

# Archiving vs Transparent Archiving

What's the difference?

With the continued massive growth in data, archiving has taken the spotlight as a logical way to keep up with growth and keep storage costs down. With so many solutions on the market, it's important to understand the differences in archiving techniques to make the best decision. These differences can have a tremendous and often unexpected effect on your budget, IT resources, and end users.

This white paper explores the different types of archiving solutions, how they work, and what you can expect in terms of disruption. Everything from data access, vendor lock-in, hidden costs, and end-user experience will affect the amount of data IT is allowed to archive. And the amount of data archived directly translates to the amount of money your organization can save.

Data archiving can save 70% of costs without disruption, if done correctly.

## The Value of Archiving

Archiving cold data from primary storage to less-expensive storage can add up to significant savings. Cold data is usually defined as files not accessed in about a year. The cheaper storage you're archiving it to is usually an object store or cloud. Based on what we've seen among customers, roughly 75% of their data is cold. Archiving this cold data frees up that much of NAS capacity and substantially lowers storage costs.

### Savings beyond primary storage

Archiving cost savings doesn't stop at offloading your NAS. When archiving is done correctly, the entire file should be transparently moved off the NAS so that even backup and DR applications don't make extra copies of the full file. Now you've reduced your backup window and license costs because you're no longer maintaining three or more backup copies. You also no longer need a mirrored copy of that cold data for DR. It adds up to significant savings.

### Total Archiving Savings

- Frees up NAS
- Reduces backup storage and license costs
- Reduces DR storage and license costs

## The 5 keys of good archiving are:

- **Maximizes cost savings:** Saves not only primary storage footprint and costs but also backup and DR footprint and costs to provide maximum savings.
- **Non disruptive to users and apps:** User and apps continue to operate as before without disruption even after cold data is archived.
- **Outside hot-data path:** Does not degrade performance by getting in the way of hot data and meta data paths.
- **No vendor lock-in:** Archiving does not lock you into the same vendor for your primary NAS and you can freely choose any secondary storage to archive data.
- **Native access from secondary storage:** Archived data should be directly accessible from the secondary storage to enable on-going and future use of the data.

## Archiving: Disruption is the differentiator

Many vendors offer data archiving, but there's a vast difference between solutions. Some are saying that their archiving is transparent to users, apps, and workflows. But when you take a closer look at how the solution archives cold data, you discover a big difference—and lots of disruption—from data access, hidden costs, and vendor lock in.

### Disruption needn't be inevitable

There's no reason data archiving should present any disruption at all. Cold data that's been moved to a cheaper capacity storage should look the same to users and apps—as if it's still on your fast, expensive primary storage.

The ability for both users and applications to still access files exactly where they were before without having to rehydrate the file is possible with a true transparent archiving solution, such as delivered with Komprise Intelligent Data Management. That's why it's important to ask the right questions to make an educated choice, avoid surprises, and save your organization costs.

To highlight the differences in archiving, we've categorized solutions into three categories:

1. Traditional archiving
2. Proprietary transparent archiving
3. Standards-based transparent archiving

### 1. Traditional Archiving: Lift & shift is disruptive and minimizes savings

In this scenario, end users can literally wake up and find their data gone. Because, at its most basic, archiving simply moves cold data from the primary storage onto another medium. This is a lot easier than transparent archiving, which is why so many vendors offer it, but this simplicity comes at a cost.

*IT file retrieval – Disruptive to users.* If users need to access a cold file or run an older application that requires accessing a cold file that's been traditionally archived, they must file a support ticket. IT administrators like these “go fetch” activities about as much as users like waiting for them to be handled.

*Inefficient approval process – eliminates most of the savings.*

Traditional archiving requires a manual approval process between users and IT. First to gain permission, then to painstakingly go through which files can be archived, and then to repeat this process on an ongoing basis to identify cold data to offload primary storage. Not only is this highly inefficient, but it results in only archiving less than 10% of the 70% cold data they have—a tremendous savings loss.

#### More worry, less archiving

When users are worried about being able to access archived data, they're less likely to allow archiving, which reduces savings.

*Project-based archiving requires user behavior change – minimizes savings.* Traditional archiving is limited to projects whose data is neatly organized into a collection of data, such as a share/volume or a directory. IT relies on users letting them know when projects are completed and data becomes cold. Users will need ample warning for project-based, or batched archiving to avoid surprises finding their data. Obviously many workplace projects aren't neatly defined, and when they are, they often run for years.

The combined manual approval process and project-based approach of traditional archiving is not only highly inefficient but results in **<10% of all cold data being archived, which minimizes cost savings.**

## 2. Proprietary Transparent Archiving: HSM and storage tiering

*Hierarchical Storage Management (HSM).* HSM is a long-established concept, dating back to the 70's. As one of the first attempts at transparent archiving, it had limited success due to its use of problematic proprietary interfaces, such as stubs. These stubs make the moved data appear to reside on primary storage, but the transparency ends there. To access data, the HSM intercepts access requests, retrieves the data from where it resides, and then rehydrates it back to primary storage. This process adds latency and increases the risk of data loss and corruption.

The brittleness of stubs is also problematic. When stubbed data is moved from its storage (file, object, cloud, or tape) to another location, the stubs can break. The HSM no longer knows where the data has been moved to and it becomes orphaned, preventing access. Existing HSM solutions on the market use client-server architecture and do not scale to support data at massive scale.

Since HSM solutions have so many problems, they have not had mass market adoption and have been supplanted by storage tiering, which is what we will focus on for the rest of the comparison.

*Storage Array Tiering.* Increasing data growth and the high cost of Flash and NAS prompted storage vendors to tier data. Tiering has become an integral part of the storage array and is used to migrate cold blocks from hot, expensive tiers to lower cost cheaper tiers. However, soon primary vendors began marketing these as archive solutions. In this case, they tier blocks off the storage array and onto another storage target such as an on-premises object store or the cloud.

The problem is that these tiering solutions move cold blocks of the file from the primary storage to a capacity storage device. When an application, such as a 3rd-party backup system accesses these files, the storage array must bring back the tiered blocks and provide it to the application. This has multiple adverse implications.

First, it does not reduce the backup footprint. So all savings from footprint reduction and backup licenses is lost. Secondly, it can potentially *increase* the backup window since fetching all of these blocks back from the capacity storage unit will be slower. Many storage vendors claim they can tier to the cloud, but this not only results in more degraded performance, it also incurs avoidable egress costs.

### HSM

Proprietary interfaces add complexity

- Proprietary stubs are brittle, hard to keep up to date and complex to manage
- Agents interfere with hot data access

### Storage Tiering

Proprietary block-based solution that limits savings:

- **Eroded Savings:** because it's not transparent to 3rd-party backups and does not reduce backup footprint, thus eliminating significant savings
- **Lock-in:** Limited ability to switch storage vendors or clouds.
- **No native access on target:** Only cold blocks, not the full file are moved so the data cannot be directly accessed on secondary storage.

Storage tiering or “pools” solutions from storage vendors are good for tiering secondary copies of data such as snapshots from Flash. But a large amount of files are cold, and these need to be transparently archived at the file level to capture both the storage and backup savings.

### 3. Standards-based Transparent Archiving: Komprise Intelligent Data Management

Komprise offers the only true transparent archiving solution because it creates literally no disruption. Its modern standards-based patented approach, Transparent Move Technology™ (TMT), uses symbolic links instead of proprietary stubs.

*Transparency with TMT.* When a file is archived using TMT, it’s replaced by a symbolic link, which is a standard file system construct available in both NFS and SMB/CIFS file systems. It retains the same attributes as the original file thereby saving space on the primary storage. The symbolic link points to the Komprise Cloud File System (KCFS), and when a user clicks on it, the file system on the primary storage forwards the request to KCFS, which fetches the file from the secondary storage where the file actually resides.

*Continuous monitoring saves more.* Komprise continuously monitors all the links to ensure they’re intact and pointing to the right file on the secondary storage. It extends the life of the primary storage while reducing the size of the primary storage required for your hot data. This makes it affordable to replace your existing primary storage with a smaller, faster, flash-based primary storage.
















*Reduces backup.* Most backup systems can be configured to just *backup* symbolic links and not follow them. This reduces the backup footprint and costs because only the links are backed up, not the original file. If an archived file needs to be restored, the standard restore process for that backup system needs to be followed to restore the symbolic link that is then transparently used to access the original file.

*Reduces rehydration.* To control rehydration, Komprise allows you to set policies when an archived file can be rehydrated back onto the primary server. You can limit rehydration upon first file access, or when a file is accessed a set number of times within a set time period. When a file is accessed for the first time, KCFS caches it thereby ensuring fast access on all subsequent access.

*Recalls files in bulk.* Komprise lets you recall files in bulk, if required. For instance, say you have setup a policy to archive all data not accessed in a year, but you had a chip design project that was completed three years ago. Most likely all the project files have been archived, but if you’re about to develop the next version of the chip, it’s more efficient to simply recall all those files in bulk.

#### Standards-based Transparent Archiving

- Keeps original file attributes Available in NFS, SMB/CIFS
- No proprietary agents or stubs

Key Archiving Feature	Traditional Archiving	Proprietary Transparent Archiving via Storage Tiering	Standards-based Transparent Archiving
<b>Maximizes storage &amp; backup savings</b>	<b>No</b> , manual process results in archiving <10% of total cold data 	<b>No</b> , only 20% of savings since backup and DR footprint and license costs aren't reduced. 	<b>Yes</b> , provides 100% savings since primary storage footprint is reduced as is backup and DR. 
<b>Non disruptive to users and apps</b>	<b>No</b> , users must search for data in two places 	<b>No</b> , it's transparent to users but not applications, such as backup and virus scanners. 	<b>Yes</b> , users and apps are not disrupted. Backup footprint is reduced; virus scanners are not impacted. 
<b>Vendor agnostic</b>	<b>Yes</b> , data typically moved in native format 	<b>No</b> , migrating to primary storage of another vendor is not simple. All archived data must be rehydrated and then migrated to the new primary storage. 	<b>Yes</b> , you can readily migrate from one primary vendor to another without re-hydrating all the archived data. 
<b>Outside hot data path</b>	<b>Yes</b> , the file is fully moved 	<b>No</b> , tiering cold blocks to slow secondary storage (e.g., cloud) can adversely impact performance of these arrays. 	<b>Yes</b> , sits behind hot data and meta data paths 
<b>Native access on secondary storage</b>	<b>Yes</b> , so long as the file was stored in a format native to the secondary storage 	<b>No</b> , the blocks and files moved are in proprietary format and can only be accessed from the primary storage device. 	<b>Yes</b> , the full file is written in a format native to the secondary storage device. 
<b>Example solutions</b>	Project-based archiving, adhoc archiving or backup-based archiving	Storage array tiering, HSM	<b>Komprise Intelligent Data Management</b> Provides both project-based and policy-based archiving

## Summary

Archiving takes many forms, many of which can cause different types of disruption to organizations. Some solutions claim the ability to transparently archive but are unable to reduce backup footprint and make it nearly impossible to switch primary vendors thus eroding savings and imposing vendor lock-in.

Archiving can be categorized into three categories. **Traditional archiving** creates significant disruption to user access, requires manual approvals and archives only in batches, eliminating a substantial amount of cold data to be archived and cost savings to be realized. **Proprietary transparent archiving** is the method behind HSM, which is cumbersome and includes brittle and unreliable stubs and agents. Storage vendors provide archiving with tiering software, which eliminates stubs but can impact performance and imposes vendor lock-in. It also drastically reduces the cost savings afforded by transparent archiving because it fails to reduce the backup and DR footprint. **Standards-based transparent archiving** is the only true transparent archiving method that eliminates all disruption and delivers maximum archiving savings.

Only Komprise Intelligent Data Management uses a patented Transparent Move Technology™ that offers standardsbased transparent archiving and provides maximum savings. It archives data without disruption to users or getting in front of hot data—all without using stubs or agents or creating vendor lock-in.

Komprise provides a systemic, automated approach to archiving. It can be run across your file servers in your data center to continuously remove cold data from your primary storage, saving primary storage and storage and licensing costs for backup and mirroring for ongoing maximum savings.

With the pressure to save costs amidst soaring unstructured data growth, many companies are turning to Komprise Intelligent Data Management to achieve maximum savings without any disruption to their organization.

## Learn More

Go to [Komprise.com/product](https://komprise.com/product) to learn more.



Komprise, Inc.  
1901 S. Bascom Ave. Suite 500  
Campbell, CA 95008  
United States

For more information:  
Call: 1-888-995-0290  
Email: [info@komprise.com](mailto:info@komprise.com)  
Visit: [komprise.com](https://komprise.com)

For media requests email  
[marketing@komprise.com](mailto:marketing@komprise.com)  
©2020 Komprise, Inc. All rights reserved.