



The Scale Computing Platform™ Difference

Modern infrastructure that's easier to run —
day one through years to come

From the Experts at
Scale Computing™



Table of Contents

Introduction.....	3
Reducing Lifecycle Burden Through Innovation	4
Real World Requirements.....	4
Where Lifecycle Burden Comes From	5
How is Scale Computing Platform™ Edge Computing Simpler?	5
Deploy and Stand Up Infrastructure	6
Operate Day to Day.....	7
Secure Operations Without Added Complexity.....	8
Protect and Recover.....	9
Maintain, Patch, and Keep Consistency.....	10
Scale and Refresh	11
Storage Efficiency and Architecture	11
Lower Total Cost of Ownership.....	14
Conclusion	15
Real-World Requirements: Simplicity Checklist.....	15

Introduction

Modern infrastructure isn't a clean slate. Most organizations are running a mix of long-lived, business-critical applications alongside newer services and initiatives, often across a blend of centralized environments and remote locations. And while the technology choices vary, the operational pressures are consistent: limited IT time, limited on-site support, and rising expectations for uptime.

That puts a spotlight on day-to-day realities: keeping applications available, recovering quickly from disruption, and maintaining consistent performance as needs change. Too often, modernization efforts add layers until the "new" environment becomes harder to run than what it replaced: more components, more consoles, more patching cycles, more vendor coordination.

Many organizations are also introducing AI-enabled applications, often starting with inference and automation use cases that add new operational demands around consistency, updates, and performance.

This paper looks at a different approach: a platform designed to reduce lifecycle burden through tighter integration and smarter automation. We'll break down the architectural choices that matter in real operations, how integration impacts simplicity, resiliency, efficiency, and manageability, and why those differences become more important as environments grow and requirements get stricter.

	Desktop & Server Virtualization	Converged/Hyperconverged Infrastructure	Multicloud Software Platform (Previously Edge Computing)	Hybrid Cloud Infrastructure
2025	N/A	N/A	N/A	Scale Computing
2024	N/A	N/A	N/A	Scale Computing
2023	N/A	Scale Computing	Scale Computing	
2022	N/A	Scale Computing	Scale Computing	
2021	N/A	Scale Computing	Scale Computing	
2020	N/A	Scale Computing	Scale Computing	
2019	Scale Computing	Scale Computing		
2018	Dell/VMware	Dell/EMC		
2017	Dell/VMware	Cisco		
2016	Dell/VMware	HPE (Simplivity)		
2015	Dell/VMware	Dell/EMC		
2014	Microsoft	Dell		



Reducing Lifecycle Burden Through Integration

Modernization isn't only a technology decision; it's an operating model decision. Organizations support a mix of legacy and newer applications, often with lean IT teams and limited time for routine maintenance. In that environment, the biggest risk isn't personal performance, it's operational complexity: too many tools, too many dependencies, and too many "hands-on" tasks required to keep infrastructure stable.

Lifecycle burden spans everything from initial deployment, daily monitoring and troubleshooting, planned maintenance, upgrades, scaling, hardware refresh, and data recovery. When infrastructure is assembled from loosely connected components, each lifecycle event tends to require more coordination, more manual steps, and more specialized knowledge. Integration reduces that burden by simplifying how the core pieces work together, how they're managed, and how they're updated over time.

Real World Requirements

Regardless of where an organization is on the modernization curve, the operational requirements are consistent:

- **Availability and recovery by design.** Environments need to stay online through routine issues and recover quickly from disruptions—without requiring complex processes or specialized expertise.
- **Ease of management.** Fewer consoles, fewer "places to look," and fewer handoffs between teams and vendors.
- **Automation that reduces routine work.** The platform should handle everyday lifecycle tasks reliably, so IT isn't spending nights and weekends babysitting infrastructure.
- **Predictable updates and lifecycle maintenance.** Updates shouldn't require assembling a sequence of independent upgrades across firmware, storage layers, hypervisors, and management tools.
- **Straightforward scale and refresh.** Adding capacity or refreshing hardware should be an extension of everyday operations, not a redesign.
- **Security without operational sprawl.** Modernization shouldn't expand the attack surface. Secure access and consistent operations should be built in, without adding a patchwork of remote-access tooling, special network paths, or one-off processes.

The goal of modernization is not just "new infrastructure." It's infrastructure that is easier to run from day one through year five and beyond.

“Coming from hypervisor worlds filled with VMWare and Hyperv, the full-stack hyperconverged infrastructure software developed by Scale Computing is user-friendly and very intuitive to use. It's reliable from VM provisioning to maintenance, to replication, to restores. It simply does everything that a hypervisor infrastructure should do without needing a multitude of licenses or a litany of training/classes to use the product.”

Gartner.
Peer Insights.

Where Lifecycle Burden Comes From

Lifecycle burden increases when infrastructure relies on separated layers that were not designed to operate as a single, unified system..

Split Ownership and Split Tooling

Many solutions that describe themselves as hyperconverged infrastructure rely on third-party hypervisors, which introduces a second vendor relationship, separate licensing, and separate tooling for core virtualization operations. When management is divided between multiple consoles—one for virtualization and another for storage—the “simple” tasks of day-to-day operations become slower and more error-prone.

“Not Really” Converged Architectures

So-called “disaggregated HCI” puts the HCI label on architectures where storage remains separate from compute. That increases the number of systems to deploy, scale, refresh, and support, which often raises the minimum footprint well beyond what many environments need.

Virtual Storage Appliance (VSA) Overhead and Layered Complexity

Some architectures add a virtual storage appliance layer to present storage to the hypervisor. That can create unnecessary resource overhead and additional layers to manage, especially when storage is effectively emulated by multiple software components rather than integrated directly into the platform’s core architecture.

Manual Storage Configuration

When a solution requires manual storage configuration before workloads can even run, it adds time, risk, and operational dependency on specialized skills—both at deployment and throughout ongoing management.

How is Scale Computing Platform™ Simpler?

Simplicity matters most during the lifecycle moments that consume real time: deployment, day-to-day monitoring, troubleshooting, maintenance, upgrades, and scaling. When infrastructure requires multiple consoles, manual configuration steps, or separate update processes across different layers, lifecycle work expands, and so does risk.

[*Scale Computing Platform™*](#) edge solutions is designed to reduce that lifecycle burden through tighter integration and automation. SC//Platform™ is delivered as an appliance that brings together the components needed to run virtual machines and scale out as needed. [*Scale Computing HyperCore™*](#) virtualization suite runs on the appliance and provides the core virtualization and infrastructure capabilities. For organizations operating multiple clusters, [*Scale Computing Fleet Manager™*](#) edge orchestration solutions adds a cloud-hosted layer to securely stage, manage, and monitor fleets of clusters running SC//HyperCore™.

In the sections that follow, we’ll highlight where that operational simplicity shows up in practice, from getting started quickly, to minimizing routine management effort, to maintaining resiliency, updates, and growth without disruption.



Deploy and Stand Up Infrastructure

Modernization often stalls before workloads ever move, not because teams can't rack hardware, but because "day one" turns into a checklist of prerequisites: multiple installs, multiple consoles, storage setup, and the coordination required to make separate layers work together.

SC//Platform™ is designed to reduce that first-wave friction so teams can go from unboxing to creating VMs quickly and repeatably.

A Straightforward Path from Hardware to First Workloads

Aside from racking and cabling, initial setup is intentionally simple: assign IP addresses, enter a license code, and point a web browser to the appliance to complete configuration. This browser-based model reduces the "management workstation" problem. There are no client components to install or maintain, and day-to-day administration begins immediately with a standard HyperCore UI experience.

When teams need repeatable deployments across many sites or want to minimize the need for hands-on IT presence, SC//Fleet Manager™ adds a deployment model designed for scale. [With zero-touch provisioning \(ZTP\), organizations can centrally stage thousands of clusters](#) from a single pane of glass, from node purchase and installation through powering on, so that when systems are powered on, they automatically provision, initialize, and register with SC//Fleet Manager™

Cluster Formation That Mirrors Real Operations

Creating a cluster follows a similar simple operational pattern. Additional appliances are pointed to the IP address of the first appliance and joined using a straightforward join command. That matters for both initial deployments and growth, because the steps to expand capacity mirror those used to stand up the environment in the first place, reducing "special-case" procedures that only appear during expansion or refresh.

Fewer Prerequisites, Fewer "First-mile" Bottlenecks

Deployment complexity often lies in prerequisite configuration, particularly in storage and management-layer setup. SC//HyperCore™ is designed so teams don't have to complete a manual storage configuration to get started (we'll cover the architectural reasons later in the paper). It also avoids the need to install a third-party hypervisor or stand up a separate third-party management console before you can begin operating the environment.

What "Quick Deployment" Means in Practice

Because the platform reduces prerequisites and consolidates setup steps, a three-node cluster can be racked, cabled, and configured in less than an hour with live VMs running.

Deployment at Scale with Zero-Touch Provisioning (ZTP)

ZTP helps reduce deployment time by 90% and variability across environments, and it supports consistent rollouts without requiring the same level of onsite expertise everywhere. [Read More](#).

Operate Day to Day

Once workloads are running, the real cost of infrastructure shows up in the everyday moments: checking health, responding to alerts, provisioning resources, and answering simple questions like “Is performance normal?” or “Did anything change?” The more consoles and toolchains teams have to stitch together, the more time they spend correlating signals instead of keeping applications available.

SC//Platform™ is designed to keep daily operations straightforward by consolidating core infrastructure functions and making visibility and routine actions accessible without additional layers.

One Interface for the Core Infrastructure Stack

Day-to-day administration starts with the SC//HyperCore™ web-based UI, where teams manage servers, virtualization, storage, and data protection in one place, rather than bouncing between separate consoles for the hypervisor and storage layers.

Live Visibility and Alerting Without Tool Sprawl

Operational simplicity depends on being able to see what’s happening quickly and continuously. SC//HyperCore™ provides live management and alerting for clusters and individual appliances, with additional views designed to support monitoring across multiple environments.

Self-healing Automation and High Availability for Routine Issues

Routine failures shouldn’t require heroics. The [Autonomous Infrastructure Management Engine™](#) (AIME) self-healing automation detects and responds to common infrastructure fault conditions, keeping environments stable without requiring specialized expertise or constant hands-on intervention. The benefit of owning the full computing stack (from storage to hypervisor) is tighter integration between components, enabling the platform to monitor and automatically manage system behavior more closely.

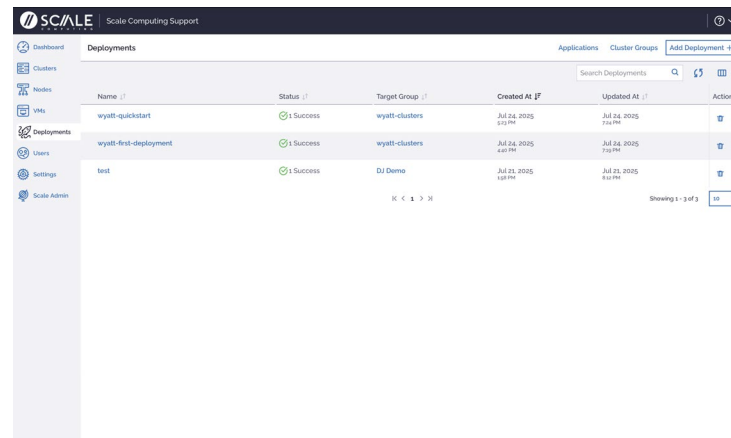
Self-healing automation is most effective when paired with built-in high availability clustering to reduce the operational impact of issues that would otherwise trigger escalations, after-hours work, or manual recovery procedures. SC//HyperCore™ is designed so routine component failures don’t require complex intervention: high availability clustering helps keep workloads running when underlying infrastructure events occur, while AIME self-healing detects and responds to common fault conditions to stabilize the environment.

Central Oversight When You Have Multiple Clusters

For organizations running more than one cluster, Scale Computing Fleet Manager adds a cloud-hosted layer to securely stage, manage, and monitor SC//HyperCore™ clusters. This helps teams standardize how they monitor infrastructure health and reduce the manual effort of checking each environment individually.

Keeping Applications Consistent with Application Lifecycle Management (ALM)

Infrastructure maintenance is only part of the lifecycle burden. Applications introduce their own versioning, rollout, and update complexity. Scale Computing has extended SC//Fleet Manager™ with Application Lifecycle Management (ALM) capabilities designed to manage application deployment and ongoing changes (including updates) across large environments from a centralized control plane.



Supporting AI Applications Without Adding Operational Burden

AI-enabled applications introduce a new kind of operational complexity: more frequent application updates, tighter dependency management, and the need to keep versions consistent as environments scale.

Automation Hooks for Advanced Operations

For teams that want to integrate platform operations into existing processes or automate repeatable tasks, SC//Platform™ provides APIs that support both cluster-level and fleet-level workflows. SC//HyperCore™ includes a [fully documented set of REST APIs](#) accessible from within the HyperCore UI, enabling programmatic management and automation where appropriate.

For broader, multi-cluster environments, [fleet-level APIs](#) provide a centralized way to work with operational data and integrate with existing tools and workflows at scale—supporting automation, reporting, and connections without treating each cluster as a one-off integration project.

Secure Operations Without Added Complexity

Modernizing infrastructure shouldn't mean expanding the attack surface. As environments grow—whether that means more systems, more sites, or simply more applications—security and operations become tightly linked. The more separate tools and access methods you add to “make management work,” the harder it becomes to keep access consistent, controlled, and maintainable over time.

SC//Platform™ is designed to help teams simplify operations without relying on a patchwork of add-ons for day-to-day administration. That includes reducing the need for separate consoles and minimizing the number of “special paths” administrators must keep open just to manage infrastructure.

A practical example is remote administration. Instead of requiring on-cluster agents, jump boxes, or traditional VPN access to reach management interfaces, SC//Fleet Manager™ includes Secure Link, which is designed for out-of-the-box use and reduces the risk that comes with manual setup and ongoing maintenance of traditional remote management tools that can provide wide levels of access to remote networks and devices.

AI Simplicity in Action

Deploy once, roll out consistently: Standardize AI application versions and updates across environments with ALM, rather than managing site-by-site change.

Security also shows up in lifecycle work. Keeping systems current is one of the most important, and most operationally painful, tasks in IT. With SC//HyperCore™, firmware and software updates are automated and can be performed on a cluster without taking workloads offline, helping teams reduce maintenance overhead while staying current as environments evolve.

For detailed security architecture, controls, and program information, refer to the [Information Security with SC//Platform™](#) white paper.

Protect and Recover

Modernization efforts only succeed if they improve operational outcomes, not just on good days, but when things go wrong. Protection and recovery can't be an afterthought or a collection of bolt-ons; it has to be part of the platform experience so teams can apply consistent policies, execute recovery actions quickly, and avoid brittle, one-off procedures during an incident.

SC//HyperCore™ includes a native Data Protection Suite with snapshots, remote replication, remote failover, failback, and recovery options for full VMs or individual files. Because these capabilities are integrated, teams can align protection policies and recovery workflows with day-to-day operations without stitching together separate data protection tools.

Replication and Failover Designed for Real Recovery Workflows

Replication is a key part of resilient operations, and in SC//HyperCore™ it's tied directly to snapshot-driven protection. VM snapshots are used throughout HyperCore for data protection features such as replication and quick VM recovery, enabling a consistent approach to restore points and recovery actions.

With native support for remote failover and failback, teams can execute disaster recovery workflows with fewer moving parts, reducing the operational friction that often slows recovery when multiple products and consoles are involved.

Snapshot-driven Protection Designed for Fast Recovery

Snapshots are a foundational mechanism for protection and recovery workflows. In SC//HyperCore™, VM snapshots are used throughout the hypervisor for data protection features like replication and quick VM recovery. Because snapshots are used frequently, they are taken efficiently—and up to 5,000 snapshots may be stored per VM, providing granular restore points when needed.

Cloning That Supports Operational Flexibility

Beyond significant recovery events, teams also benefit from everyday operational tools that reduce risk and speed change. SC//HyperCore™ supports VM cloning without child-parent dependence on the original VM, so clones are independent; deleting the original VM (or other clones) does not affect the cloned VM. That independence supports safe testing, quick rollback strategies, and faster operational workflows without introducing fragile dependencies.

For organizations that already standardize on enterprise backup products for policy, retention, or compliance, SC//Platform™ also supports integrations with leading third-party backup providers, so teams can keep existing backup workflows without adding operational complexity to the core platform experience.

Maintain, Patch, and Keep Consistency

As environments grow, the maintenance work tends to grow faster—patching across multiple layers, coordinating downtime windows, tracking version drift, and repeating the same steps across systems. SC//Platform™ is designed to reduce that operational overhead by integrating infrastructure updates and automating work that typically stretches maintenance into nights and weekends. With SC//HyperCore™, system updates can be fully automated and performed on a cluster without taking workloads offline. The update process covers both firmware and software across system components, rather than forcing separate update processes for each layer.

Traditional stacks often require you to update firmware, storage software, and hypervisor software as separate exercises for tooling, sequencing, and risk. With SC//HyperCore™, those updates are integrated and automated at the platform level, helping teams keep systems current without stitching together multiple update workflows.

Maintenance at Scale

For IT managers operating multiple clusters, SC//Fleet Manager is positioned to reduce maintenance time by 50% or more through simplified workflows designed into the platform.



Scale and Refresh

Scaling shouldn't force redesign. In many environments, growth happens in steps, adding capacity here, more compute there, or refreshing aging hardware without disrupting what's already running. SC//Platform™ is designed to keep expansion and refresh straightforward by making scale-out feel like an extension of the initial deployment, not a separate project with new rules.

Scale Out Without Being Locked Into Identical Nodes

When scaling out a cluster, it's designed to be as simple as the initial deployment, even when the new appliance being added has a different CPU, RAM, or storage configuration. This flexibility matters because real-world growth is rarely uniform; teams may need to add a specific resource (more memory, more storage, or more compute) rather than replicate the exact original node configuration.

By avoiding a "must match the original nodes" model, SC//Platform™ enables more practical scaling decisions as needs change, without constraining expansion to a narrow hardware profile.

Refresh in Place Instead of Rip-and-Replace

Refresh cycles are where lifecycle burden often spikes, especially if refresh is treated as a disruptive cutover event. When it's time to refresh nodes, SC//HyperCore's™ ability to scale out with different and newer appliances supports the ability to gradually phase out older appliances as the cluster grows with newer appliances. Rather than a rip-and-replace approach, a cluster can refresh in place through scaling out and retiring older nodes over time.

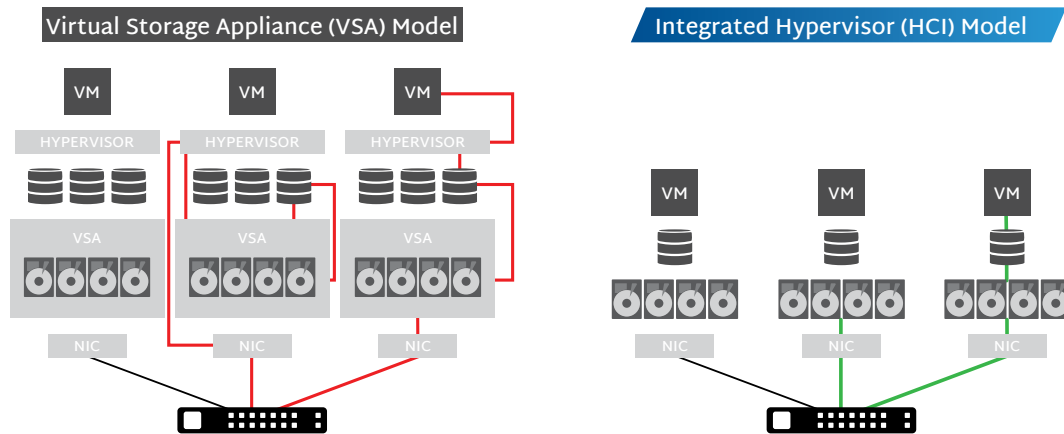
What "Flexible Scale" Enables

- Add resources as needed (compute, memory, storage) without forcing identical node configurations
- Modernize gradually by introducing newer appliances and retiring older ones over time

Storage Efficiency and Architecture

Storage is where many virtualization stacks quietly accumulate complexity. When storage is delivered through layered file systems, protocols, and virtual appliances, day-to-day operations get harder (more components to manage), and performance efficiency often suffers (more hops per I/O). SC//Platform™ reduces that lifecycle burden by embedding storage directly into SC//HyperCore™, so the storage model stays simple, efficient, and consistent as environments grow.

When it comes to data access for users, efficiency is better for productivity. Many factors affect how efficiently the data is delivered to users including networking, processing, and maybe the most important, storage architecture efficiency. Many storage architectures are complex, layering multiple storage protocols along with virtual appliances simply to present block storage to virtual machines. With SC//HyperCore™, users experience the benefits of SSD as a tier, and have the benefit of clustered redundancy that absorbs the loss of failed disks or entire nodes.



Automatic Configuration—No Storage Setup Project

With SC//HyperCore™, storage is automatically configured across the entire cluster. There’s no manual configuration required for deployment or ongoing management. All storage is combined into a single, cluster-wide pool available to all nodes. When new nodes are added, their storage is automatically and seamlessly added to the pool again, without manual configuration.

Embedded, System-wide Block Storage

The Scale Computing Reliable Independent Block Engine™ (SCRIBE) is the core component of SC//HyperCore™, combining the storage drives from each node into a single, logical, system-wide storage pool. The pooling occurs automatically with no user configuration required. Blocks are redundantly stored across the system to tolerate the loss of individual drives or an entire system node.

The SCRIBE storage pool is available to all system nodes and presented without any file systems, protocols, or virtual appliances. SCRIBE is embedded directly in the SC//HyperCore™ operating system. When a VM is created, the virtual disks provide direct block access between the virtual machines and the SCRIBE storage pool. The only file systems created in SC//HyperCore™ are those used by the guest operating systems in the VMs to address the virtual disks.

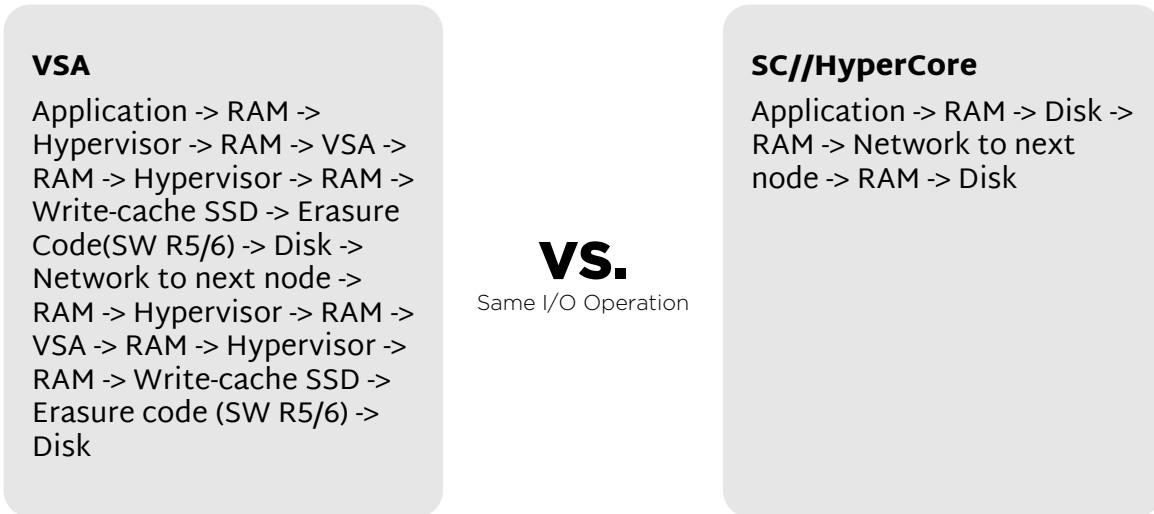
Avoiding VSA Overhead and Layered I/O Paths

Other storage architectures tend to emulate SAN or NAS devices that were traditionally used in virtualization. These start with a storage pool on the lowest level, with a file system layered on top; the result is presented to the hypervisor, where another file system is layered on and managed by a virtual storage appliance (VSA); and finally, it is presented to a VM, where another file system is layered on. Aside from the multiple levels of protocol that must be traversed for each I/O operation, the VSAs managing the storage can consume a large portion of the RAM (often 24GB+ per cluster node) that would otherwise be used to create more VMs.

“The Scale HCI platform is by far the most intuitive and stress-free product I’ve utilized as an engineer in terms of hyperconverged solutions. The firmware updates are seamless and I’ve never experienced any issues applying them. I’ve become confident enough with the product that I tend to deploy them during peak production hours with no worries that it will cause any performance issues.”



As a comparison, the route of an I/O operation in a VSA architecture may look something like:

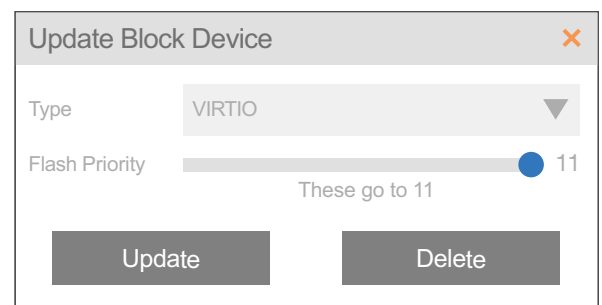


Hybrid Storage—Enhanced Automated Tiering

Data sets are typically characterized by a small percentage of data being actively accessed. In a hybrid, tiered storage system, the greatest efficiency benefit comes from having active data on SSD and inactive data on slower HDD storage. The HyperCore Enhanced Automated Tiering™ (HEAT) technology in SCRIBE monitors data access and dynamically maps active data blocks to SSD storage, while moving inactive blocks off SSD to HDD.

This movement of data blocks between tiers is transparent to users and even administrators, happening automatically. By default, each disk is actively trying to achieve an equal level of efficiency by moving data blocks between SSD and HDD tiers. Sometimes, one disk requires an even greater level of efficiency than another disk; in these cases, administrators can give those virtual disks priority.

For each virtual disk, the relative priority of SSD utilization can be adjusted on a scale from 0 to 11. The default for all disks is 4, and if no disk is ever changed, no disk has priority over any other. Any disk can be dynamically adjusted in priority, up or down, to increase or decrease the SSD utilization on that disk. Priority 0 will completely bypass the SSD and only utilize the HDD for that disk. Priority 11 will attempt to put the data for that disk completely on SSD, or as much as possible, given the available SSD on the system, taking into account all other disk priorities.



SC//HyperCore™ has built-in, real-time IOPS meters for each virtual disk, so the results of changing the priority on a particular disk can begin to be seen immediately. Because virtual disks can be dynamically changed, the priority can be adjusted as needed to achieve the optimal balance of IOPS efficiency across virtual disks and VMs.

Lower Total Cost of Ownership

Total cost of ownership isn't just what infrastructure costs to buy; it's what it costs to deploy, operate, protect, maintain, and grow over years of change. For most organizations, the biggest costs show up after day one: the time spent coordinating across tools and vendors, the effort required to keep environments consistent, and the downtime or risk introduced when routine tasks become complex projects.

SC//Platform™ is designed to reduce that long-term cost by removing lifecycle friction. When deployment is repeatable, day-to-day operations require fewer consoles, resiliency is built in, and maintenance is automated, IT teams spend less time on “infrastructure overhead” and more time on the work the business actually cares about. The result is lower operational labor cost, fewer disruption-driven costs, and less cumulative drag as environments evolve.

- Fewer tools and handoffs: Reducing the number of separate management surfaces and “glue” tools helps cut time spent correlating issues, coordinating changes, and escalating across vendors.
- Less downtime and fewer fire drills: Built-in resiliency and recovery capabilities help limit the business impact of routine infrastructure issues and disruptions.
- Lower maintenance overhead: Predictable, automated updates reduce the planning, sequencing, and weekend work that often comes with multi-layer stacks.
- More efficient growth and refresh: Flexible scale-out and refresh-in-place reduce redesign work and avoid disruptive forklift upgrades.
- Reduced “hidden tax” of layered architectures: Simpler, integrated architecture reduces operational overhead and can avoid resource waste caused by additional management or storage layers.

For a deeper cost breakdown and ROI/TCO analysis, see the [How SC//Platform™ Lowers the Total Cost of Ownership](#) white paper.

When Small Inefficiencies Multiply

In multi-cluster environments, small amounts of extra time, including per deployment, per patch cycle, per troubleshooting event, compound quickly. A platform that reduces steps and standardizes lifecycle work can materially lower total operational cost over time.

Conclusion

Most infrastructure complexity doesn't show up on a spec sheet—it shows up in the hours spent troubleshooting, coordinating updates, managing access, and recovering from disruptions. SC//Platform is designed to eliminate that hidden tax through tighter integration and smarter automation, so teams can keep applications available, keep environments consistent, and keep operations predictable as requirements change. That's what "simpler" looks like in practice.

Real-World Requirements: Simplicity Checklist

Requirement	SC//Platform™ Simplicity in Action
Availability and recovery by design	Self-healing automation and high availability clustering to stay online through routine issues, with integrated data protection for fast recovery (snapshots, replication, failover/failback, VM + file recovery) without complex runbooks
Ease of management	Fewer consoles and fewer "places to look," with a single-vendor experience for core infrastructure operations; fleet-level visibility when you have multiple clusters; and on-demand deep access without building separate remote-access stacks
Automation to reduce routine work	Repeatable deployment and day-to-day operations: streamlined setup, zero-touch provisioning at scale, APIs for integration/automation, and application lifecycle management to standardize application deployment and updates across environments
Predictable updates and lifecycle management	Integrated updates across system components so patching doesn't require stitching together independent upgrade sequences, and can be performed without turning maintenance into downtime planning
Straightforward scale and refresh	Scale out with non-identical nodes as needs change, and refresh in place over time by adding newer appliances and phasing out older nodes—without redesign or forklift upgrades
Security without operational sprawl	Our secure-by-design approach incorporates zero-trust principles and outbound-only connectivity to reduce exposure. By minimizing the need for open inbound network access and locking down unnecessary services, we reduce the attack surface by default