Flash Memory Summit

Storage Performance Development Kit (SPDK) Flash Translation Layer Library for Zoned Namespace SSDs

Wojciech Malikowski

Intel Non-volatile Memory Solutions Group (NSG)

Legal Disclaimer

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure**.** Check with your system manufacturer or retailer or learn more at intel.com.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel, the Intel logo, and others are trademarks of Intel Corporation in the U.S. and/or other countries.

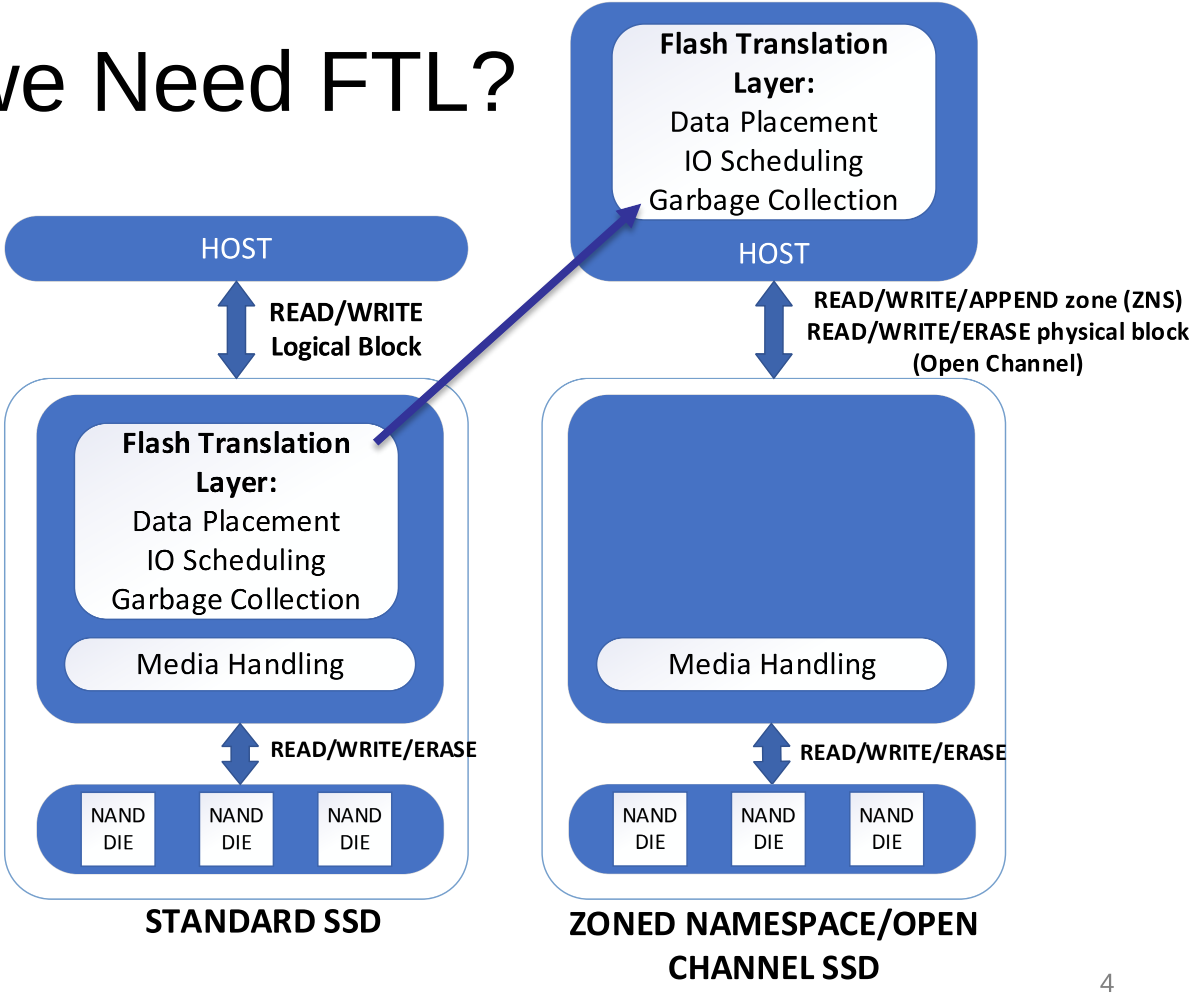
*Other names and brands may be claimed as the property of others.

Agenda

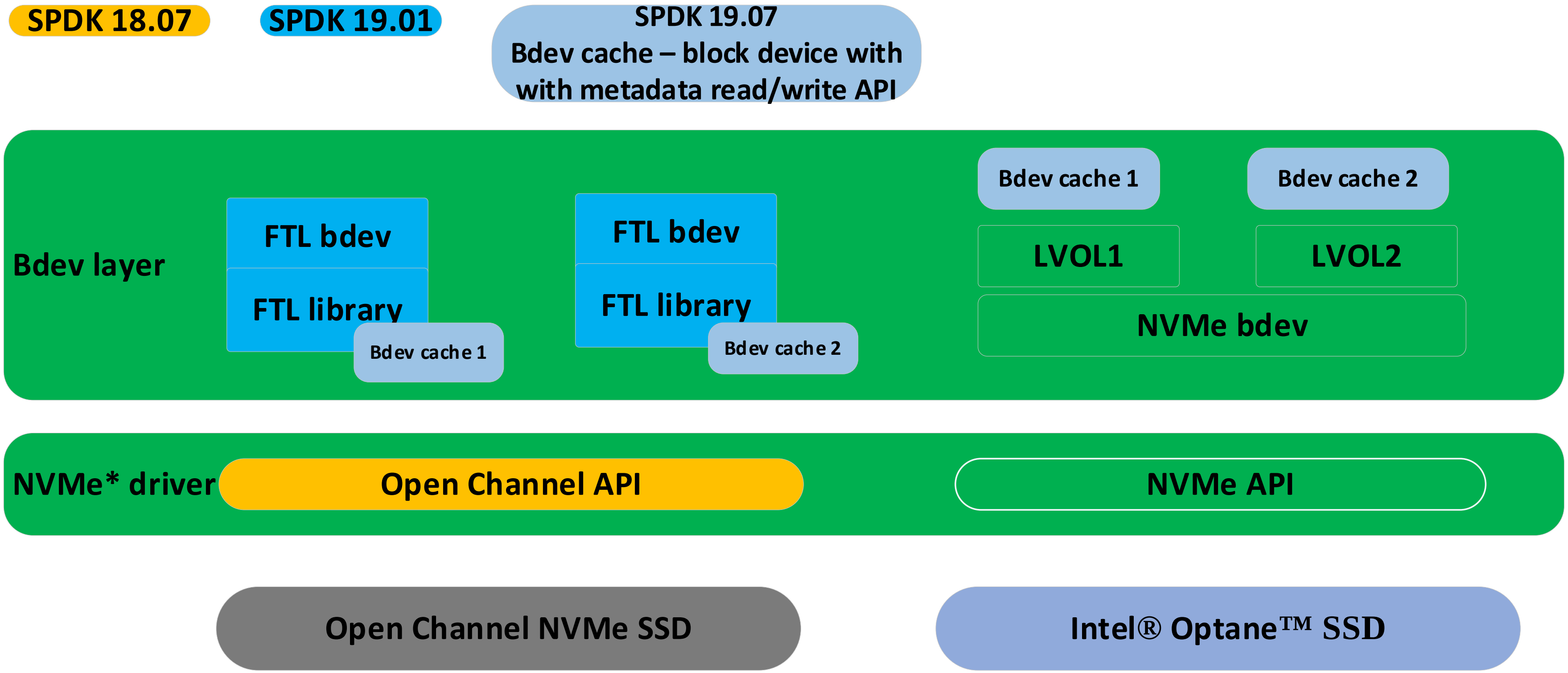
- Storage Performance Development Kit (SPDK) Flash Translation Layer (FTL) library current state
- FTL core components overview
- Moving to Zoned namespace

Why do we Need FTL?

- Standard SSD provides Flash Translation Layer inside firmware
- Storage Performance Development Kit (SDPK) FTL provides block device access on top of non block SSD device implementing Open Channel/Zoned Namespace interface
- FTL logic should be moved from SSD firmware to the host



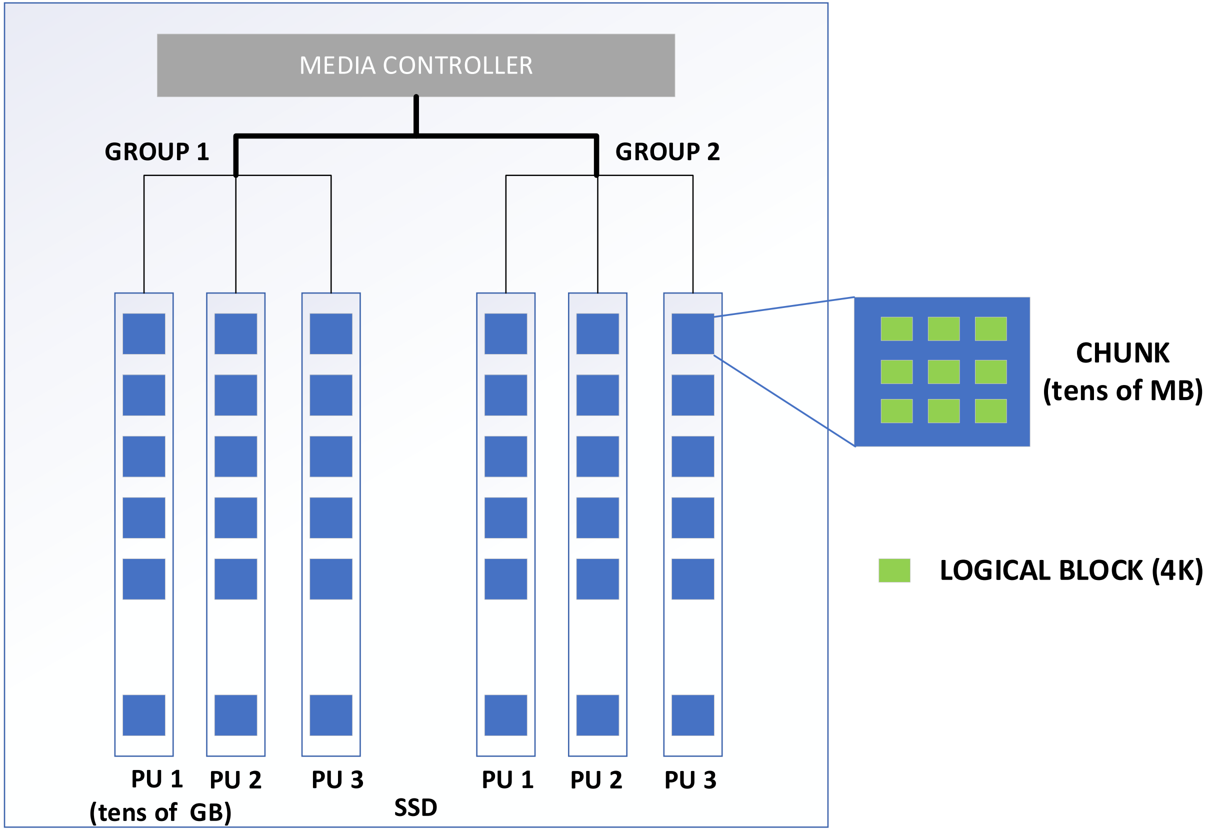
Storage Performance Development Kit (SPDK) FTL Library



*Other names and brands may be claimed as the property of others.

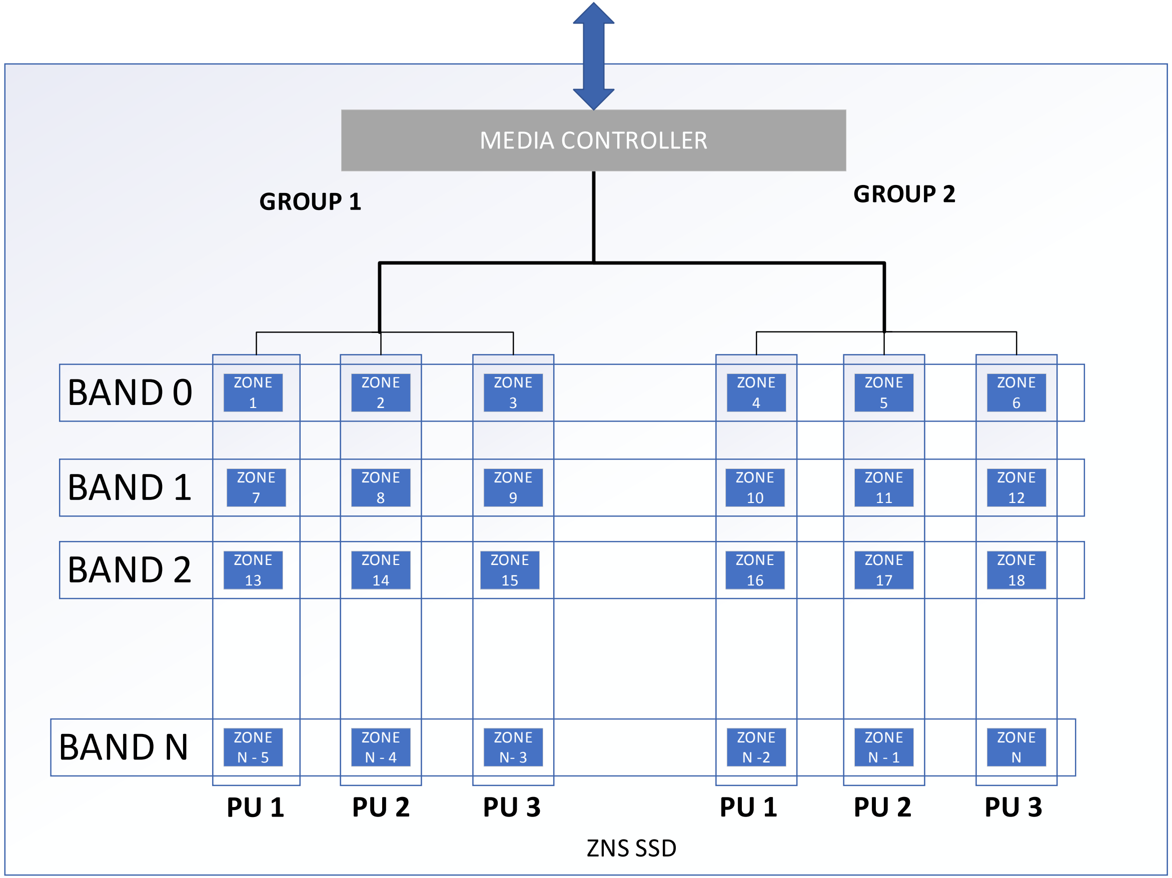
SSD Geometry

- GROUP
- PU – parallel unit
- CHUNK
- LOGICAL BLOCK



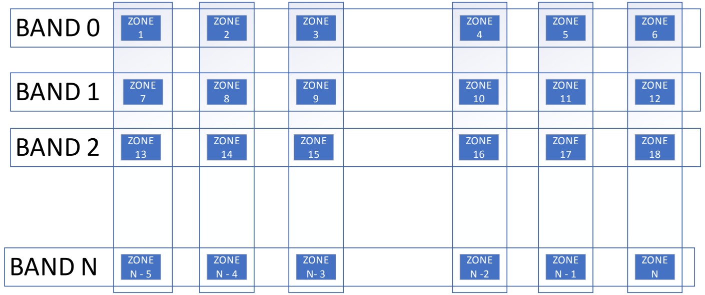
Band

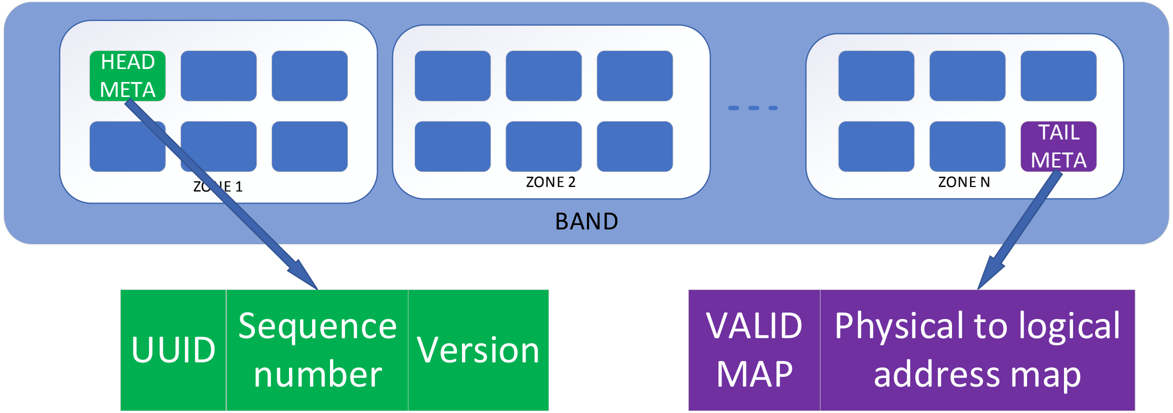
- CHUNK -> ZONE
- Band - collection of zones, each belongs to a different parallel unit
- FTL write pointer iterates over the zones in the band to achieve maximum write parallelism
- Band could be in open, close or free state



Metadata

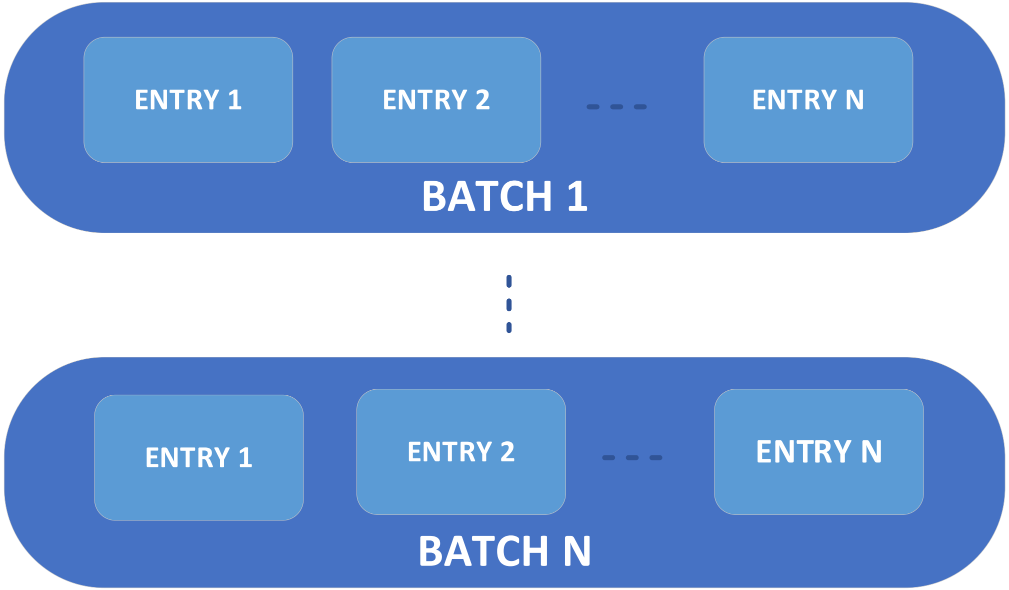
- We need to store metadata in each band for device restoring and defragmentation process
- When band is opening, head metadata is written and when band becomes full, we write tail metadata
- Head metadata contains: device UUID, sequence number, version etc.
- Tail metadata contains LBA map for its band and its validity





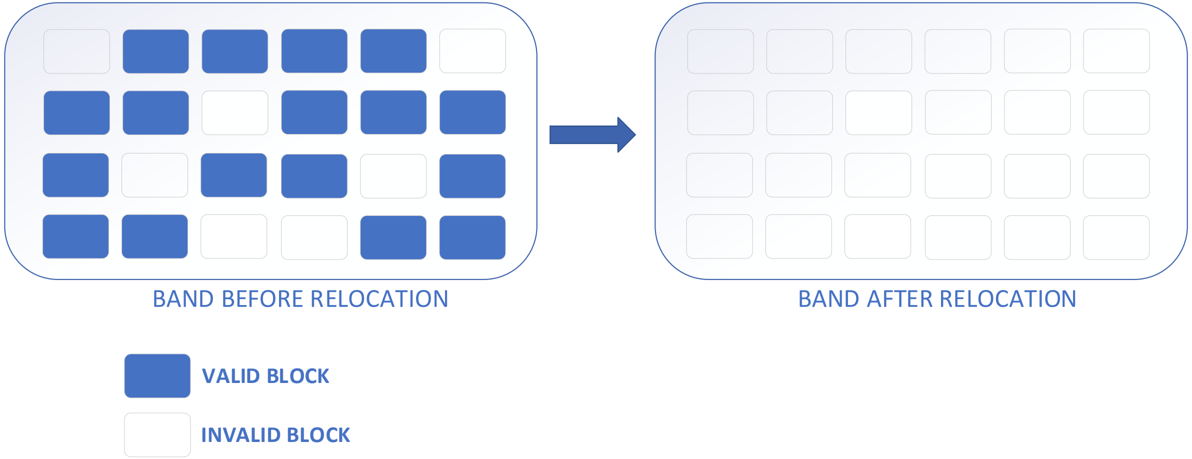
Write Buffer

- Optimal write size unit for ZNS SSD is greater then 4K block size e. g. 128K
- Write buffer collects writes before they can be submitted onto disk
- To provide power fail safety at 4K level we need persistent write buffer.



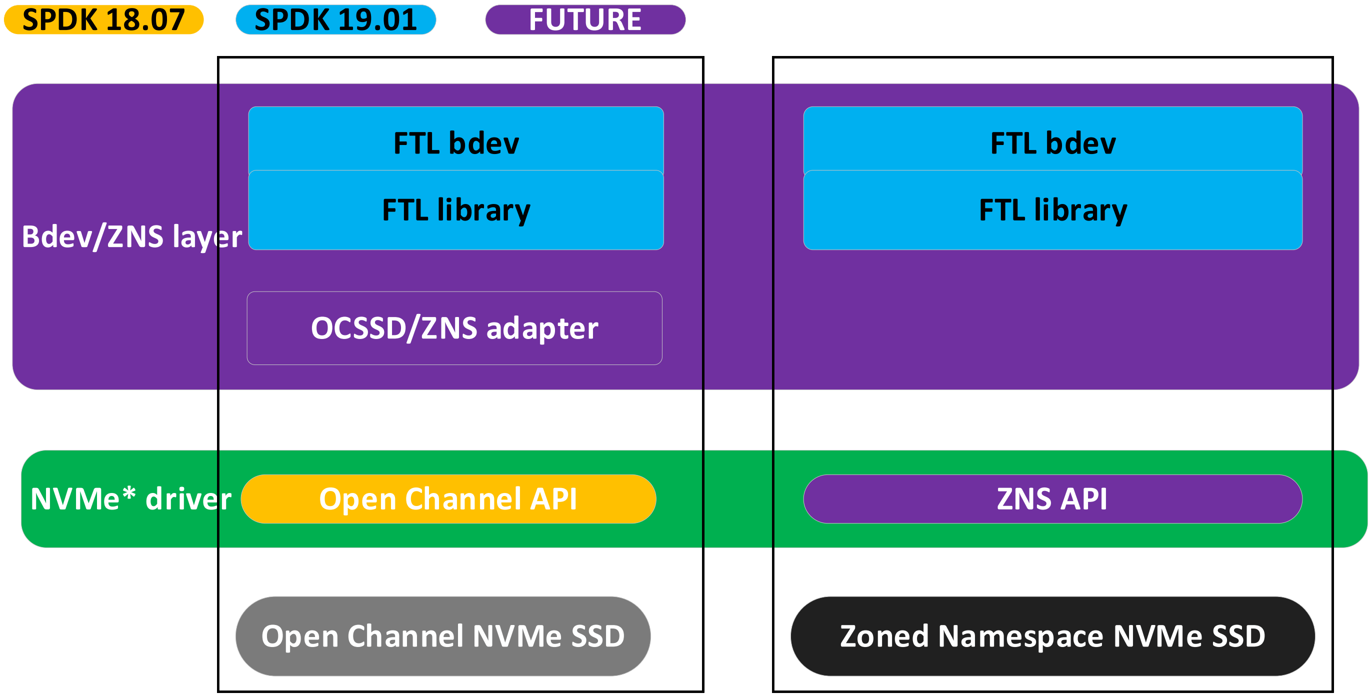
Relocation Module

- Manages band’s defragmentation process
- Each band has its own merit based on its age, write count and validity



Moving to Zoned Namespaces

- Extend existing BDEV interface to support zoned device
- spdk_zdev_get_info()
  - zone_size
  - max_open_zones
  - optimal_open_zones
- spdk_zdev_zone_info
  - start_lba
  - write_pointer:
  - capacity
  - state
- ZONE_MANAGMENT
  - Close
  - Finish
  - Open
  - Reset



*Other names and brands may be claimed as the property of others.

Summary

- Host FTL provides more control to applications
- Extra control can be used to provide
  - WAF reduction
  - Better isolation
  - Better QoS

Start using FTL with SPDK today: https://spdk.io/doc/ftl.html