KIOXIA

Technical Brief

# KIOXIA Software-Enabled Flash™ Technology:

*Enabling Multi-Protocol Drives that Help Ease Operations and Improve TCO at Cloud Scale*

The compatibility that flash memory had with legacy hard drive protocols helped it revolutionize the data center. While this compatibility simplified the initial adoption of flash memory, it now limits the ways in which it can be used. This has led to the development of multiple unique storage protocols, such as zone- or object-based, with an objective to improve the TCO and storage utilization for certain applications. This proliferation of storage protocols has led to the challenge of sourcing multiple incompatible types of flash storage - deployed and managed in response to changing application needs. Software-Enabled Flash technology presents a new approach to storage protocols that delivers the full potential of flash storage in a software-defined solution while avoiding the pitfalls of acquiring, deploying and managing multiple and incompatible flash storage devices.

## Data Center-Scale Flash Memory Challenges

Data centers are built to run thousands of applications simultaneously, and over their lifetime they will need to accommodate different types of workloads with vastly divergent storage needs. For example, some workloads may require predictable random read performance. Other workloads may require low-latency log-based data updates, while others may need to run client virtual machines (VMs) that have their own unique set of performance and data isolation requirements. The mix of workloads can vary over time depending on the needs of the business - further complicating data center operations. To meet these needs efficiently is even more challenging when each workload requires a specific flash storage protocol.

As the number of workloads that require specific flash storage protocols increase, predicting the optimal mix of flash storage hardware to deploy becomes an exponentially more complicated problem, and not only for data center operations. Project management and development teams may need to juggle future introductions based on the availability of this storage hardware. For example, an application 'ready-to-go-viral' may be starved for storage resources or delayed because the flash memory 'in stock' doesn't implement the right interface protocol. Alternatively, if a new service 'ready-to-go-viral' doesn't grow as expected, large inventories of flash storage could wind up not utilized, which can limit application and revenue growth.

In hyperscale data centers, these application and service related challenges are compounded by sheer scale. Since hundreds or thousands of servers will be deployed per application, it is critical that each part of the compute and storage stack is optimized for the task at hand. Even the underutilization of a small resource can be magnified hundreds or thousands of times, with adverse effects such as increasing the effective cost of an application rollout, wasting large amounts of power and space in the data center, and degrading overall TCO.

The root cause of this difficulty stems from traditional drives that can only be accessed by a single, fixed storage protocol. In some use cases, flash drives emulate legacy hard drive protocols utilizing generic read and write commands. There are other drives that implement protocols to efficiently handle log-style applications, at the expense of block capabilities. In some hyperscale deployments, drives are even custom built to completely implement an application-optimized storage protocol. Regardless of the protocol used, legacy flash drives are generally locked to a single fixed protocol as they are built.

Software-Enabled Flash technology flexibly addresses the incompatible protocol issue by enabling the drives to be completely reconfigured for different protocols under software control. This agility minimizes the painful inventory management and incompatibility problems of traditional fixed protocol drives, and can simplify the supply chain and future planning needs of the data center, resulting in a better TCO for the data center and application.

## Technology Overview

Software-Enabled Flash technology (Figure 1) is a new software-defined way of deploying solid state storage and designed to maximize the value of flash memory for hyperscale users. It combines purpose-built hardware and an open-source software (OSS) API that enables fine-grained control of the entire storage stack -- from the application down to individual die-level scheduling. The hardware is designed for flexibility and can be customized to the needs of hyperscale applications for such key items as flash memory technology and vendor, onboard DRAM, input/output (I/O) path isolation and power loss protection. The open source application programming interface (API) takes advantage of this hardware flexibility to provide applications with fine-grained latency control, tenant isolation, CPU offload and the power to use flash memory in exceptionally efficient ways. It also frees applications from needing to manage flash memory differences between flash generations and vendors. This unique hardware and software abstraction fundamentally redefines the relationship between the host and solid state storage.

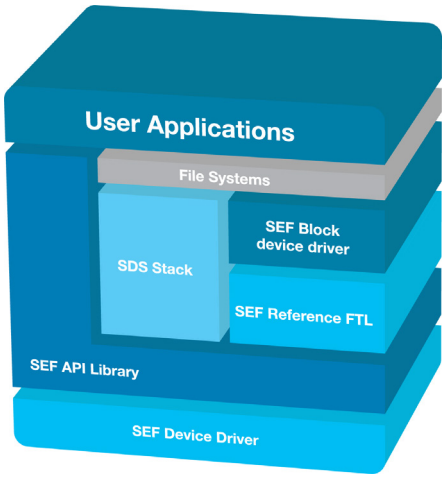


*Figure 1: Software-Enabled Flash host software block diagram*

## Enabling Multiple Protocols via Software

Software-Enabled Flash technology moves drive protocol provisioning out of the factory and into the control of the application. Depending on the application's current requirements, a Software-Enabled Flash technology storage device can be deployed to support multiple software-defined protocols including simple block interfaces, zone-based storage interfaces or fully customized interfaces that can prioritize (in hardware) different Quality of Service (QoS) classes with an application-specific flash translation layer (FTL).

## Multi-Protocol Flexibility for Data Center Operators

The power and flexibility of Software-Enabled Flash technology means that instead of having to stock and predict future needs for multiple types of single-protocol drives (Figure 2), the data center only needs to maintain a common pool of Software-Enabled Flash drives.

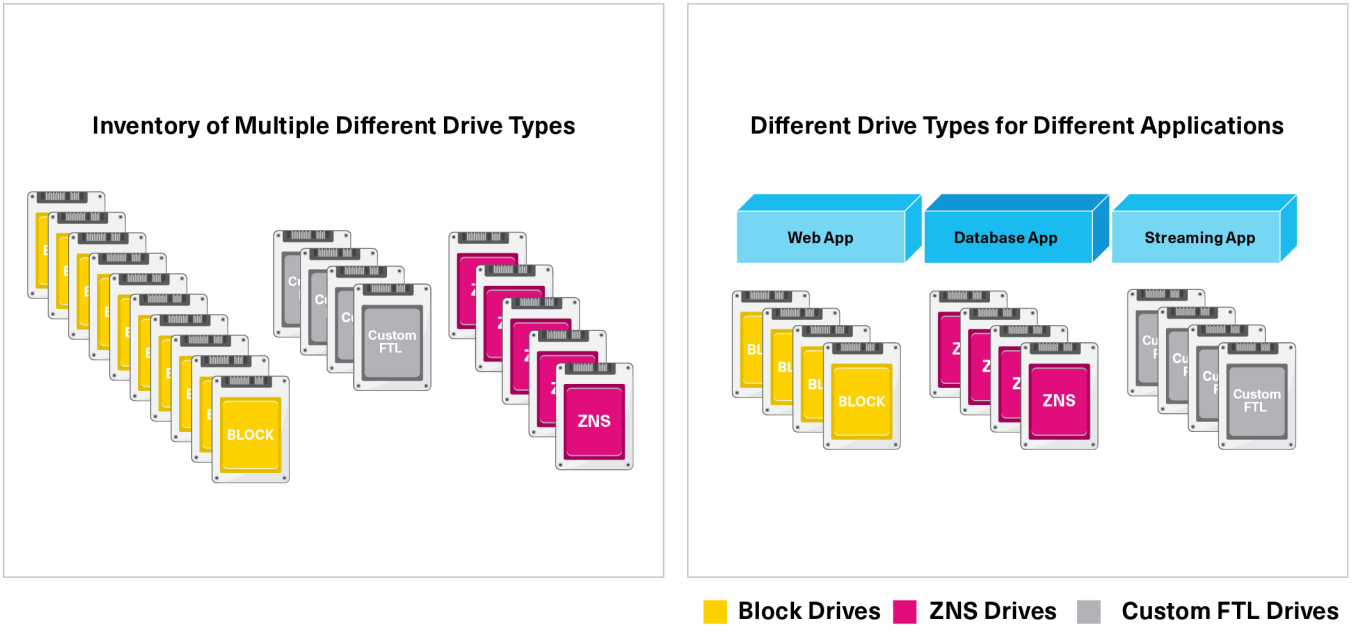


*Figure 2: Traditional drives require support for many different protocols that must be kept on-hand in the data center*

As flash storage is deployed to new applications, or as applications are retooled, the installed Software-Enabled Flash drives can be adapted to new protocols as required, maximizing their utilization and minimizing inventory management complexities (Figure 3).

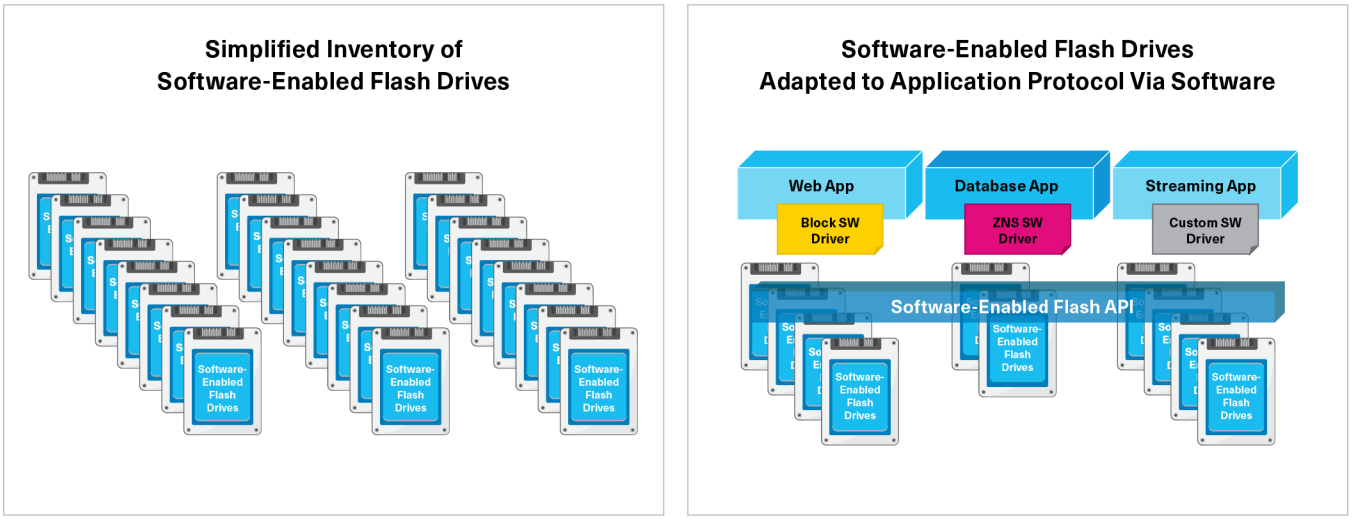


*Figure 3: Software-Enabled Flash technology enables a single drive type to be adapted via software to the specified application*

A good example of this adaptation occurs in log-based database applications. In this use-case, a Software-Enabled Flash drive can be adapted to the application with log-based storage software that offloads portions of the database background operations (such as log compaction). This enables CPU cycles and host memory to be freed up to address more pending tasks or operations.

Virtual machine servers can deploy Software-Enabled Flash drives with a standard FTL whereby the VMs can be isolated from each other down to the individual flash-die or flash block level. This can reduce or even eliminate the noisy neighbor problem where a particular I/O-intensive VM can hurt the responsiveness of other co-located VMs. Individual VMs can have their QoS and flash overprovisioning defined by the hypervisor and managed by the Software-Enabled Flash drive itself. VMs also can be configured to never share a single flash block and to use their own unique encryption key, helping to protect critical customer data. This enables the same storage device to support multiple service classes simultaneously, dynamically and securely.

## Enabling Agile Application Development

The flexibility of Software-Enabled Flash technology and its software-driven approach enables developers to build custom hyper-tuned FTLs that precisely complement their workloads. For example, the difference between a 'write-optimized' drive and a 'capacity-optimized' drive is often just the amount of the flash memory overprovisioning. Therefore, a 'write-intensive' application can configure a Software-Enabled Flash drive for a higher amount of overprovisioning to enable greater sustained write performance. This type of application could also use the built-in scheduling control to prioritize and allocate flash die time to the appropriate ratio of writes to reads. Conversely, a minor API call change to a 'read-intensive' application could minimize its overprovisioning in order to maximize the amount of flash memory that is available for data storage. A Software-Enabled Flash drive can support either application, or both simultaneously, without deploying custom storage hardware.

## Summary

Software-Enabled Flash Technology support of multiple protocols present a unique opportunity to simplify the storage supply chain for data centers. Instead of managing demand and usage for a large mix of different drive types, data center operators can stock a single SKU of Software-Enabled Flash drives that can be configured by individual applications at the time of need, and that uses the appropriate storage protocol. This approach can reduce overhead and improve TCO for cloud applications, can improve the effectiveness of applications by enhancing QoS, hardware isolation and performance management capabilities, and can be available on a per-application or per VM basis.

Software-Enabled Flash technology has been released to the OSS community with the API definition and specification document available and downloadable from the KIOXIA repositories on its GitHub® site[1] that includes https://github.com/kioxiaamerica and https://github.com/KioxiaAmerica/SoftwareEnabledFlash.

For more information about Software-Enabled Flash technology visit https://softwareenabledflash.com.

**NOTES:**

[1] The GitHub branch platform includes a website and cloud-based service that stores and manages the code that a company develops, and tracks and controls any changes to their code. GitHub is an exclusive trademark registered in the United States by GitHub, Inc.

**TRADEMARKS:**

GitHub is a registered trademark of GitHub, Inc. All other company names, product names and service names may be trademarks or registered trademarks of their respective companies.

**DISCLAIMERS:**

 Information in this tech brief, including product specifications, tested content, and assessments are current and believed to be accurate as of the date that the document was published, but is subject to change without prior notice. Technical and application information contained here is subject to the most recent applicable KIOXIA product specifications.