Software-Enabled Flash™ Capabilities and Demonstrations

Scott Stetzer

Strategy Lead for the Software-Enabled Flash Project

VP Technical Marketing at KIOXIA America, Inc.

Agenda

- What is the Software-Enabled Flash (SEF) Project
- Unique software components
- Developer-focused capabilities
- Show-and-tell time!
- How you can get involved

The next evolution of flash is Software-Defined

- Fine-grained data placement
- Workload isolation
- Write amplification reduction
- Latency outcome control
- Advanced queueing methods
- Real-time I/O prioritization
- Customized protocols
- Open source API and SDK

SEF

The Software-Enabled Flash™ Project

An Open, Linux Foundation® managed organization

Open Source

Vendor Neutral

Managed under

THE LINUX FOUNDATION

Open Source Governance Processes

Open Technical Steering Committee (TSC) meetings

Open Source API Available on GitHub®

SDK coming soon

Specified for multiple implementations

Flash technology independent
Controller technology independent

* The Linux Foundation and The Linux Foundation logo design are registered trademarks of The Linux Foundation.
* GITHUB® and the GITHUB® logo design are trademarks of GitHub, Inc., registered in the United States and other countries.

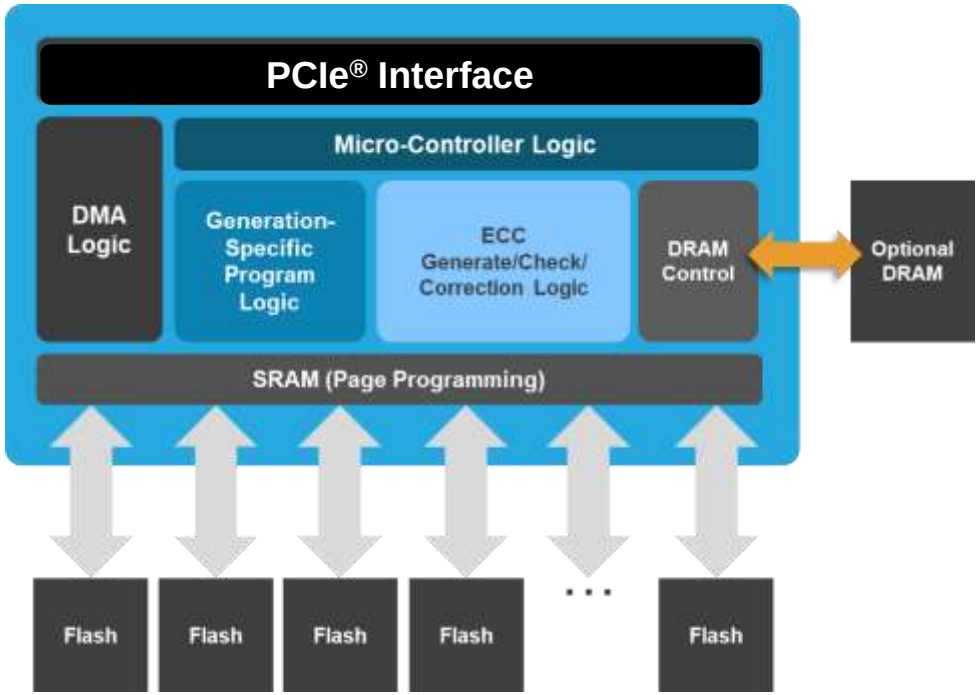
SEF

Antitrust Policy

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.
- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrust-policy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

Unique SEF Hardware Components

Configurable, Multi-Vendor Hardware

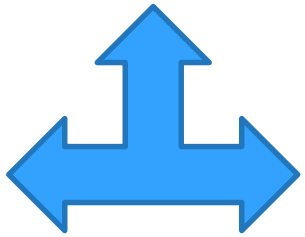


Graphic used with permission of KIOXIA America, Inc.

Vendor Configurable

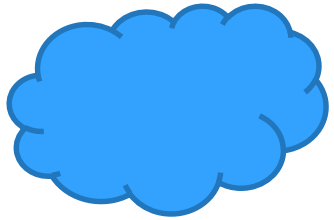
- Flash Technology
- Flash Interface Protocol
- DRAM Buffer vs. Host Buffer vs. Mixed
- Form Factor
- Power Limits

Targeted Hardware Features



Advanced queueing control

Control latencies at the flash operation level

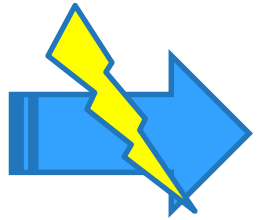


Flash abstraction & management

Simplify porting between flash generations, vendors, and technologies

Low-level hardware partitioning & isolation

Maximum performance decoupling between critical workloads



Advanced on-board copy offload

Minimize CPU and bus management for data movement operations

Unique SEF Software Components

Open Source API and SDKs

CLI with Python® Interpreter	FIO Test Tool	Reference Virtual Device Drivers	Reference Flash Translation Layer (FTL)
Device orchestration and management	Ported to SEF for fast and easy experimentation	No code changes to evaluate SEF in multi-tenant mode	Common block interface to SEF applications

High-Level SDK

Low-Level API

* Python is a registered trademark of the Python Software Foundation.

BSD Licensed Software Development Kit

- C-language based
- 32 + 64 bit, multiple CPU architectures
- Modern Linux® Kernels
- Event Driven Callbacks
- Thread Safe, Lockless Operation
- Modular, Built for Customization

Full Host Source Code

- Reference FTL, CLI, FIO, virtual device driver, Kernel/IO_URING driver

Control for the Developer

SEF Gives Control to the Storage Developer

- Complete physical isolation **control**
  - Software isolation layered on top
- Data placement **control**
  - Including instant-reclaim
- Write Amplification Factor (WAF) **control**
- Latency outcome **control**
- Housekeeping acceleration and **control**
- Software-defined protocol **control**
  - Block, FDP, ZNS, etc.

Hardware and Software Isolation, Data Placement and WAF Control

HARDWARE

Virtual Device

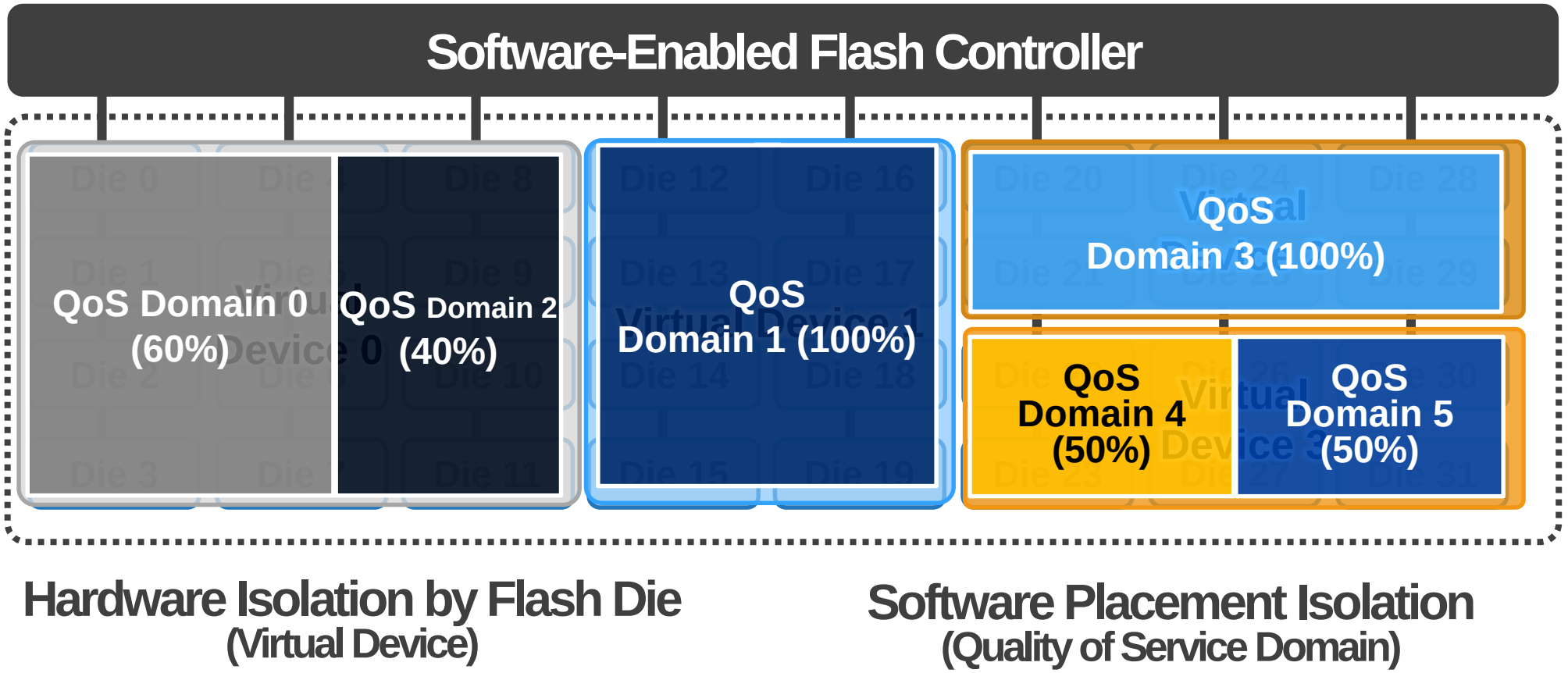
- Die-level isolation
- Complete physical separation of data
- User-configurable at deployment
- May support multiple bits-per-cell (ex: QLC + pSLC)

SOFTWARE

Quality of Service Domain

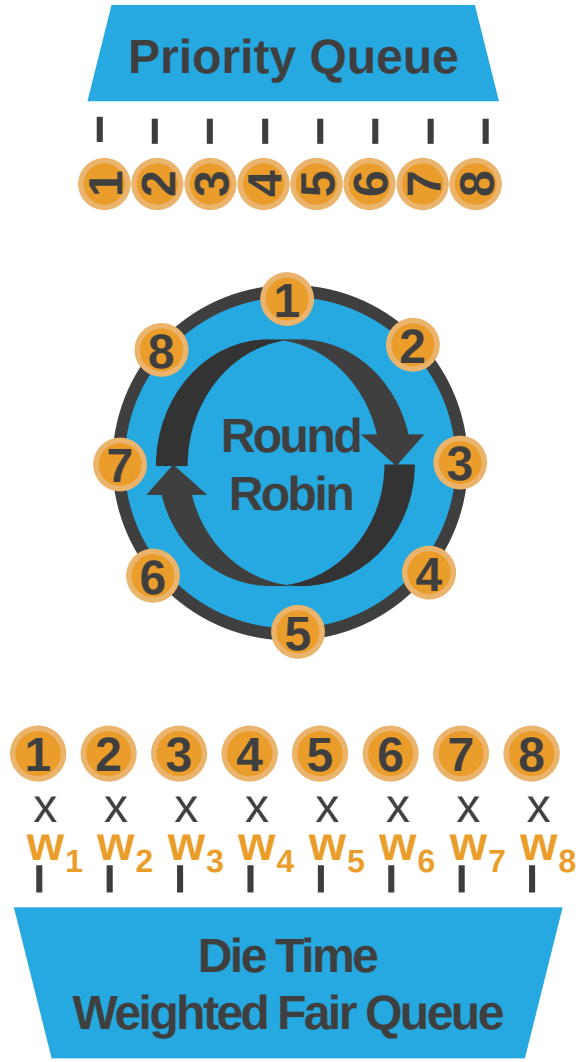
- Workload-level isolation, Placement IDs
- Separation of data by super block
- Isolated garbage collection, overprovisioning and encryption
- Can reduce WAF and support “instant reclaim” for multi-tenant

More on Hardware and Software Based Isolation Control



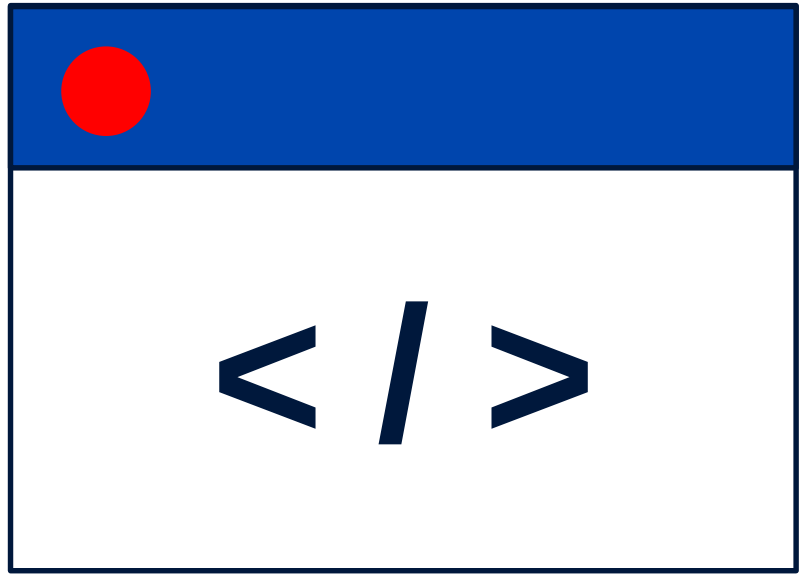
Advanced Queueing for Latency Outcome Control

- **Massively parallel I/O queues**
  - Total separation of read and write paths
  - Minimize head-of-queue blocking
- **Hardware-enforced I/O prioritization**
  - Multiple, programmable scheduling modes
  - Application controlled
- **Die-Time Weighted Fair Queueing**
  - Individual erase, program, read, copy weights



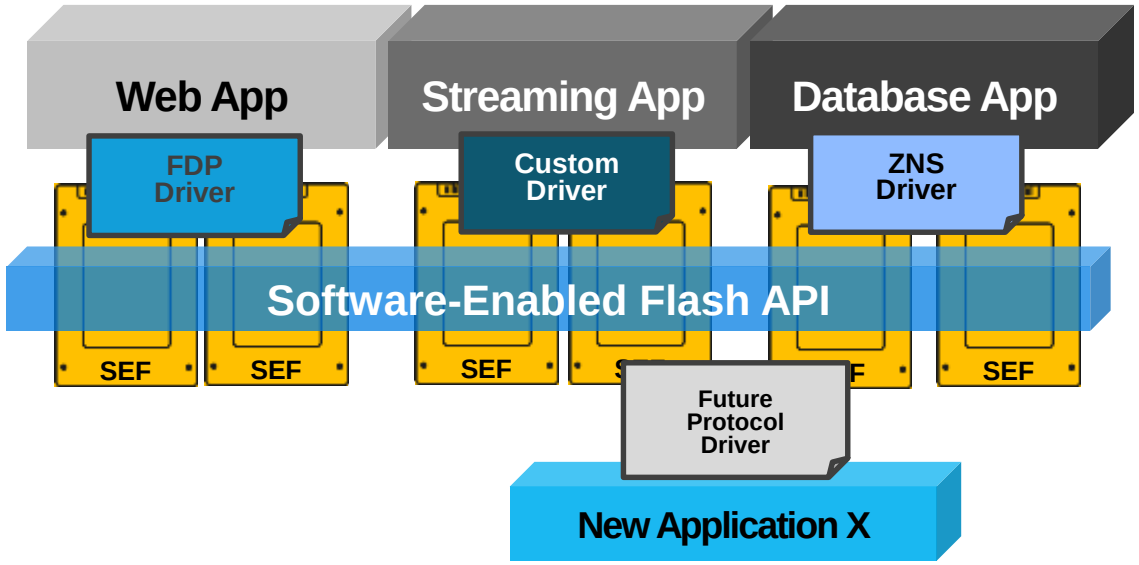
Background Process Control

- Garbage collection managed by application
  - QoS domains ensure that GC in one application doesn't effect others
  - Timing and priority under application control
- On-drive bitmap and list-based copy offload
  - Can do more than just GC (ex: database compaction)
  - No host CPU, DRAM, or busses required
- Flash memory background patrol, too



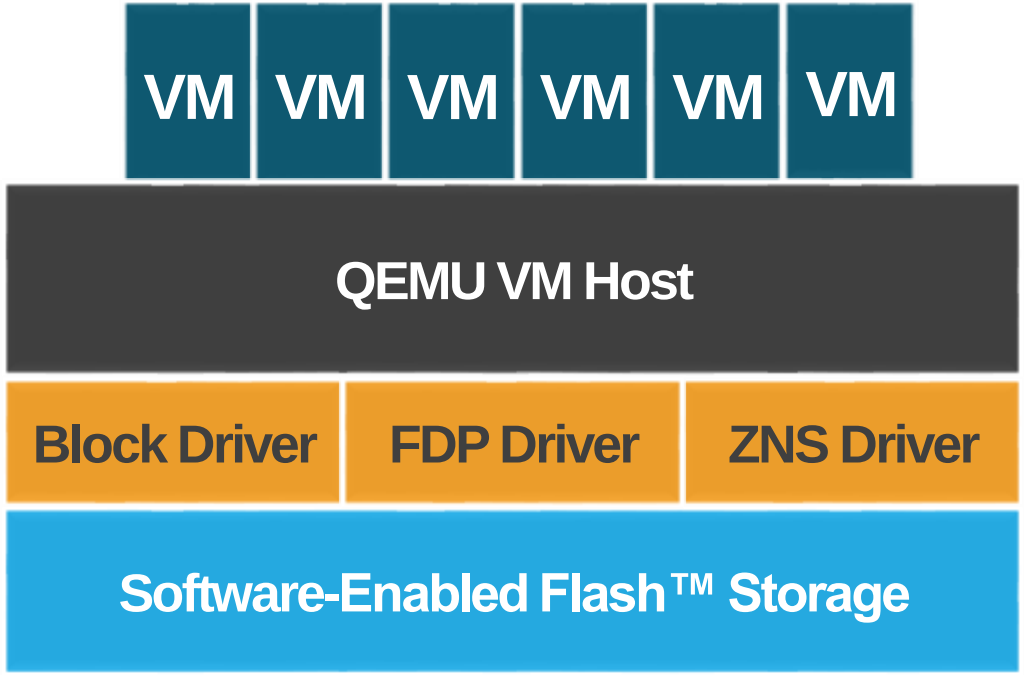
Software-Defined Protocol Control

- Host application controlled interface
  - FDP, Block, ZNS, and simple NVMe reference code included in SDK
- Optimize flash interface in real time
  - Not limited to a single lifetime protocol
- Simplifies sourcing and inventory control
  - “One drive fits all” via software-defined



Reference Virtual Drivers for Software-Defined Protocols

- No guest code changes needed
- Customize overprovisioning per VM
  - Tune for write- or read-optimized
- ZNS, FDP and block-based VMs
- Full data, performance isolation, queueing control
  - Orchestration layer or app managed

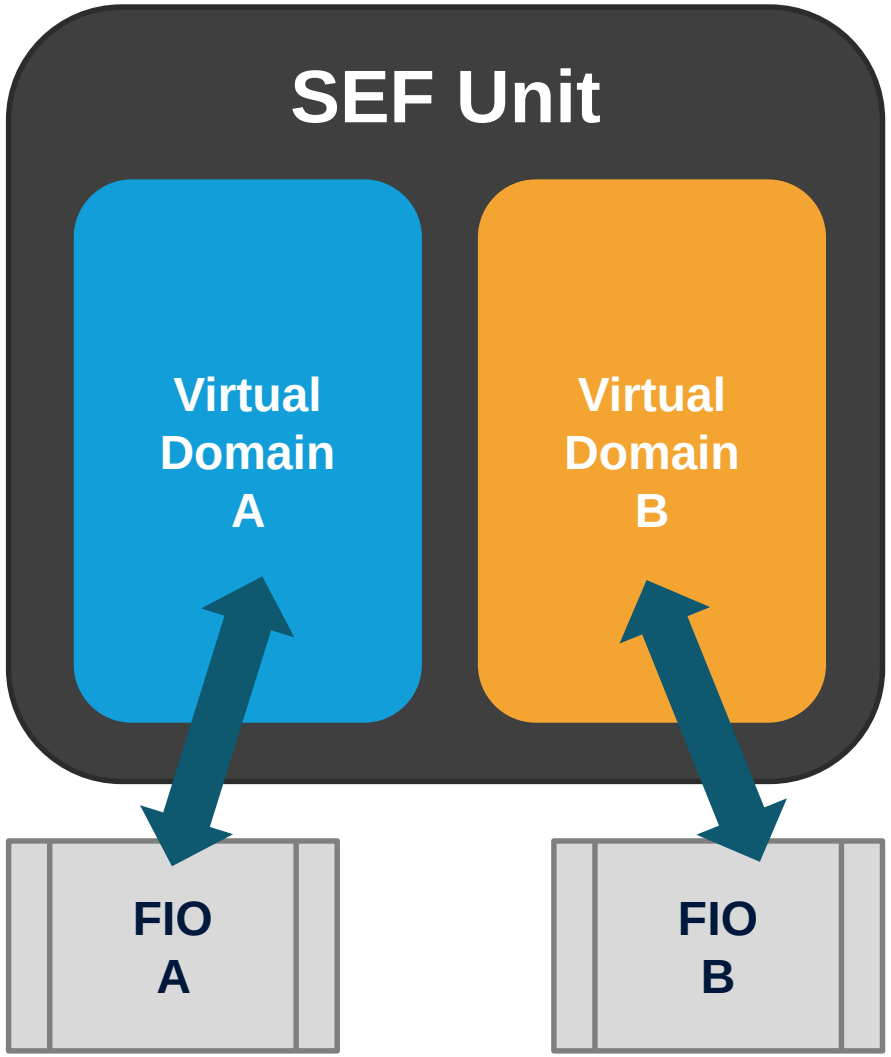


Show-and-Tell

Hardware and Software Based Isolation

Demonstrating SEF Isolation Capabilities

- One physical SEF Unit
- Split into 2 separate Virtual Domains (physical isolation)
- Identical FIO workloads on each Domain
- Jobs started and stopped without interference



SEF Hardware Isolation

Using Self Driving Interface: Running

Stop Demo

- [x] Enable Virtual Device A
- [x] Enable Virtual Device B

Software-Enabled Flash isolates workloads from each other while providing application-controlled latency outcomes

Video used with permission from KIOXIA America
Demonstration recorded on 8/4/23

Demo has sucessfully started

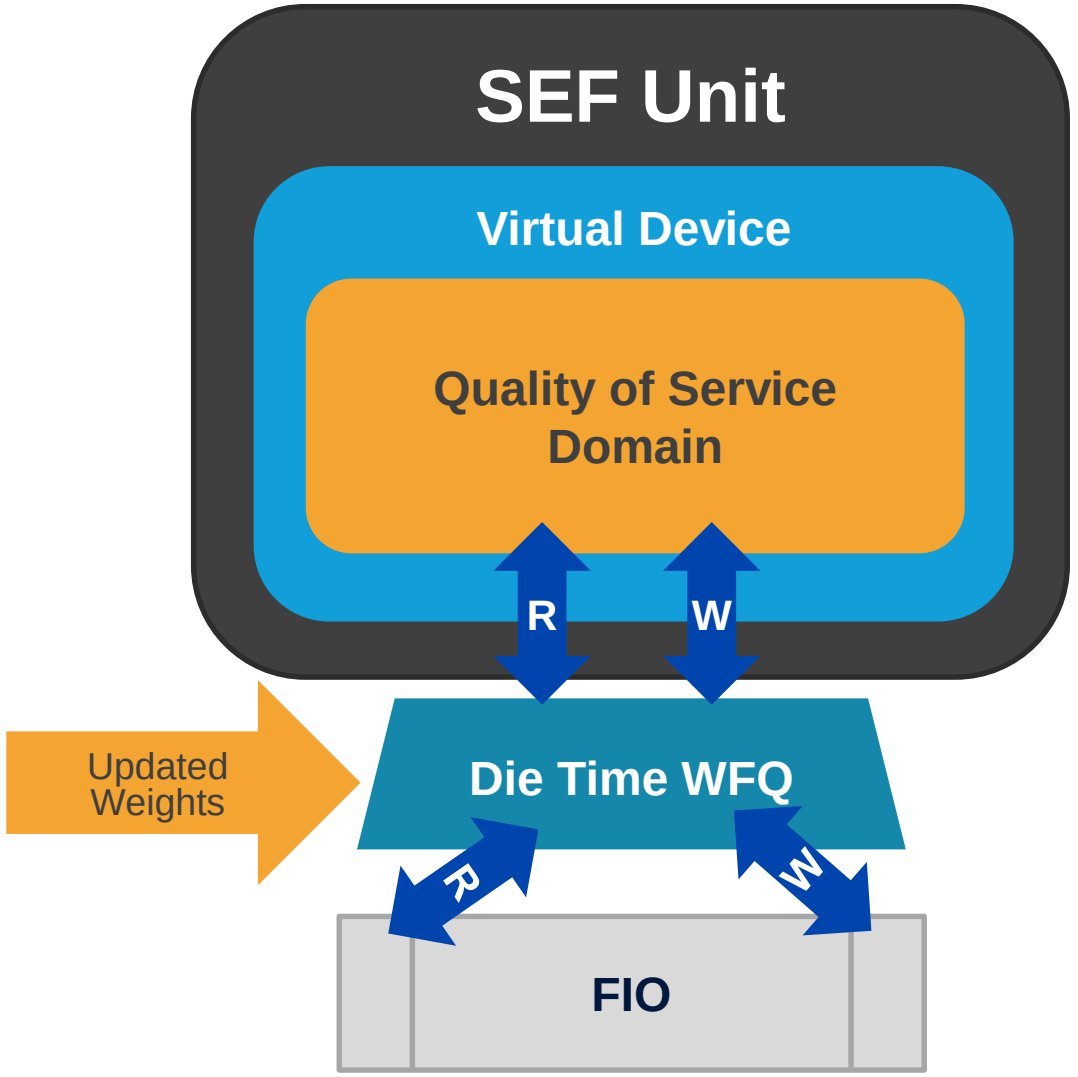
Show-and-Tell

Real-time, application controlled queueing

Demonstrating SEF Queueing Capabilities

- Single SEF Unit
- Single Virtual Device
- Single Quality of Service Domain
- One FIO job (Read and Write)

- Die-Time Weighted Fair Queueing
  - Adjust read and write weights, in real time, while job is running



SEF Weighted Fair Queueing

Using Self Driving Interface: Running

Start Demo

Die Time Balance

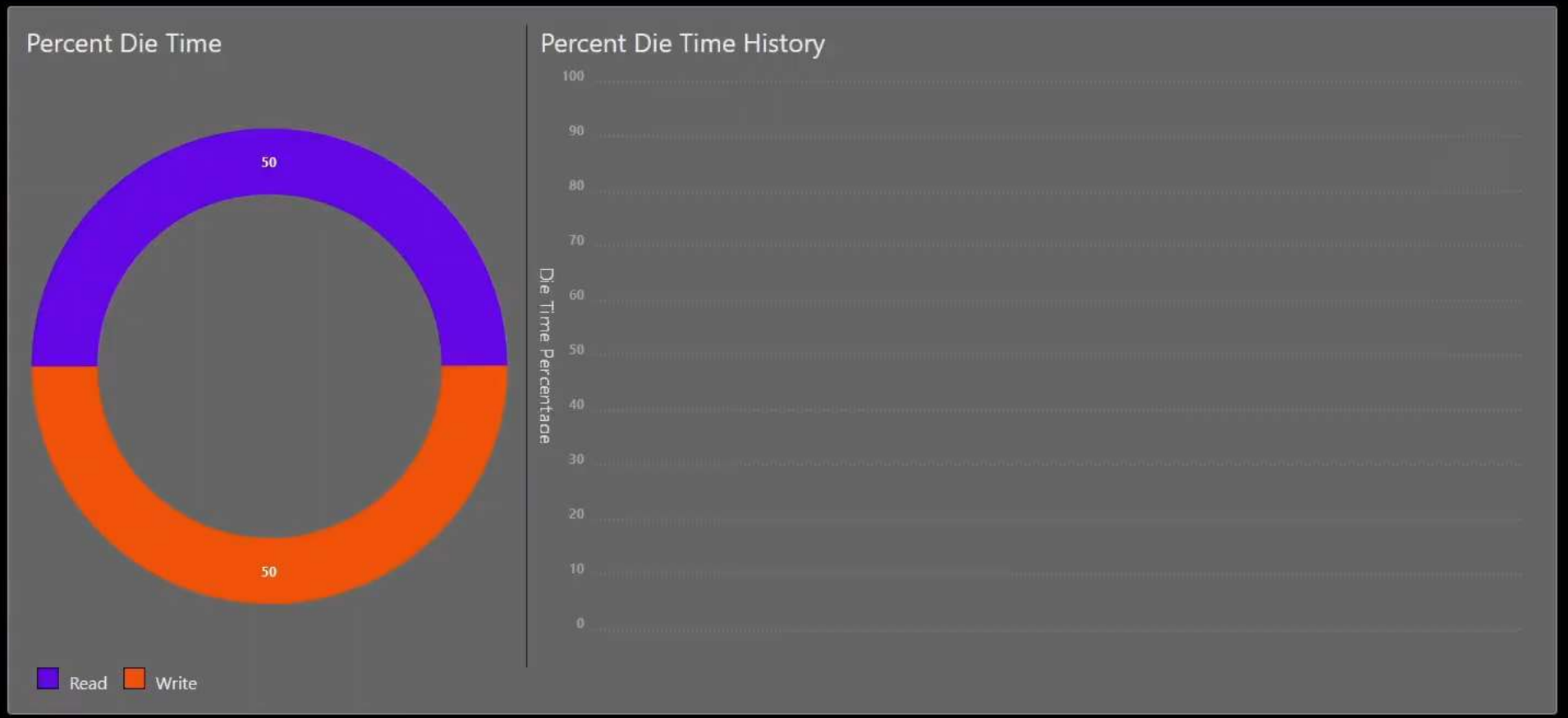
Read Write

90/10 50/50 10/90

Enable Read Enable Write

Software-Enabled Flash allows control over read and write priorities (die times) while preserving the full performance of the device

R/W IOPS Ratio: 0



Video used with permission from KIOXIA America
Demonstration recorded on 8/4/23

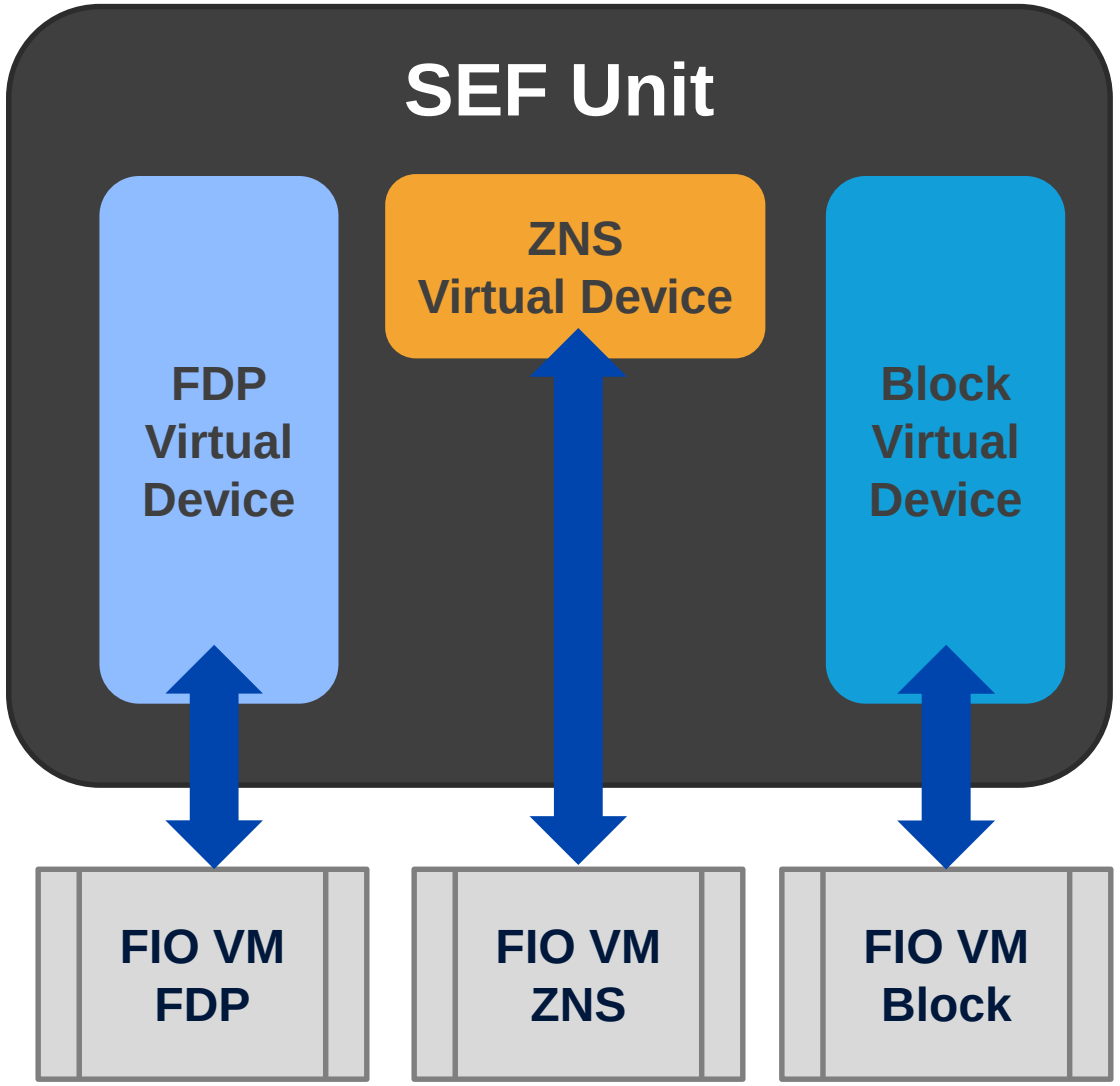
Show-and-Tell

Application controlled, software-defined protocols

Demonstrating SEF Multi-Protocol Capabilities

- Single SEF Unit
- Three Virtual Devices (separate flash die isolation)
  - Unused die visible in this example
- Three VMs with different protocols
  - FDP
  - ZNS
  - Block

- FIO job for each VM, started and stopped independently



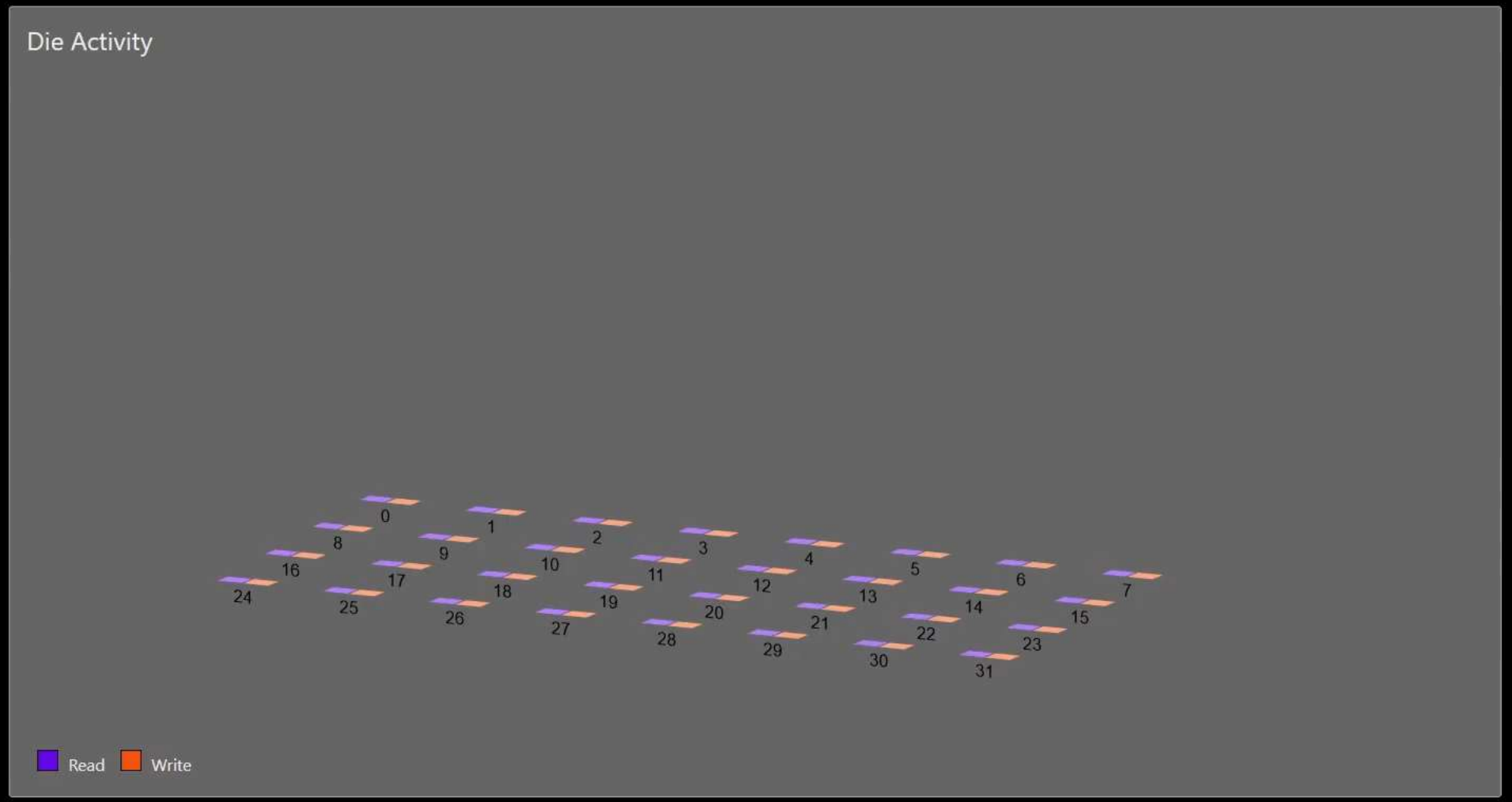
Simultaneous Multiple Software-Defined Protocols with Isolation

Using Self Driving Interface: Running

Stop Demo

- [x] Read/Write Workload (FDP)
- [x] Write Workload (ZNS)
- [x] Read Workload (Block)

Software-Enabled Flash supports standard and application-defined protocols while isolating workloads to individual flash dies for complete control and isolation



Starting workloads with multiple software-defined protocols

Video used with permission from KIOXIA America
Demonstration recorded on 8/4/23

Demo has sucessfully started

SOFTWARE-ENABLED FLASH™

The Software-Enabled Flash™ Project Needs You!

https://softwareenabledflash.org

Legal Notices

The Linux Foundation, The Linux Foundation logos, and other marks that may be used herein are owned by The Linux Foundation or its affiliated entities, and are subject to The Linux Foundation's Trademark Usage Policy at https://www.linuxfoundation.org/trademark-usage, as may be modified from time to time.

Linux is a registered trademark of Linus Torvalds. Please see the Linux Mark Institute's trademark usage page at https://lmi.linuxfoundation.org for details regarding use of this trademark.

Some marks that may be used herein are owned by projects operating as separately incorporated entities managed by The Linux Foundation, and have their own trademarks, policies and usage guidelines.

All other trademarks are the property of their respective owners. Use of such marks herein does not represent affiliation with or authorization, sponsorship or approval by such owners unless otherwise expressly specified.

LF Projects is committed to maintain a positive, professional work environment. This commitment calls for workplaces where participants at all levels behave according to the rules of the following code. A foundational concept of this code is that we all share responsibility for our work environment. https://lfprojects.org/policies/code-of-conduct/

The Linux Foundation is subject to other policies, including without limitation its Privacy Policy at https://www.linuxfoundation.org/privacy and its Antitrust Policy at https://www.linuxfoundation.org/antitrust-policy. each as may be modified from time to time. More information about The Linux Foundation's policies is available at https://www.linuxfoundation.org.

Please email legal@linuxfoundation.org with any questions about The Linux Foundation's policies or the notices set forth on this slide.

THE LINUX FOUNDATION

SEF