



Flash Memory Summit

Standardization for a Key-Value Interface underway at NVM Express and SNIA

Bill Martin (Samsung)



Disclaimer

This presentation and/or accompanying oral statements by Samsung representatives collectively, the “Presentation”) is intended to provide information concerning the SSD and memory industry and Samsung Electronics Co., Ltd. and certain affiliates (collectively, “Samsung”). While Samsung strives to provide information that is accurate and up-to-date, this Presentation may nonetheless contain inaccuracies or omissions. As a consequence, Samsung does not in any way guarantee the accuracy or completeness of the information provided in this Presentation.

This Presentation may include forward-looking statements, including, but not limited to, statements about any matter that is not a historical fact; statements regarding Samsung’s intentions, beliefs or current expectations concerning, among other things, market prospects, technological developments, growth, strategies, and the industry in which Samsung operates; and statements regarding products or features that are still in development. By their nature, forward-looking statements involve risks and uncertainties, because they relate to events and depend on circumstances that may or may not occur in the future. Samsung cautions you that forward looking statements are not guarantees of future performance and that the actual developments of Samsung, the market, or industry in which Samsung operates may differ materially from those made or suggested by the forward-looking statements in this Presentation. In addition, even if such forward-looking statements are shown to be accurate, those developments may not be indicative of developments in future periods.



What is Key Value

- A mechanism to store user data associated with a key
- Key may be any length (byte granularity)
- Value may be any length (byte granularity)



Differences between KV and Block interface

- **Block**

- User data is in multiples of block length
- Logical block address is a number from 0 to max LBA

- **KV**

- Variable length user data
- Key is a variable length identifier



Differences between Object and KV interface

- KV

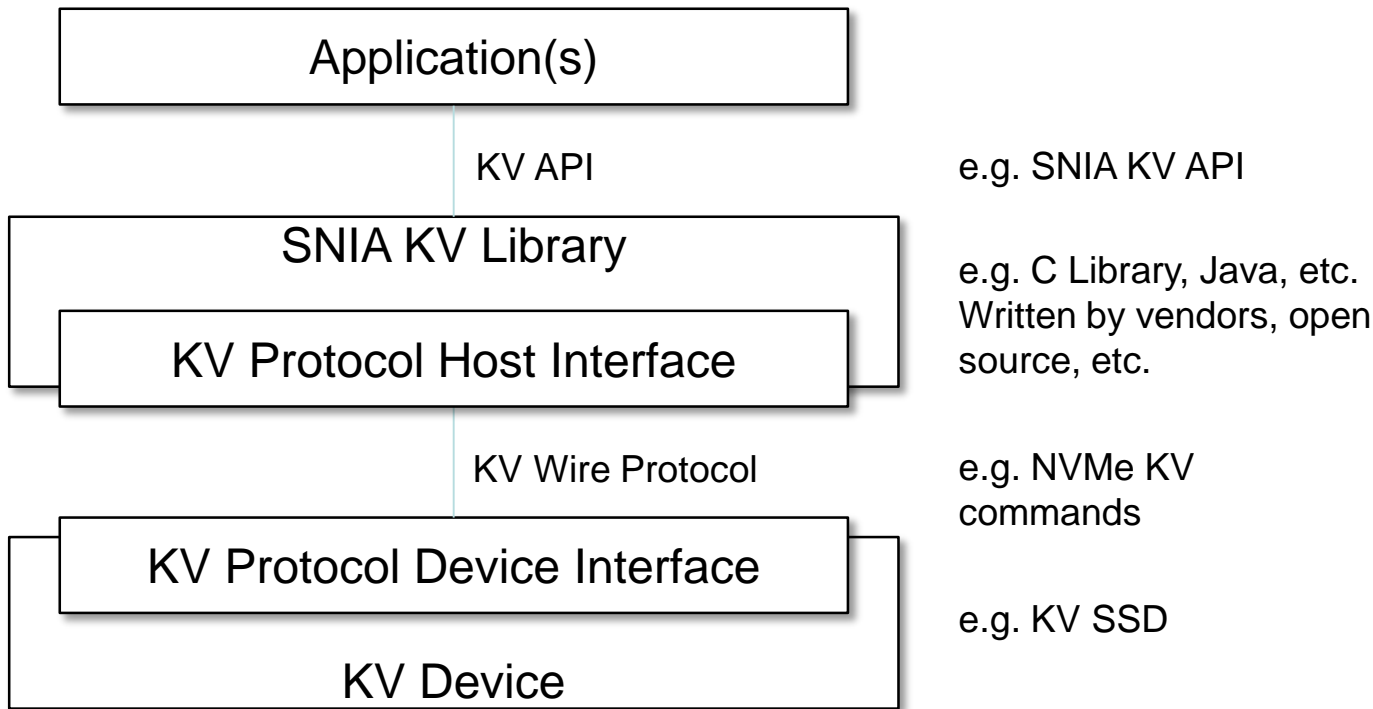
- Tool to facilitate object storage
- Keys may not be ordered
- Not searchable based on contents of value

- Object

- Keys are ordered
- More functionality to search objects
 - Does value contain "X"
- Support logging or other mechanisms to maintain database integrity



Key Value SSD layers





SNIA KV API Status

- Version 1.0 has been approved and is publically available
 - https://www.snia.org/tech_activities/standards/curr_standards/kvsapi
- Allows library calls independent of the underlying transport
 - NVME
 - SCSI
 - SATA



What does the API define?

- Structure
 - Key Space
 - Key Group
 - Key Value Pair
 - Device info
 - Key Space info
- Access
 - Store
 - Retrieve
 - Delete
 - Delete Group
 - Exist
 - List
 - Iterator



Iterator Function

- enables a device to prepare a Key Group of keys for iteration by matching a given bit pattern
- Allows listing/deleting keys/values within the iterator group



NVMe KV work

- Working on a Key Value command standard
- Relies on other work in NVMe to provide supporting structure for NVMe KV
 - Namespace Types/ Multiple Command sets
- Expect to release later this year



Features of KV Command set

- Currently Key limited to 16 bytes
- Comparable commands to the API definitions in SNIA
 - Store
 - Retrieve
 - Delete
 - List
 - Exist



Future work on NVMe KV

- Extended key length
- Append
- Retrieve (Index)
- Sorted Keys



Software/Driver Support

- Samsung has open source code available
 - Currently proprietary
 - SNIA API available in next 2 months
 - Samsung KV API, Kernel driver
 - Public github:
 - <https://github.com/OpenMPDK/KVSSD>
 - Kv userspace driver:
 - <https://github.com/OpenMPDK/uNVMe>
 - KV Ceph: Ceph object storage designed for Samsung Key-Value SSD
 - <https://github.com/OpenMPDK/KVCeph>



Flash Memory Summit

Thank You