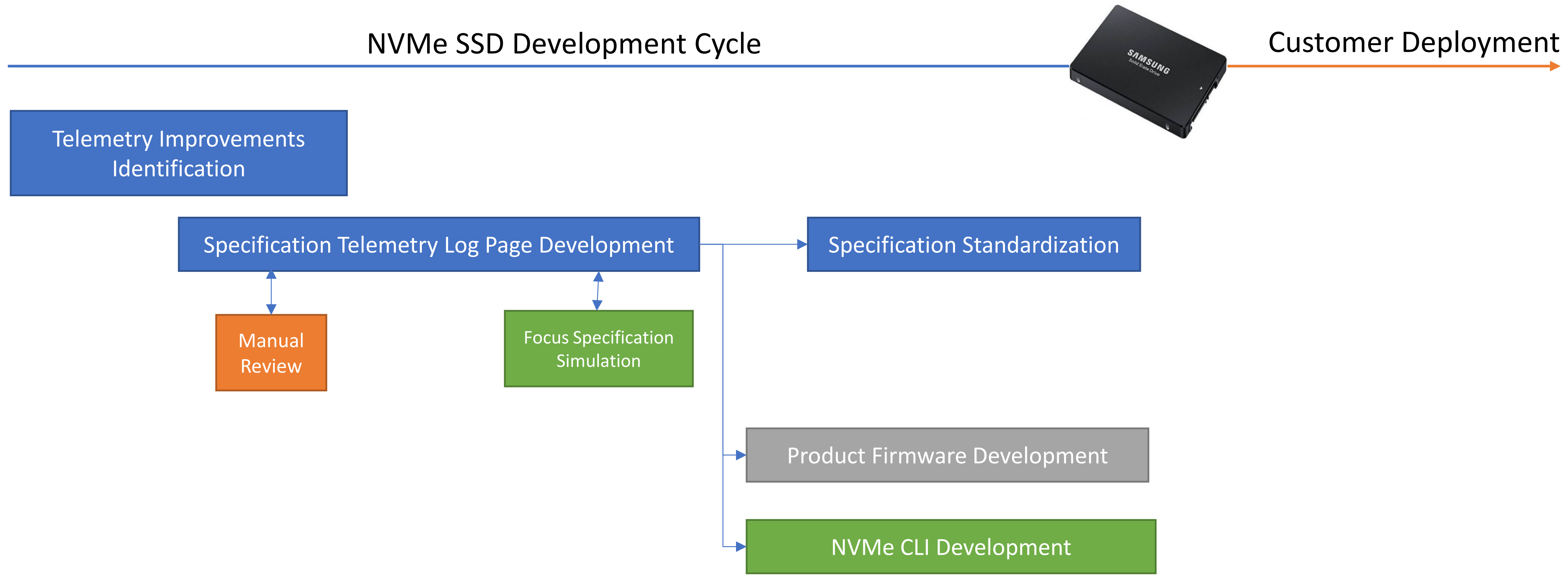


```
=================TELEMETRY Host Initiated Log Page===================
  Log Page Header
      LogIdentifier = 0x7
      Reserved1 = 0x0000
      IEEE OUI Identifier = 0x123
      Telemetry Host-Initiated Data Area 1 Last Block = 0x20
      Telemetry Host-Initiated Data Area 2 Last Block = 0x40
      Telemetry Host-Initiated Data Area 3 Last Block = 0x40
      Reserved2 =
      0x00000000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000000000
      0
      Telemetry Controller-Initiated Data Available = 0x0
      Telemetry Controller-Initiated Data Generation Number = 0x87
      Reason Identifier
          Error ID = 0x00000000000000000000000000000000000000000000011
          File ID = 0x0
          Line Number = 0x0
          Valid Flags = 0x0
          Reserved = 0x000000000000000000000
          VU Reason Extension = 0x00000000000000000000000000000000
  Telemetry Host-Initiated Data Block 1
      Major Version = 0x3
      Minor Version = 0x1
      Reserved1 = 0x0000
      Timestamp = 0x2fa0ebb22a0b0
      Log page GUID              : 0xba56a9c3043424cbc73719d87e64efa
      Number Telemetry Profiles Supported = 0x8
      Telemetry Profile Selected = 0x6
      Reserved2 = 0x000000
      Telemetry String Log Size = 0x1628
      Reserved3 = 0x000000
      Firmware Revision = 123456789101112
      Reserved4 = 0x00000000000000000000000000000000
      Data Area 1 Statistic Start = 0x200
      Data Area 1 Statistic Size = 0x160
      Data Area 2 Statistic Start = 0x0
      Data Area 2 Statistic Size = 0x164
      Reserved5 = 0x00000000000000000000000000000000
      Event FIFO 0 Data Area = Event FIFO 0 exists in Telemetry Data Area 1
      Event FIFO 1 Data Area = Event FIFO 1 exists in Telemetry Data Area 2
```

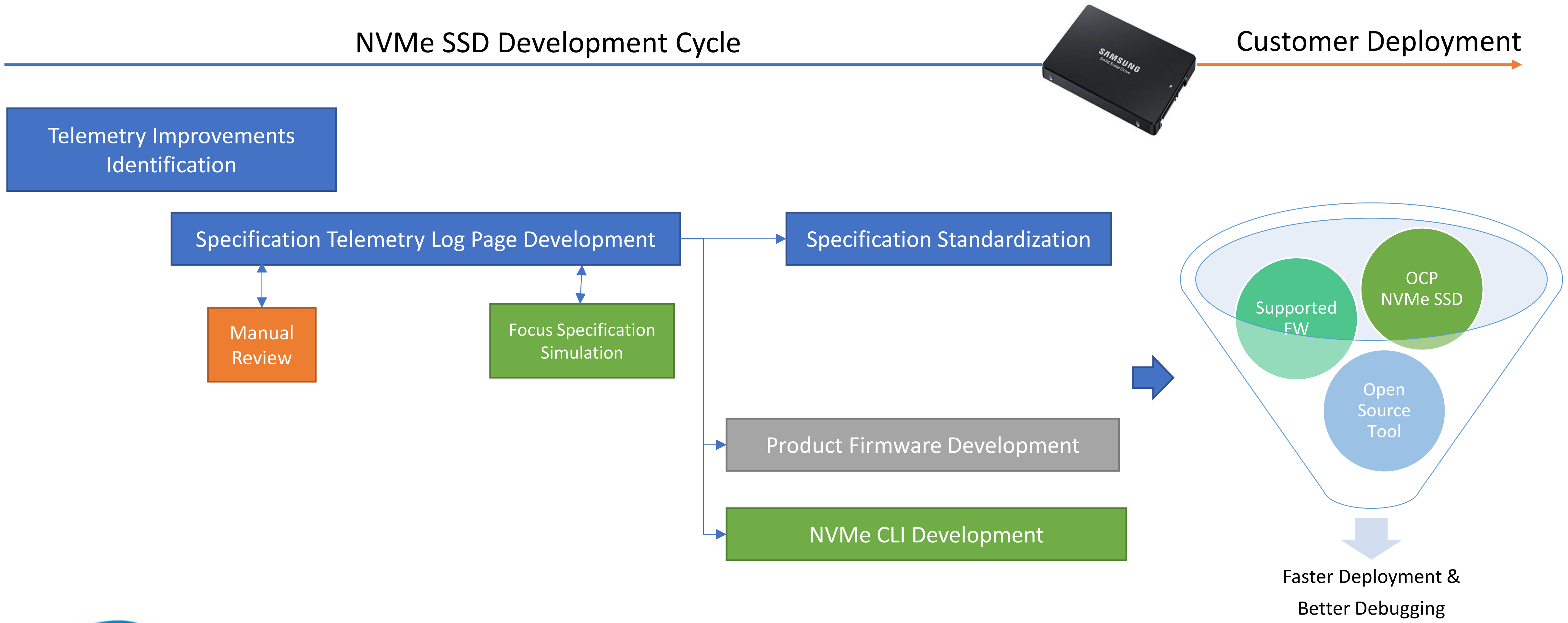
Size of Telemetry Data Area

Header of the Host Initiated Data Area 1 Details

Concurrent Readiness of Telemetry Feature


NVMe SSD Development Cycle
Customer Deployment
SAMSUNG
Solid State Drive
Telemetry Improvements Identification
Specification Telemetry Log Page Development
Specification Standardization
Manual Review
Focus Specification Simulation
Product Firmware Development
NVMe CLI Development

Concurrent Readiness of Telemetry Feature


NVMe SSD Development Cycle
SAMSUNG
Solid State Drive
Customer Deployment
Telemetry Improvements Identification
Specification Telemetry Log Page Development
Specification Standardization
Manual Review
Focus Specification Simulation
Product Firmware Development
NVMe CLI Development
Supported FW
OCP NVMe SSD
Open Source Tool
Faster Deployment &
Better Debugging

Thank You