Realizing Modularity in Storage
In *The Architecture of Platforms: A Unified View*, the authors put forth the idea that, “A platform architecture partitions a system into stable core components and variable peripheral components.” These modules come together through distinct design rules based on “thin crossing points” of interoperability; in IT, this concept is most familiar as expressed through the idea of Application Program Interfaces (APIs). The paper makes a strong case for modularity in the design of such platforms because “The whole system does not have to be invented or rebuilt from scratch to generate a new product, accommodate heterogeneous tastes or respond to changes in the external environment. In this fashion, the platform system as a whole is evolvable”.

Putting today’s storage systems, irrespective of form factor, through this prism is an interesting exercise. Are they modular? Are they able to accommodate heterogeneous tastes? Can they easily respond to changes in external environments? The answer, unfortunately, is a resounding “No” across the board.

**Insufficient Modularity In Storage**

An argument could be made that storage systems are modular. After all, they interact with applications via well-defined protocols such as iSCSI, FC and NFS. Regrettably, these interfaces are implemented using non-intuitive constructs such as volumes and LUNs which can impose significant overhead to interoperate across divisions like Operating System environments (e.g. Linux vs. Windows) or storage types (e.g. block vs. file). These lack of “crossing points” lead to siloed deployments that runs counter to the idea of a modular platform that removes the need to rebuild systems from scratch to accommodate changing or varied requirements.

Further, if storage products were sufficiently modular, it should be possible to replace the mechanical drives in older storage systems with flash-based drives seamlessly and gain the full performance potential of the newer media. Similarly, if new storage products such as All Flash Arrays (AFAs) were built with modularity in mind, then its underlying software should be portable to different hardware—an industry-standard server with an equivalent configuration, for example—and provide the same benefits. Finally, if storage products were indeed modular, scaling out the system capabilities would be as easy as combining multiple instances of a monolithic array much like a jigsaw puzzle. None of these scenarios are possible today.

The insufficient modularity in storage has impaired its ability to advance as a platform. Two fundamental changes in the data center environment—virtualization and new media—are prime examples of externalities storage hasn’t been able to evolve with adequately.

**Virtualization**

The widespread adoption of virtualization has put unique requirements and constraints on storage (e.g. the ‘I/O Blender’ effect). However, the lack of modularity in storage has resulted in two critical problems for virtualized environments:

1. Storage has become a prime candidate to be enormously overprovisioned because capacity had to be purchased in order to gain enough performance to reach the desired service levels for virtualized applications.

2. Even though the rest of the infrastructure revolves around the requirements of the virtual machine (VM), storage systems aren’t provisioned and managed in a VM-aware fashion. With operations based on the aforementioned LUNs and volumes which map poorly to VMs, designing and tuning storage for virtual workloads becomes a needlessly complex task.
Emergence of New Storage Media

There have been tremendous innovations in storage media over the past few years. Flash, large DRAM systems, and non-volatile memory are some of the examples of the exciting new developments in this space. Unfortunately, the lack of modularity has meant that storage products have not been able to adopt these innovations very well.

For example, if flash was procured inside an array, that particular model of flash must be in use for many years until the next storage upgrade. This means that the rapidly improving price/performance characteristics of this technology isn’t being fully harnessed (Figure 1 below from technology research and advisory organization Wikibon illustrates this dynamic). In addition, flexibility is also limited as the flash options made available by the storage vendor is, without fail, a vastly reduced set relative to the open market.

![10-year Technology Cost/Terabyte Projections 2014-2023](image)

**Figure 1 Price/performance projections for Tape, Disk and NAND Flash**

The emergence of ‘hyperconverged’ appliances—products that converge the functions historically provided by servers and storage into one box—has been another industry reaction. The motivating factor of this approach is the perception of simplicity—why buy storage and servers independently when you can get both from one box? This is, of course, the exact opposite of improved modularity and extends the limitation described above where performance gains were tied to the acquisition of unneeded capacity; now entire servers must be procured in order to increase storage functionality. While this model may work at small scale, it definitely breaks in large and heterogeneous environments where there are enormous variations in requirements and any inefficiencies are magnified.
Redefining Modularity For Storage

In order to truly apply the concept of modularity to the storage environment, the elemental functions asked of it must be deconstructed and examined. It is also vital to shift the design focus from a storage-centric model revolving around ideas like LUNs/volumes to a framework built to address what is intuitive and actually matters to the end user.

At its core, storage systems provide:

1. Raw capacity for retaining the data that the application needs and generates;
2. Performance guarantees that make sure the application/end user expectations are met

Fundamentally, as a first step, these two requirements should be decoupled from each other and addressed separately as independent modules in their storage deployment. Proper mechanisms would then be put in place to ensure that these components interacted with each other via ‘thin crossing points’ that did not require a lot of operational overhead and did not require altering one component to enable another.

Next, the performance component should reside as close to the application as possible in order to provide maximum value. The Capacity module would reside wherever the user deemed appropriate and the product itself would not force any design choices on the user. Figure 2 below depicts roughly what this would look like.

Finally, an understanding of the I/O patterns for each VM, not LUN or volume, is needed to ensure that policy management operations occurs in alignment with the virtualized infrastructure and application. And, the best way to support virtualization in this manner is to let software define the storage.

True modularity in storage is achieved once such a decoupled architecture is implemented. This means users can better manage existing deployments and take on new strategic projects while being able to leverage external innovations as they happen. Below are some of the benefits a decoupled storage deployment offers:

**Capacity**

Capacity growth can occur in a modular way without the need to balance the media choice with the performance requirements. For instance, capacity can be optimally delivered with lower RPM drives providing better cost-per-GB characteristics rather than compromising with higher speed spinning disk or value-oriented SSDs. Adding capacity also wouldn’t require the procurement on ancillary components such as a server, nor introduce operational disruption or downtime for applications.

**Performance**

The performance need can be addressed independently by adding high-speed resources at the server tier without the burden of having to take on unnecessary capacity. The performance investments could grow precisely, as needed over time instead of relying on the step-function cycle of storage upgrades (Figure 3).
Since performance is satisfied by server-side resources right next to the application, the rate of improvement is more rapid and delivered closer to the moment of need. Hardware innovation can also be leveraged without a cost overhead.

**Conclusion**

Current storage products are insufficiently modular and impose unnecessary and outdated constraints on the end user. The lack of modularity has created a gap between storage and the user requirements defined by two major trends currently predominant in the industry—virtualization and the emergence of new storage media.

By independently addressing storage capacity and performance, decoupled storage better meets the needs of the modern data center where heterogeneous tastes are the norm rather than the exception. By embracing the concept of modularity to the fullest, it is capable of dynamically evolving to changes to the external environment and brings a fully realized platform architecture to storage.