

**SDC** | 19

September 23-26, 2019  
Santa Clara, CA

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

**Bill Martin**  
**Samsung**



# Disclaimer

June 10, 2019  
Santa Clara, CA

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

Storage Developer Conference  
Santa Clara, CA

- 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

September 23-26, 2019

SDC<sup>19</sup>

- **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

September 23-26, 2019

SDC<sup>19</sup>

- KV

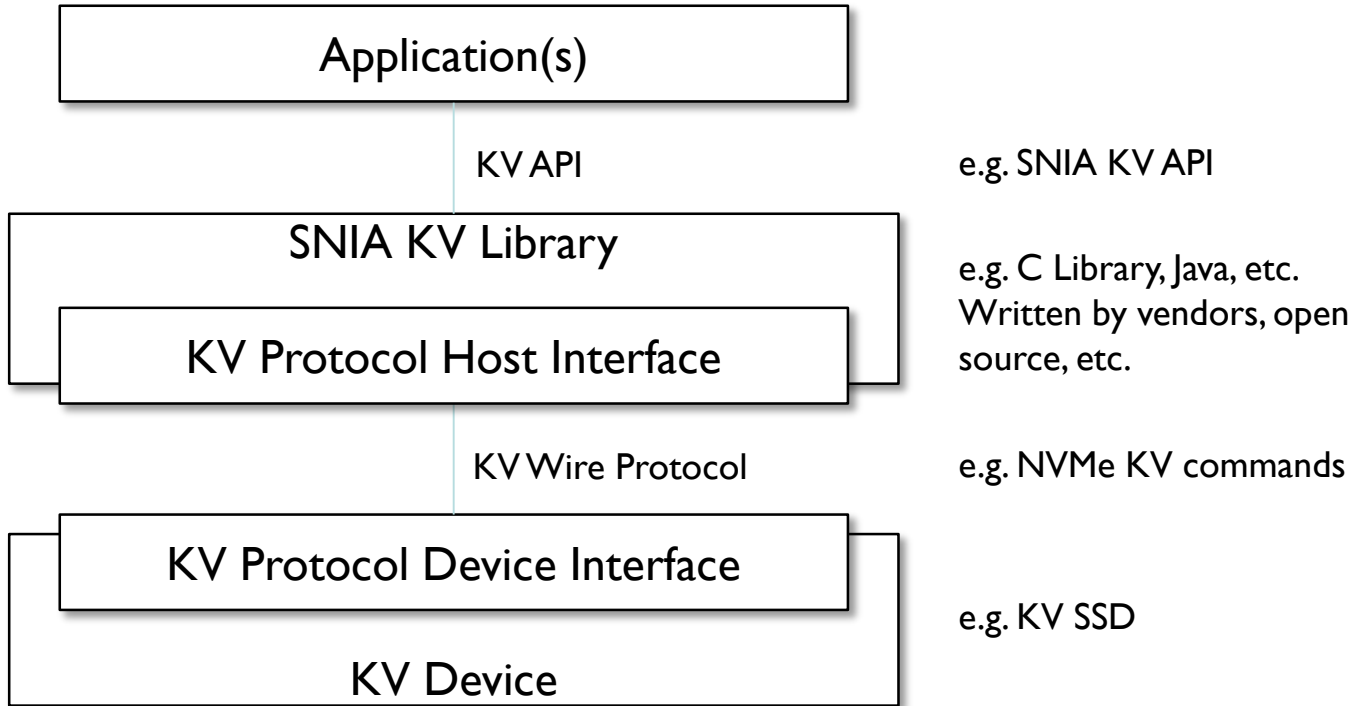
- Tool to facilitate object storage
- Keys may not be ordered
- Not searchable based on contents of value

- 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

September 23-26, 2019  
Santa Clara, CA



# SNIA KV API Status

September 10-12  
Santa Clara, CA

- Version 1.0 has been approved and is publically available
  - [https://www.snia.org/tech\\_activities/standards/curr\\_standards/kvsapi](https://www.snia.org/tech_activities/standards/curr_standards/kvsapi)
- Open Source code available:
  - <https://github.com/OpenMPDK/KVSSD>
- Allows library calls independent of the underlying transport
  - NVME
  - SCSI
  - SATA

# What does the API define?

## Structure

September 23-26, 2019

SDC<sup>19</sup>

- Structure
  - Key Space
    - Equivalent of Namespace in NVMe
  - Key Group
    - Group of keys based on portion of key and mask
  - Key Value Pair
    - Unit of user data storage
  - Device info
    - Characteristics of device (all key spaces)
  - Key Space info
    - Characteristics of specific Key Space



# What does the API define?

September 23-26, 2019

## Access, CA

SDC<sup>19</sup>

- Access
  - Store
  - Retrieve
  - Delete
  - Delete Group
    - Delete all Key Value pairs in a Key Group
  - Exist
  - List
  - Iterator

# Iterator Function

Storage  
Santa Clara, CA

- 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

Santa Clara, CA

- 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

# Namespace Type

## Multiple Command sets

- One Command set per namespace type
- NVM Command set is traditional block storage
- Key Value Command set being defined
- Zoned Namespace Command set being defined

# Features of KV Command set

## Keys

September 23-26, 2019

San Jose, CA

SDC<sup>19</sup>

- Currently Key limited to 16 bytes
- A Key of 4 bytes CANNOT match a Key of 8 bytes even if it is null padded

AD87 90EF

<>

0000 0000 AD87 90EF

- A Key is of byte granularity
- Future work will extend Key length

# Features of KV Command set

- Comparable commands to the API definitions in SNIA
  - Store
    - Key Value pair stored
  - Retrieve
    - Key Value pair retrieved
  - Delete
    - Delete a single Key Value Pair
  - List
    - List in device specific order Keys
      - Start from a specified Key if that Key exists in the device
  - Exist
    - Does a specific Key exist in the device

# Future work on NVMe KV

Santa Clara, CA

- Extended key length
- Append
- Retrieve (Index)
- Key Group support
- Sorted Keys
- Exist multiple
- Delete multiple

# Software/Driver Support

Santa Clara, CA

- 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>



September 23-26, 2019  
Santa Clara, CA

SDC<sup>19</sup>

# Thank You