



Software-Enabled Flash Storage™ for Hyperscale Data Center

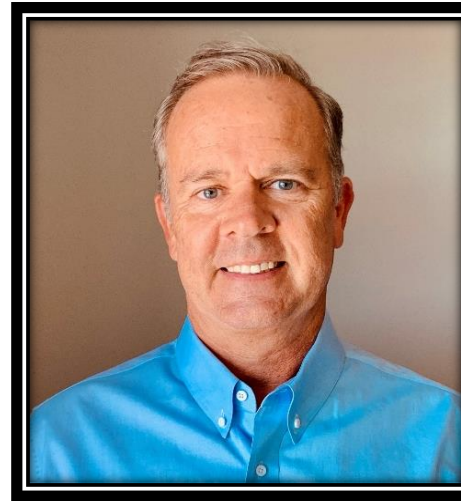
Sponsored by NVM Express organization, the owner of NVMe® specifications

Speakers



Earle Philhower
Senior Marketing
Manager

KIOXIA



Sean Stead
Senior Staff

KIOXIA

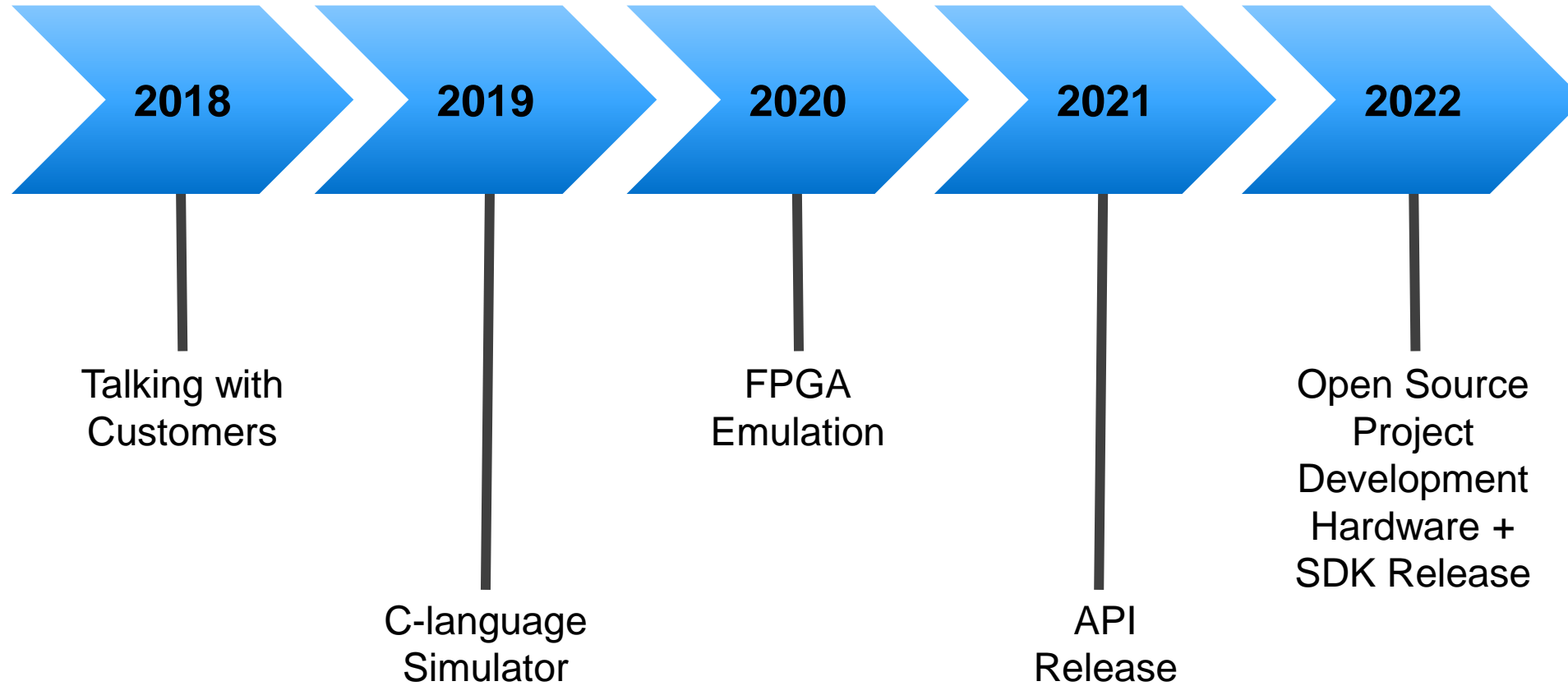
Agenda

- Development
- Capabilities
- Open Source Platform
- Hardware
- Software
- Join the Project

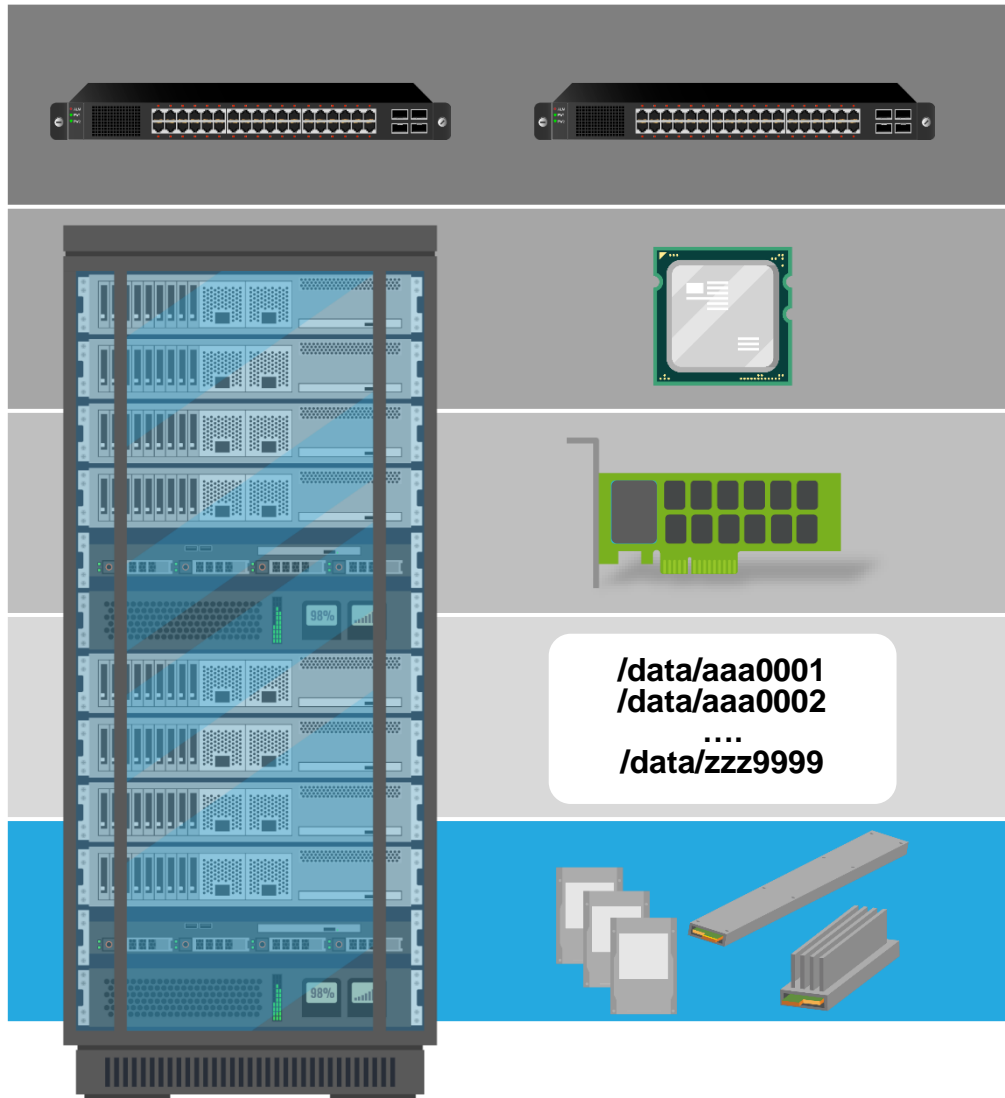
Well, how did we get here?

Hyperscale needs drove open source development.

Path to Software-Defined Flash



Software-Defined Data Center



Software-Defined Networking

Software-Defined Silicon

Software-Defined Accelerators

Software-Defined Storage

Apply the same idea to flash...

Hyperscale Drives the Development

Flash **Is NOW** ~~needs to be~~
Software-Defined

Software-Enabled Flash™ Technology

Making Flash Storage Software-Defined

Remove HDD-based limitations



Optimize for flash memory's digital nature



Engineered to meet cloud challenges



Software-Enabled Flash™

A Software-Defined Flash API

Sheds the legacy HDD paradigm



Flash can behave in a much more predictable and uniform manner with:

- Data placement and workload isolation
- Latency control with advanced queuing
- Multi-protocol capabilities

Unlock the full power of flash storage

Software-Enabled Flash™

**A force
multiplier
in data
center
economics**



Isolated tenant environments



Deliver tiered service levels



Better flash economics (TCO)



Faster time to market (TTM)



Open Source ecosystem

Join, contribute, lead.



The Project is Live

Membership is open

Join the project
contribute to the future of
flash becoming
“Software-Defined”

Governance Model for Software-Enabled Flash





<https://softwareenabledflash.org>

Vendor-neutral collaboration

Software-Enabled Flash™ Hardware

Built for optimal flash use and cloud-based workloads

Hardware and workload isolation

Data placement control for optimal layout

Dynamic QoS domains for dynamic workloads

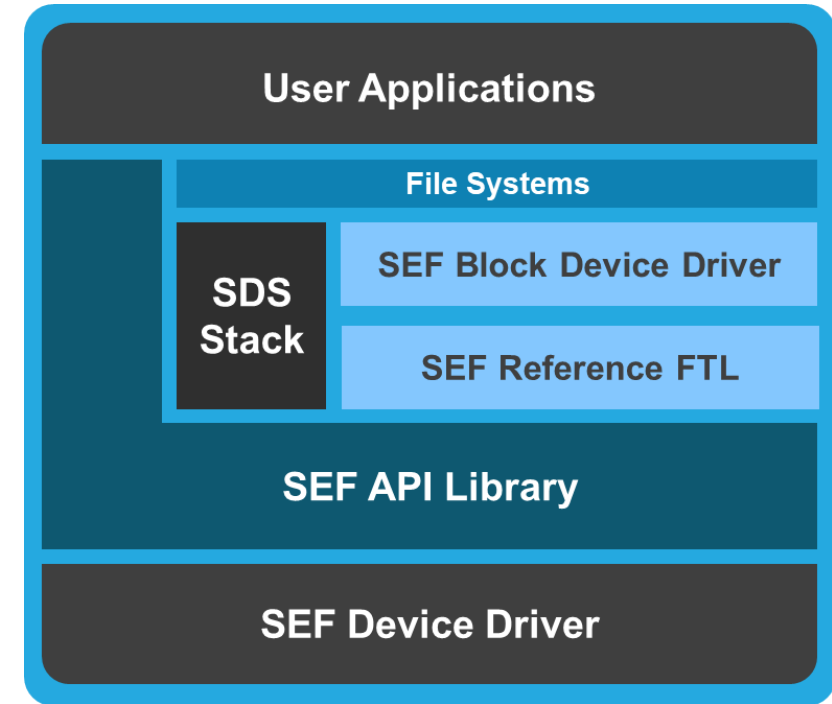
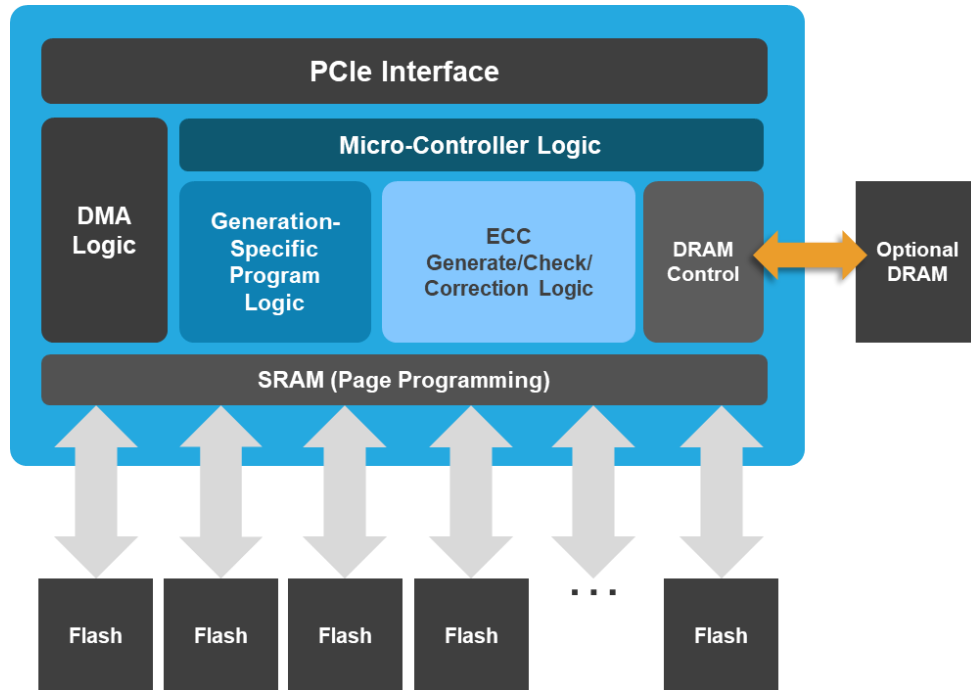
Multiple queueing modes for latency outcome control

Copy offload for bandwidth and CPU savings

Flash abstraction for easier migration

Optimized Hardware

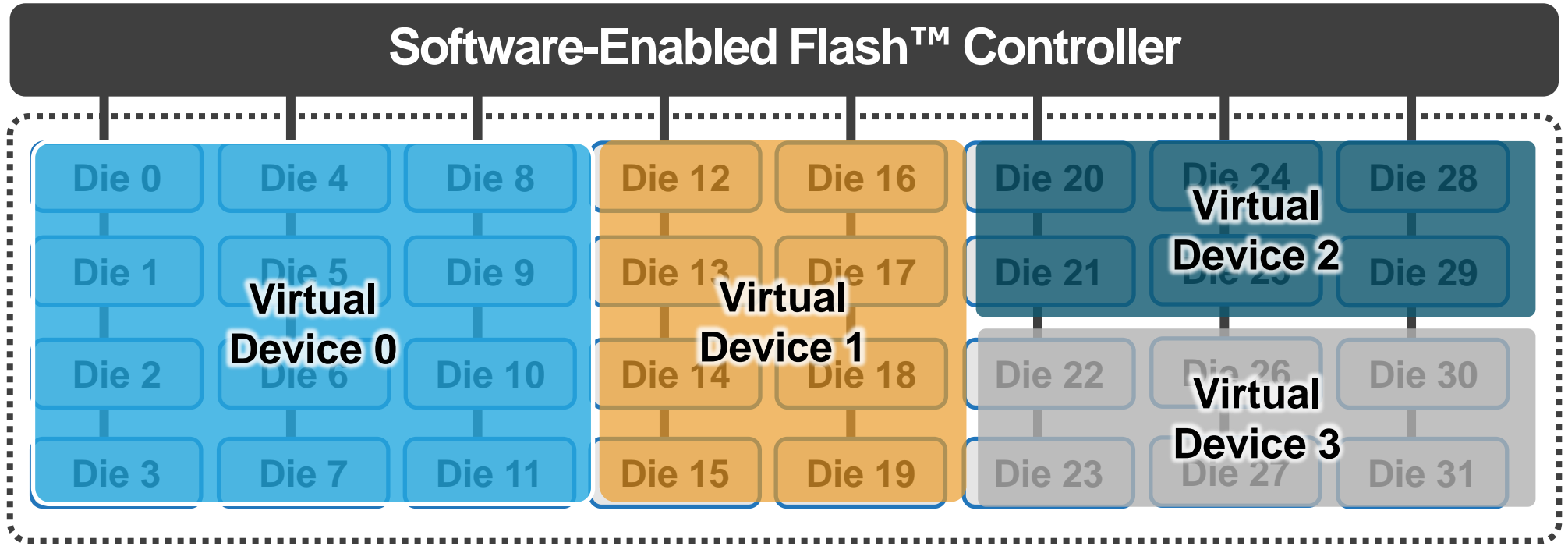
+ Open Source Software



The hardware focuses on media management and host offload functionality...

...allowing software to orchestrate and manage protocols, latencies and data placement

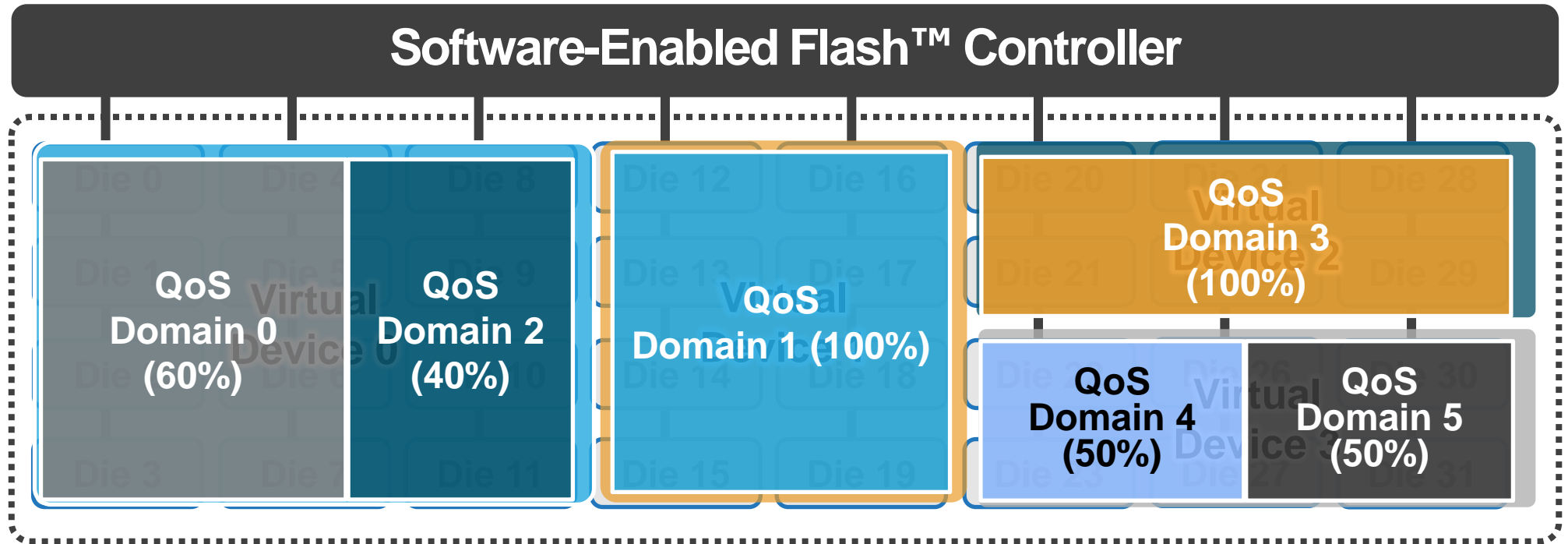
Hardware Based Isolation



Virtual Device
Hardware Isolation

Ensure contention-free performance

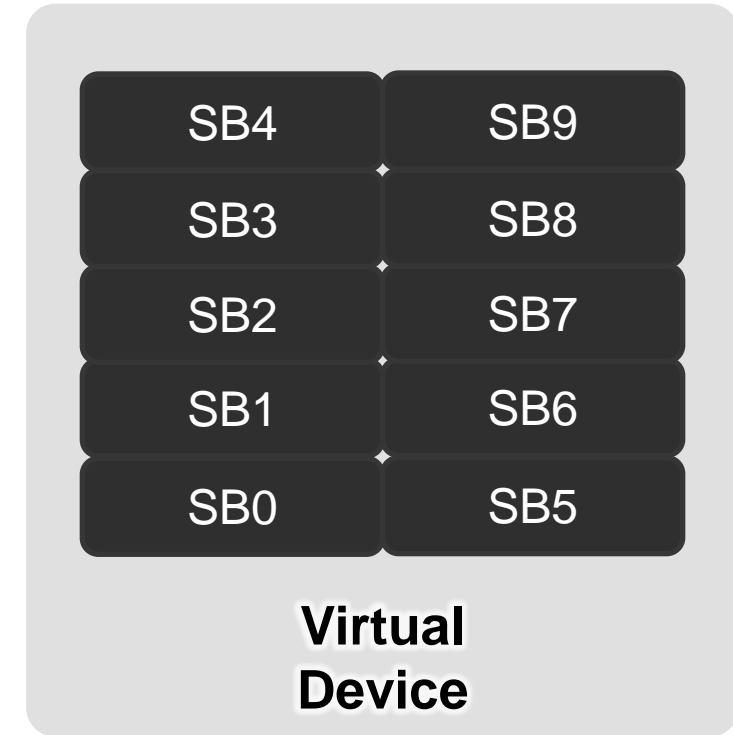
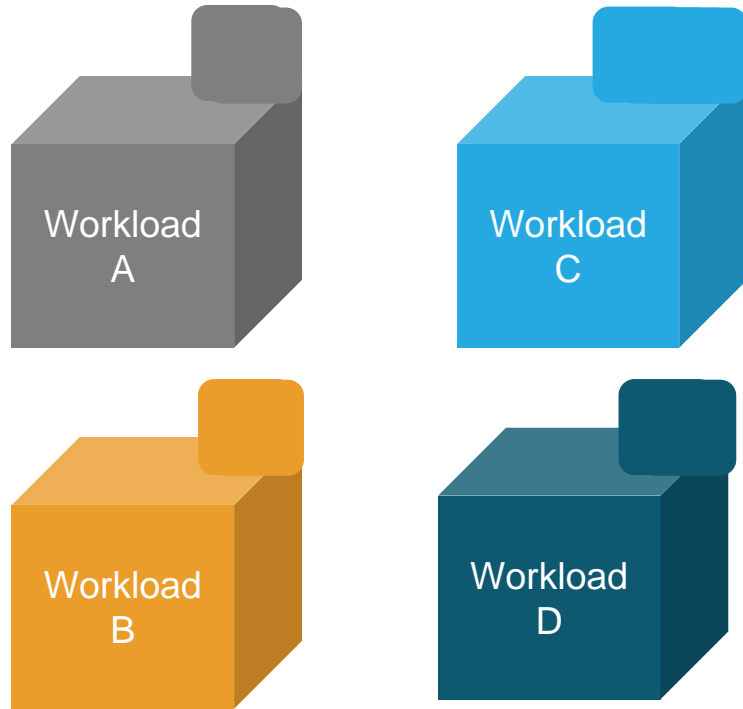
Workload Based Isolation



Quality of Service (QoS) Domain

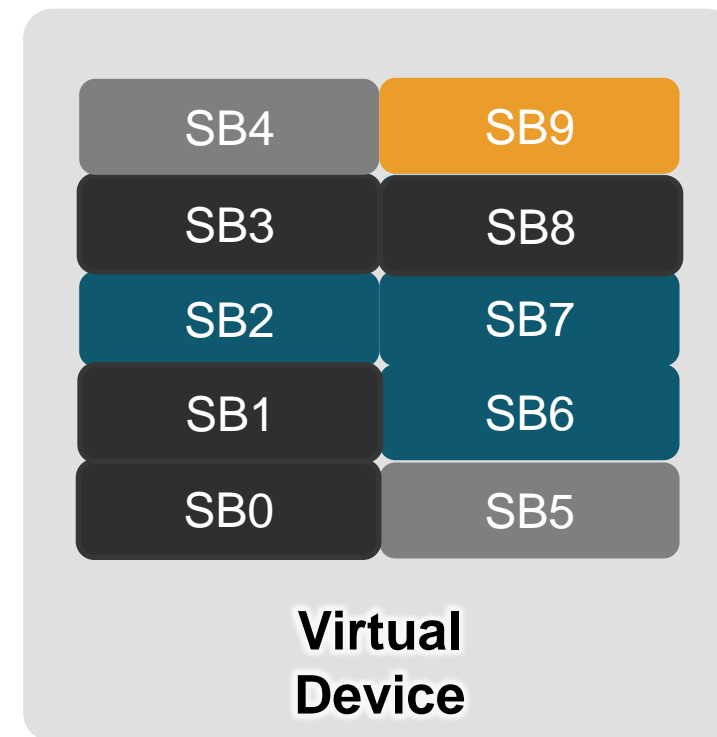
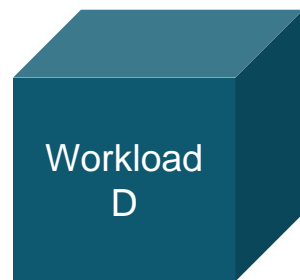
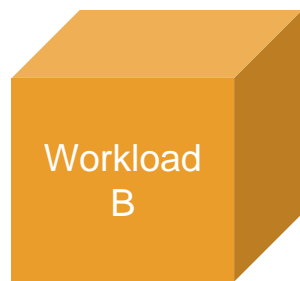
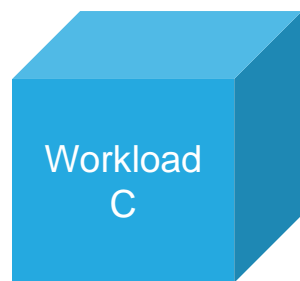
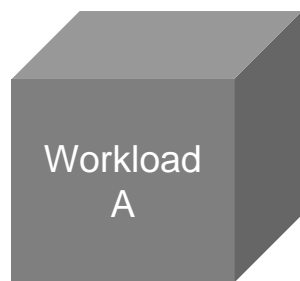
Isolate dynamically, per workload

Data Placement Control



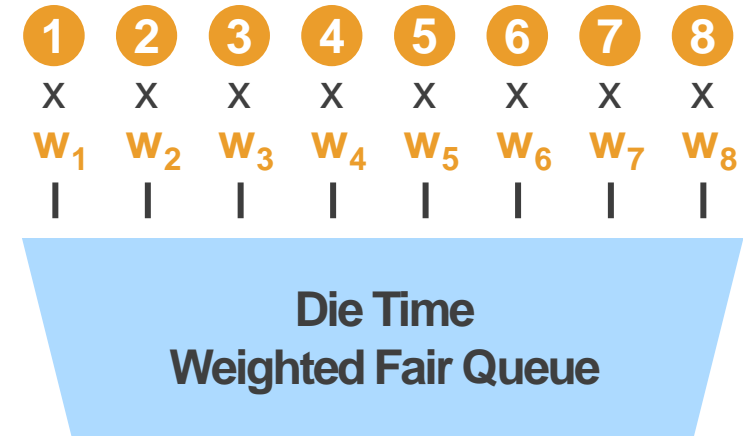
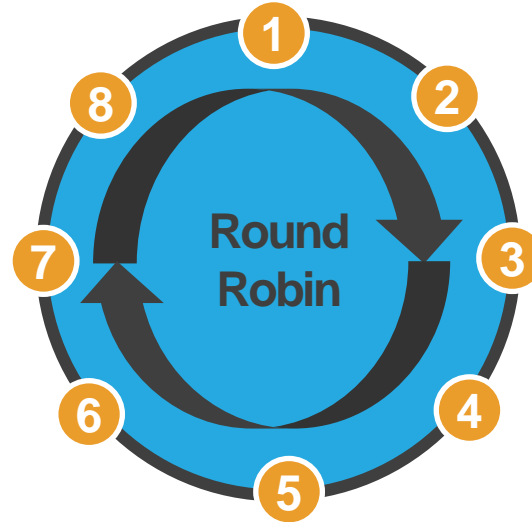
Maximize flash lifetime, performance

Dynamic Control for Multi-Tenant Workloads



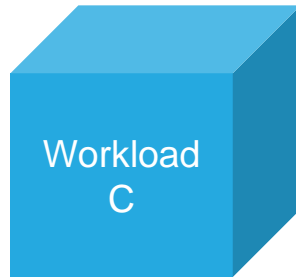
Orchestrate per-workload QoS Domains

Per-I/O Queueing Modes



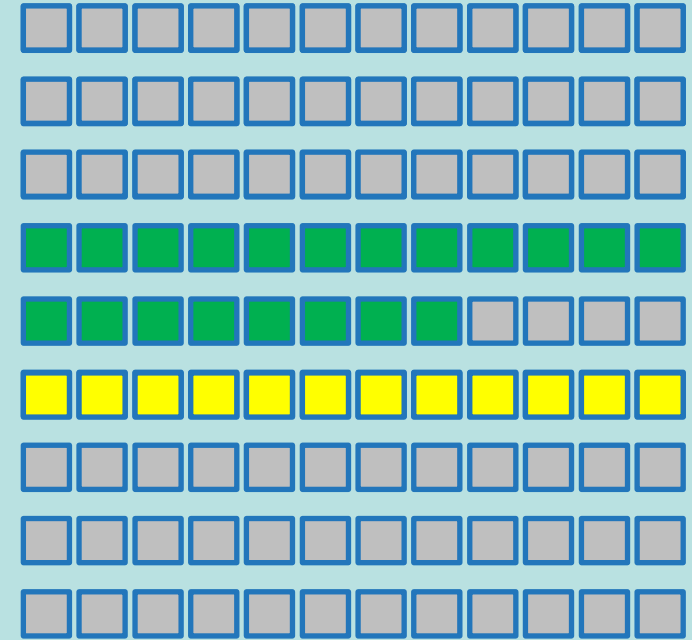
Control multi-tenant latency outcomes

In-drive Copy Offload Capabilities



1 Command

3, 14, 159, 2,
65, 35, 979,
3238, 46, 6,
433, 832, 79,
5028, 841, 971,
6939, 93, 7,
510, 5820,
9749, 4459,
2307, 81, 64,
6286, 20899, ...



**Software-Enabled Flash
Unit**

Minimizes CPU, DRAM, PCIe® Bandwidth

Flash Abstraction



Accelerate flash technology transition

Software-Enabled Flash™ Software

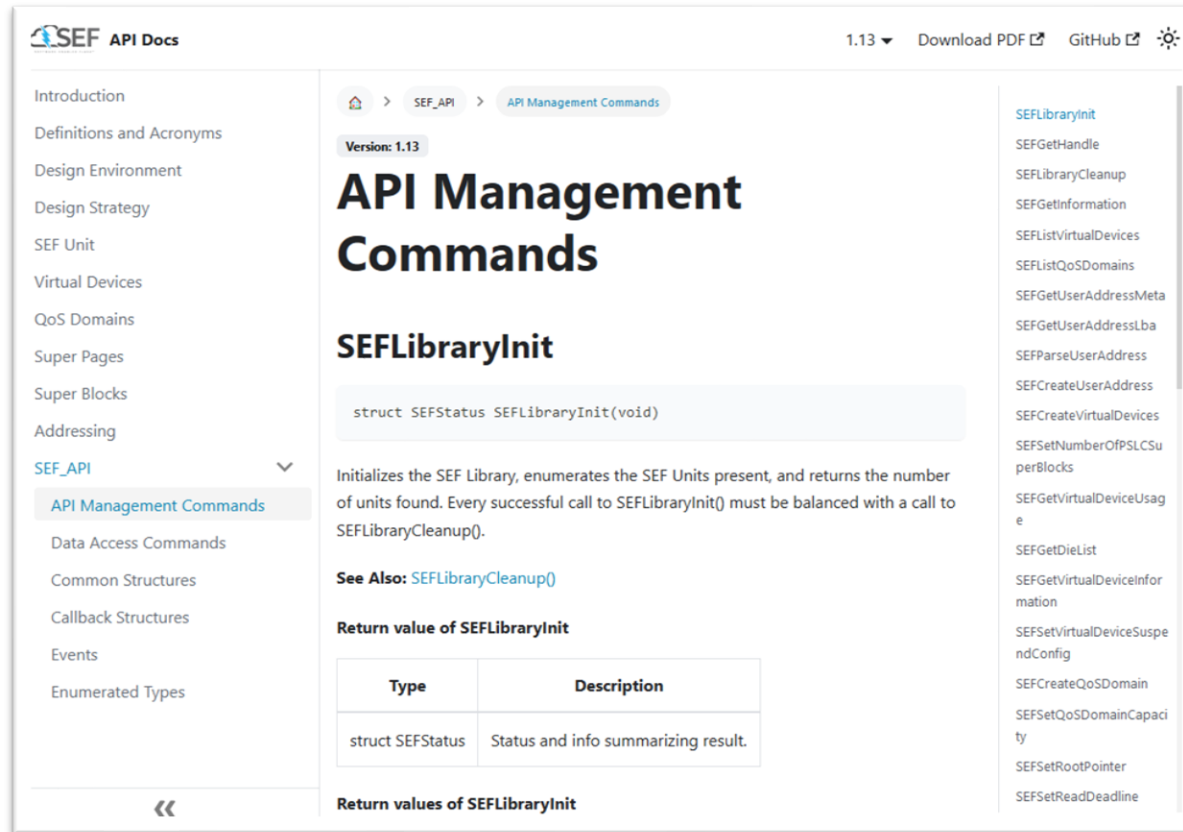
API and SDK to speed adoption and maximize developer results

API for low-level access

SDK for rapid development

Multiple software-defined protocols

Low-Level API for Device Control



The screenshot displays the SEF API Docs interface. The main content area is titled 'API Management Commands' and features a section for 'SEFLibraryInit'. The function signature is shown as `struct SEFStatus SEFLibraryInit(void)`. Below this, a paragraph explains that the function initializes the SEF Library, enumerates SEF Units, and returns the number of units found. A 'See Also' link points to `SEFLibraryCleanup()`. A table lists the return value, `struct SEFStatus`, with a description: 'Status and info summarizing result.' A right-hand sidebar lists various other SEF API functions.

SEF API Docs 1.13 Download PDF GitHub

Introduction
Definitions and Acronyms
Design Environment
Design Strategy
SEF Unit
Virtual Devices
QoS Domains
Super Pages
Super Blocks
Addressing
SEF_API
API Management Commands
Data Access Commands
Common Structures
Callback Structures
Events
Enumerated Types

SEFLibraryInit
SEFGetHandle
SEFLibraryCleanup
SEFGetInformation
SEFListVirtualDevices
SEFListQoSDomains
SEFGetUserAddressMeta
SEFGetUserAddressLba
SEFParseUserAddress
SEFCreateUserAddress
SEFCreateVirtualDevices
SEFSetNumberOfPSLCSu
perBlocks
SEFGetVirtualDeviceUsag
e
SEFGetDieList
SEFGetVirtualDeviceInfor
mation
SEFSetVirtualDeviceSuspe
ndConfig
SEFCreateQoSDomain
SEFSetQoSDomainCapaci
ty
SEFSetRootPointer
SEFSetReadDeadline

Version: 1.13

API Management Commands

SEFLibraryInit

```
struct SEFStatus SEFLibraryInit(void)
```

Initializes the SEF Library, enumerates the SEF Units present, and returns the number of units found. Every successful call to `SEFLibraryInit()` must be balanced with a call to `SEFLibraryCleanup()`.

See Also: [SEFLibraryCleanup\(\)](#)

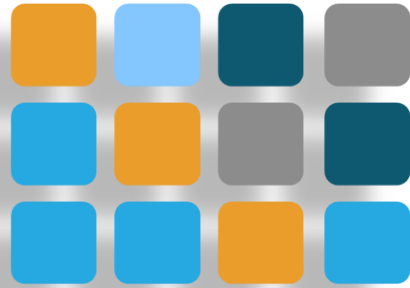
Return value of SEFLibraryInit

| Type | Description |
|------------------|-------------------------------------|
| struct SEFStatus | Status and info summarizing result. |

Return values of SEFLibraryInit

Low-level control without bare metal code

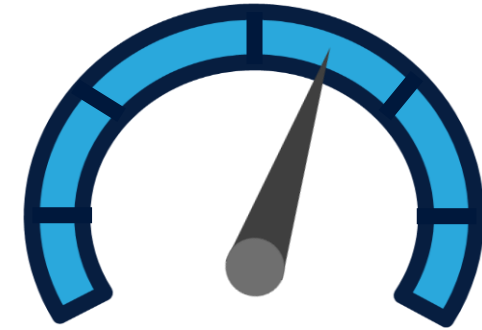
High-Level SDK for Rapid Development



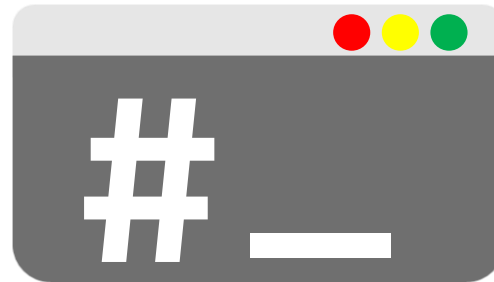
Reference
Flash Translation Layer



Reference
Virtual Drivers



Performance Test Tool
(FIO)

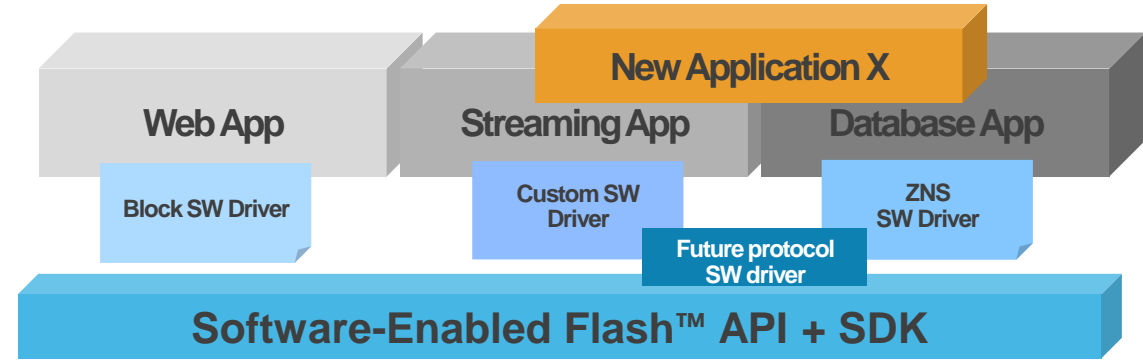
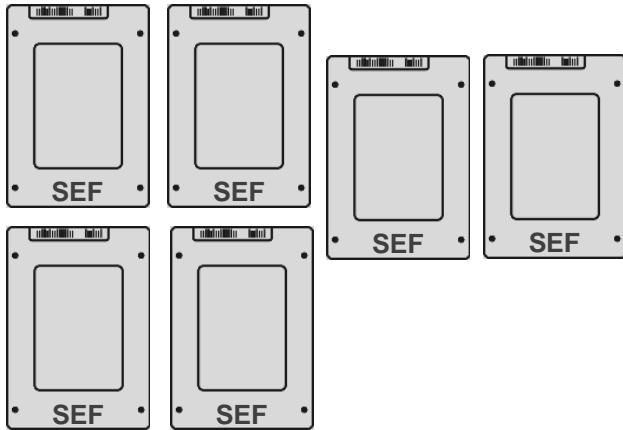


CLI

Accelerate adoption, testing, development

Multiple Software-Defined Protocols

Inventory of identical SEF Units



Simplify SKU management

Get involved!

Visit our booth!



Flash Memory Summit

Join the project

softwareenabledflash.org

**Make flash
Software-
Defined!**



SOFTWARE-ENABLED FLASH™

Definition of capacity: KIOXIA defines a megabyte (MB) as 1,000,000 bytes, a gigabyte (GB) as 1,000,000,000 bytes and a terabyte (TB) as 1,000,000,000,000 bytes. A computer operating system, however, reports storage capacity using powers of 2 for the definition of 1GB = 2^{30} = 1,073,741,824 bytes and therefore shows less storage capacity. Available storage capacity (including examples of various media files) will vary based on file size, formatting, settings, software and operating system, such as Microsoft Operating System and/or pre-installed software applications, or media content. Actual formatted capacity may vary.

All company names, product names and service names may be trademarks of their respective companies.

Images are for illustration purposes only.

© 2022 KIOXIA America, Inc. All rights reserved. Information, including product pricing and specifications, content of services, and contact information is current and believed to be accurate on the date of the announcement, but is subject to change without prior notice. Technical and application information contained here is subject to the most recent applicable KIOXIA product specifications.