WHITE PAPER

Introduction to Sohonet Media Storage

November 2013

Executive Summary

The increased demand on the media infrastructure is higher than the provided capital budget that is required to support the increase and upgrade of current infrastructure – in particular storage infrastructure.

Hence, Sohonet Storage has been developed and used for both Film and TV pipelines to provide extensible short-term and long-term parking, back-up, digital negative storage and an alternative to LTO or portable disks. By having the Storage attached and embedded within the Sohonet Media Network, productions and post-production companies can use, share and move content via our production-proven a secure and highly redundant storage platform.

This white paper introduces Sohonet Media Storage (SMS) and provides an insight into how the SMS can be interfaced by users, how it works and why it is relevant in the media and entertainment industry.

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Version 2.0 November 2013

CONTENTS

CONTENTS 2
Introduction to Sohonet Media Storage 4
Sohonet Media Storage Characteristics 5
High-level Features and Benefits 5
Sohonet Media Storage Compatible Tools 6
REST Application Programmers Interface (API): 7
Command Line Interface (CLI) 7
Summary of Sohonet Media Storage Interface Comparison 8
Using Sohonet Media Storage 10
How does Sohonet Media Storage Work? 10
Building Blocks 10
Proxy Servers 10
The Ring 11
Zones: Failure Boundaries 12
Accounts & Containers 12
Partitions 13
Replication 14
Component inter-relationship? 15
Sohonet Media Storage Cluster Architecture 16
Considerations 17
Why is Sohonet Media Storage relevant? 18
Sohonet Media Storage is Scalable 18
Sohonet Media Storage is Extremely Durable 18
Sohonet Media Storage is based on Open Standards and Open Source Software 18

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Sohonet Media Storage uses Similar API to AWS S3 18

Sohonet Media Storage uses Industry-standard Components 19

Sohonet Media Storage is privately deployed Infrastructure As-a-Service 19

Summary 20

APPENDIX A – API and CLI 21

Sohonet Media Storage API Commands - The Basics 21

Sohonet Media Storage CLI Commands - The Basics 22

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Introduction to Sohonet Media Storage

Sohonet Media Storage expands and extends production or post-production storage capability without having to purchase, manage and maintain infrastructure. Sohonet Media Storage is developed and implemented using Open Stack technology, with embedded redundancy and is securely accessible via the private Sohonet Media Network.

Sohonet Media Storage is part of the Sohonet Hub platform enabling any existing or future Sohonet customers to use 1 Terabyte to 10s of Petabytes and more, as if the storage was on-site. Within minutes, customers can configure their own storage requirements and allocate permissions to users. Integrated, seamless and private transfers between site and our private cloud are able to run at full line speed.

Sohonet Media Storage expands and extends production or post-production storage capability without having to purchase, manage and maintain assets. Sohonet Media Storage uses Openstack Swift technology and is securely accessible via the private Sohonet Media Network using AES encryption and HTTPS. Sohonet Media Storage can be accessed using a web application within the Sohonet Hub platform, enabling any existing or future Sohonet customers to use a single Terabyte, or scale-out to multiple Petabytes.

There are four other ways to move and access your content via the Sohonet Media Network dependent on requirements and technical expertise as shown in figure 2 and below:-

1) Cyberduck or Cloudberry Explorer/Backup/Drive or Expandrive or FTP/Secure FTP
2) RESTful[1] Application Programmers Interface (API) and Command Line Interface (CLI)
3) Technology Component Partners providing CIFS/NFS[2] gateway via Maldivica or Panzura or QStar (Monthly rental - Price on Application)
4) Technology Integrated Partners providing native support for SMS, Marquis Project Parking, GBLabs Space/Echo Core V3 backup, Nativ MIO Cloud and PixIT Media Warp Backup.

Typical uses of Sohonet Media Storage are backup, parking, short-term archive, alternative to LTO, backoffice backup, transition storage between ingest and post-production, shadow storage and replacement of other portable media. This provides customers the ability to utilise Sohonet Media Storage as remote storage that is designed for write many, read many and modify none.

Sohonet is synonymous with optimized private network interconnectivity for the media and entertainment industry with over 15 years of experience in Film, Commercials and TV production and post-production.

[1] Representational State Transfer web API provides a set of operations supported by the web API using HTTP methods (e.g., GET, PUT, POST, or DELETE).
[2] CIFS/NFS presentation does not allow file modification after writing to the Sohonet Media Storage and functionality is limited to Write (Delete) Many, Ready Many, Modify None

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Sohonet Media Storage is a multi-tenant, highly scalable and durable object storage system that has been designed to store large amounts of media data and content at low cost via a RESTful (Representational State Transfer) HTTP API.

Sohonet Media Storage is designed to be horizontally scalable as there is no single point-of-failure. Sohonet Media Storage is also ideal for storing and serving content to many concurrent users especially during production or post-production.

What differentiates Sohonet Media Storage from many other storage systems is that it originated from the large-scale production environment. Thus, Sohonet Media Storage has been designed to withstand hardware failures, without any downtime, and provide operation teams the ability to maintain, upgrade and enhance a cluster while in use or during production. Sohonet Media Storage scales linearly enabling operations to transparently add storage capacity when it is needed without worrying about performance overhead costs.

Sohonet Media Storage Characteristics

The key characteristics of Sohonet Media Storage include:

- Sohonet Media Storage supports any application that will support OpenStack™ Swift
- Sohonet Media Storage is fully integrated within the Sohonet Media Network
- All objects stored are replicated three times (x3) across the storage cluster, which can be defined as a group of drives, a node, a rack etc.
- All objects have their own metadata that is stored separately from the actual data
- Developers interact with the object storage system through a RESTful HTTP API
- The cluster scales by adding additional nodes – without sacrificing performance, which allows a more cost-effective linear storage expansion
- New nodes can be added to the cluster without downtime and failed nodes and disks can be swapped out with no downtime

High-level Features and Benefits

Directly connected to Sohonet Media Network	All members of the Sohonet Media Network can access the Sohonet Media Storage in accordance to access control lists maintained within the Sohonet Hub.
High Speed Dedicated Access	Tight integration with the Sohonet Media Network (SMN) allows the provision of access speeds far in excess of other 'cloud' based solutions, with 10Gb/s (or more) available depending on location and budget requirements.
Media optimized private storage	Sohonet Media Storage has been developed and tested for both Film and TV pipelines including the optimized support of both image sequences e.g. DPX, OpenEXR and wrapped or contained media e.g. MXF, DNxHD, ProRes.

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Sohonet Media Storage Compatible Tools

There are several tools compatible with Sohonet Media Storage, including storage gateways, file managers, backup tools and file system adapters. Here is a list of some of the tools, which are compatible with Sohonet Media Storage:

Sohonet Media Storage Hub

- Sohonet Hub
  - Web portal and administration tools
  - Uploading of files, multiple files and folders
  - Downloading of files, multiple files and folders
  - Container, folder and folder within folder hierarchy
  - Ability to set user access via account administration
  - Two levels of user access per account, administrator and User

Storage Hardware Gateways (exposes NFS and/or CIFS):

- Limited Protocol Translator (NFS/CIFS) for OpenStack Sohonet Media Storage (Price on Application)
- Simultaneously access your data using CIFS, NFS and HTTP
- Multi-device synchronization for data sharing / collaboration
- File-system permissions retained with data in object storage

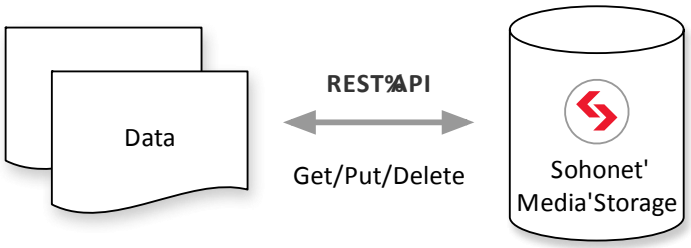
File Managers:

- CloudBerry Explorer
  - Software gateway
- Cyberduck
  - Software gateway
- Expandrive
  - Software gateway and drive mount

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

REST Application Programmers Interface (API):

- The Sohonet Media Storage has a REST-ful API and
- Uses HTTP verbs (PUT, GET and POST) to signal the requested action.



Command Line Interface (CLI)

Developers can either write directly to the Sohonet Media Storage API, CLI or use one of the many client libraries that exist for all popular programming languages, such as Java, Python, Ruby and C#. Amazon S3 and RackSpace Cloud Files users should feel very familiar with Sohonet Media Storage.

You can run the CLI from a desktop machine or remote system. A common Openstack CLI is in development also.

To install a client on a Mac OS X or Linux system, you can use easy_install or pip or install the package from your Linux distribution. Using pip is recommended because it is easy and it ensures that you get the latest version of the nova client from the Python Package Index. Also, it lets you update the package later on.

Here are the CLIs for use with the Sohonet Media Network:

- Keystone - Controls and creates users, tenants, roles, endpoints, and credentials.
- Swift - Provides access to a swift installation for adhoc processing, to gather statistics, list items, update metadata, upload, download and delete files stored by the object storage service.

To use the CLI the following packages need to be installed:-

Keystone, http://docs.openstack.org/cli/quick-start/content/install_openstack_keystone_cli.html

Open Stack Swift, http://docs.openstack.org/cli/quick-start/content/install_openstack_swift_cli.html

For more information on how to use the API and CLI please reference **APPENDIX A – API and CLI.**

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Summary of Sohonet Media Storage Interface Comparison

Interface Functionality	Sohonet Hub	3rd Party Software	Application Programmers Interface (API)	Command Line Interface (CLI)	3rd Party Hardware (protocol translation)
Installation	No Installation	Simple	Expert	Intermediate	Intermediate
Administration	Simple	Simple	Expert	Intermediate	Intermediate
Scripting	None	Limited	Unlimited	Unlimited	Some
Container/Folder Management	Some	Limited	Unlimited	Some	Limited
Upload per session	1-100	101-10,000	10,000+	10,000+	1000+
Download per session	1-100	101-10,000	10,000+	10,000+	1000+
Max File Size	<10GB	<4-5GB & <1TB [3]	>5GB [4]	>5GB [3]	> 5GB < total cache [5]
Onsite Caching	No	No	No	No	Yes
Throughput (1Gbps conn.)[6]	68-108MB/s[7]	<117MB/s[8]	<117MB/s	<117MB/s	<117MB/s
Application	Sohonet Hub	Cyberduck, ExpanDrive, Cloudberry etc	Openstack Swift REST API	Openstack Swift Client CLI	NFS /CIFS protocol gateway, Panzura, Maldivica
Use Case	Account/container administration and basic transfers	Simple Client application for uploading, downloading & folder synchronisation	Full pipeline integration used by development teams	Full pipeline integration used by system administrators	Protocol translation for NFS/CIFS presentation

[3] ExpanDrive <4GB, Cyberduck <5GB and Cloudberry Apps (Drive, Explorer & Backup) <1TB
[4] Files larger than 5GB can be supported when Openstack Swift segmentation is used
[5] Files larger than 5GB can be supported as long as the file doesn't exceed local cache (Maldivica), Panzura and GStar provide paging and hence support files larger than cache.
[6] Test conditions used a dedicated 1Gbp/s Sohonet Media Network connection. API and CLI can exceed 117MB/s on a 10Gbp/s but this requires Sohonet engineering assistance
[7] Depends on Internet Browser, Firefox & Chrome 68MB/s and Safari 108MB/s on Mac OS
[8] Multiple application sessions are required to saturate 1Gbp/s connection

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

3rd Party Software

Native support for Openstack Swift is provided by a variety of applications for:-

- File browsing/management, similar to a FTP client
- Virtual drives (soft mount) with operating system
- Command Line Interface (CLI) – for low level access to the Sohonet Media Storage
- Appliance or CIFS/NFS gateway – for presenting the Sohonet Media Storage as a CIFS/NFS share. NOTE: This is a not a NAS and enables Write (Delete) Many, Read Many, Modify None. The exception is the Panzura appliance as this provide modification capability.

Application	Application Type	Windows	Mac OS	Linux	Link
Sohonet Hub	File Browser	Yes	Yes	Yes	storage.sohonet.com
Cyberduck	File Browser	Yes	Yes	No	www.cyberduck.ch
ExpanDrive	Virtual Drive	Yes	Yes	Yes[9]	www.expandrive.com
Cloudberry Explorer	File Browser	Yes	No	No	www.cloudberrylab.com/free-openstack-storage-explorer.aspx
Cloudberry Enterprise Backup	Backup Software	Yes	No	No	www.cloudberrylab.com/amazon-s3-enterprise-backup.aspx
Cloudberry Drive	Virtual Drive	Yes	No	No	www.cloudberrylab.com/virtual-drive-amazon-s3-azure.aspx
Openstack Swift CLI	Command Line Interface	Yes[10]	Yes	Yes	www.launchpad.net/python-swiftclient
Maldivica[11]	Appliance/CIFS /NFS Gateway	Yes	Yes	Yes	www.maldivica.com
Panzura[10]	Appliance/CIFS /NFS Gateway	Yes	Yes	Yes	www.panzura.com
Qstar[10]	Appliance/CIFS /NFS Gateway	Yes	Yes	Yes	www.qstar.com
Ctera[10]	Appliance/CIFS /NFS Gateway	Yes	Yes	No	www.ctera.com

We also work with a variety of Technology Partners that provide native support for Sohonet Media Storage provide direct access i.e. without the need for software or appliance, access to the Sohonet Media Storage. Current Technology Partners are Nativ, Marquis, GBLabs, PixIT Media and QStar. Please contact Sohonet about the latest Technology Partners.

[9] Linux version will be support by December 2013
[10] Openstack Swift Client can be used in Windows, but requires Python 2.7 or above
[11] Support for operating systems is via CIFS/NFS presentation

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Using Sohonet Media Storage

All communication with Sohonet Media Storage is done over a REST-ful HTTP API either directly or via the Sohonet Hub or 3rd party software or hardware. Application Developers who'd like to take advantage of Sohonet Media Storage for storing content, documents, files, images etc. can use one of the many client libraries that exist for all popular programming languages, including Java, Python, Ruby, C# and PHP.

How does Sohonet Media Storage Work?

Building Blocks

The components that enable Sohonet Media Storage to deliver high availability, high durability and high concurrency are:

- **Proxy Servers:** Handles all incoming API requests.
- **Rings:** Maps logical names of data to locations on particular disks.
- **Zones:** Each Zone isolates data from other Zones. A failure in one Zone doesn't impact the rest of the cluster because data is replicated across the Zones.
- **Accounts & Containers:** Each Account and Container are individual databases that are distributed across the cluster. An Account database contains the list of Containers in that Account. A Container database contains the list of Objects in that Container
- **Objects:** The data itself.
- **Partitions:** A Partition stores Objects, Account databases and Container databases. It's an intermediate 'bucket' that helps manage locations where data lives in the cluster.

Proxy Servers

The Proxy Servers are the public face of Sohonet Media Storage and handle all incoming API requests. Once a Proxy Server receives a request, it will determine the storage node based on the URL of the object, e.g. https://storage.sohonet.com/v1/account/container/object. The Proxy Servers also coordinates responses, handles failures and coordinates timestamps. Proxy servers use a shared-nothing architecture (each node is independent and self-sufficient and can be scaled, as needed based on projected workloads). A minimum of three Proxy Servers are deployed for redundancy and should one proxy server fail, the others will take over.

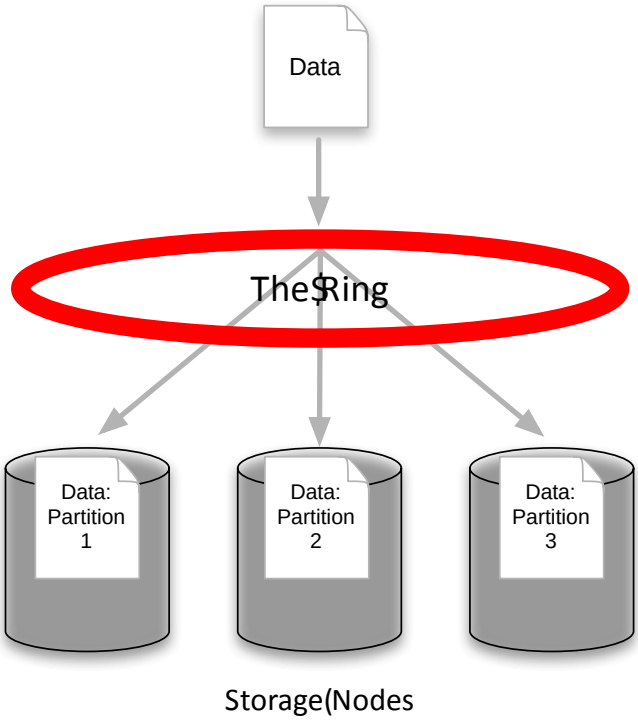
The Ring

The Ring maps Partitions to physical locations on disk. When other components need to perform any operation on an object, container, or account, they need to interact with the Ring to determine its location in the cluster.

The Ring maintains this mapping using zones, devices, partitions, and replicas. Each partition in the Ring is replicated three times by default across the cluster, and the locations for a partition are stored in the mapping maintained by the Ring. The Ring is also responsible for determining which devices are used for handoff should a failure occur.

Fundamentally, the rings determine where data should reside in the cluster. There is a separate ring for account databases, container databases, and individual objects but each ring works in the same way. These rings are externally managed, in that the server processes themselves do not modify the rings, they are instead given new rings modified by other tools.

The ring uses a configurable number of bits from a path's MD5 hash as a partition index that designates a device. The number of bits kept from the hash is known as the partition power, and 2 to the partition power indicate the partition count. Partitioning the full MD5 hash ring allows other parts of the cluster to work in batches of items at once, which ends up either more efficient or at least less complex than working with each item separately or the entire cluster all at once.



Sohonet Media Storage stores 3 replicas of your data. For a given partition number, each replica's device will not be in the same zone as any other replica's device. Zones are used to group devices based on physical locations, power separations, network separations, to lessen multiple replicas being unavailable at the same time.

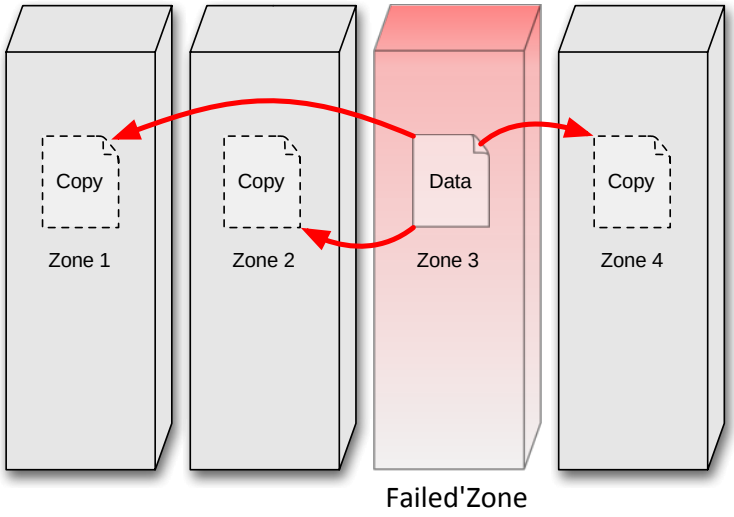
The Ring maps partitions to physical locations on disk.

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Zones: Failure Boundaries

Sohonet Media Storage is configured to isolate failure boundaries. Each replica of the data resides in a separate zone. Each rack (or multiple racks) of object servers, represent a zone. The goal of zones is to allow the cluster to tolerate significant outages of storage servers without losing all replicas of the data.

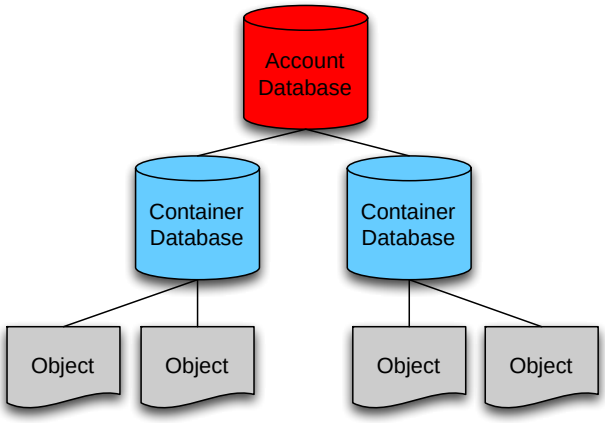
As previously mentioned, everything in Sohonet Media Storage is stored, by default, three times. Sohonet Media Storage will place each replica "as-uniquely-as-possible" three times (x3) across the storage cluster to ensure both high availability and high durability. This means that when choosing a replica location, Sohonet Media Storage will choose a server in an unused zone before a server in a zone that already has a replica of the data.



Accounts & Containers

Each account and container is an individual SQLite database that is distributed across the cluster. An account database contains the list of containers in that account. A container database contains the list of objects in that container.

To keep track of object data location, each account in the system has a database that references all its containers, and each container database references each object.



T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

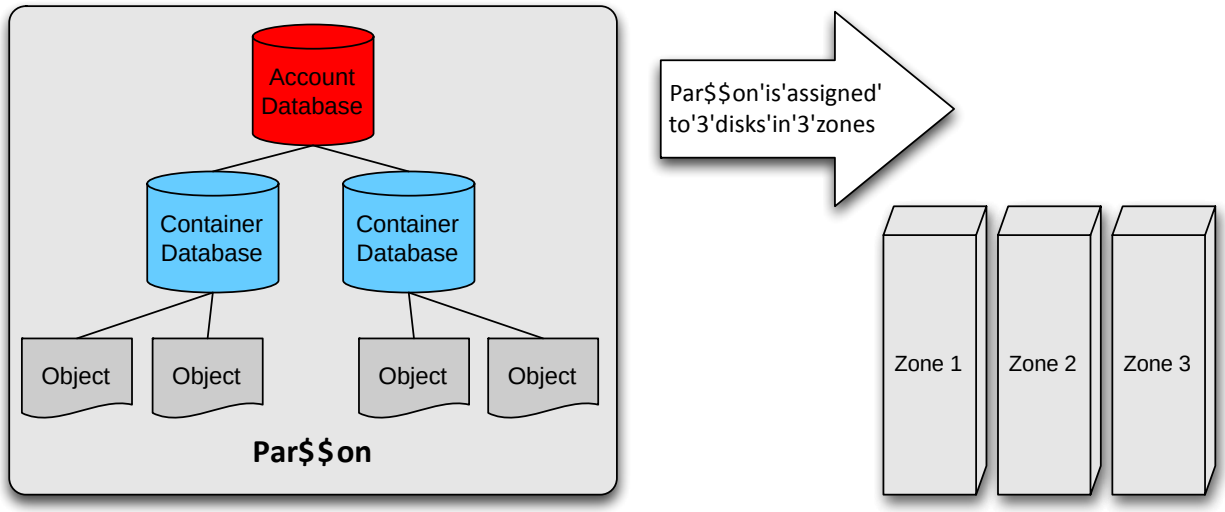
Partitions

A Partition is a collection of stored data, including Account databases, Container databases, and objects. Partitions are core to the replication system.

Think of a Partition as a bin moving throughout a fulfillment center warehouse. Individual orders get thrown into the bin. The system treats that bin as a cohesive entity as it moves throughout the system. A bin full of things is easier to deal with than lots of little things. It makes for fewer moving parts throughout the system.

The system replicators and object uploads/downloads operate on Partitions. As the system scales up, behavior continues to be predictable as the number of Partitions is a fixed number.

The implementation of a Partition is conceptually simple – a partition is just a directory sitting on a disk with a corresponding hash table of what it contains.



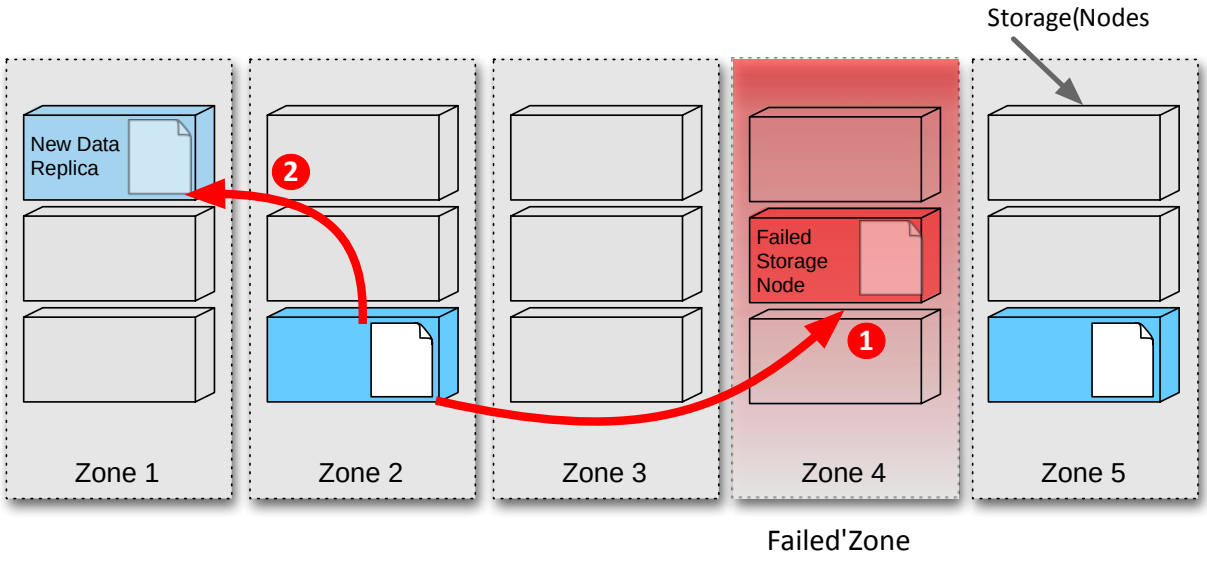
*Sohonet Media Storage partitions contain all data in the system.

Replication

In order to ensure that there are three copies of the data everywhere, replicators continuously examine each Partition. For each local Partition, the replicator compares it against the replicated copies in the other Zones to see if there are any differences.

How does the replicator know if replication needs to take place? It does this by examining hashes. A hash file is created for each Partition, which contains hashes of each directory in the Partition. Each of the three hash files is compared. For a given Partition, the hash files for each of the Partition's copies are compared. If the hashes are different, then it is time to replicate and the directory that needs to be replicated is copied over.

The cluster has eventually consistent behavior where the newest data wins.



1 In'the'event'of'a'zone'failure,'one'of'the'storage'nodes'containing'a'replica'will'no9ce'

2 The'node'will'then'proac9vely'copy'data'to'a'new'handoff'loca9on

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Component inter-relationship?

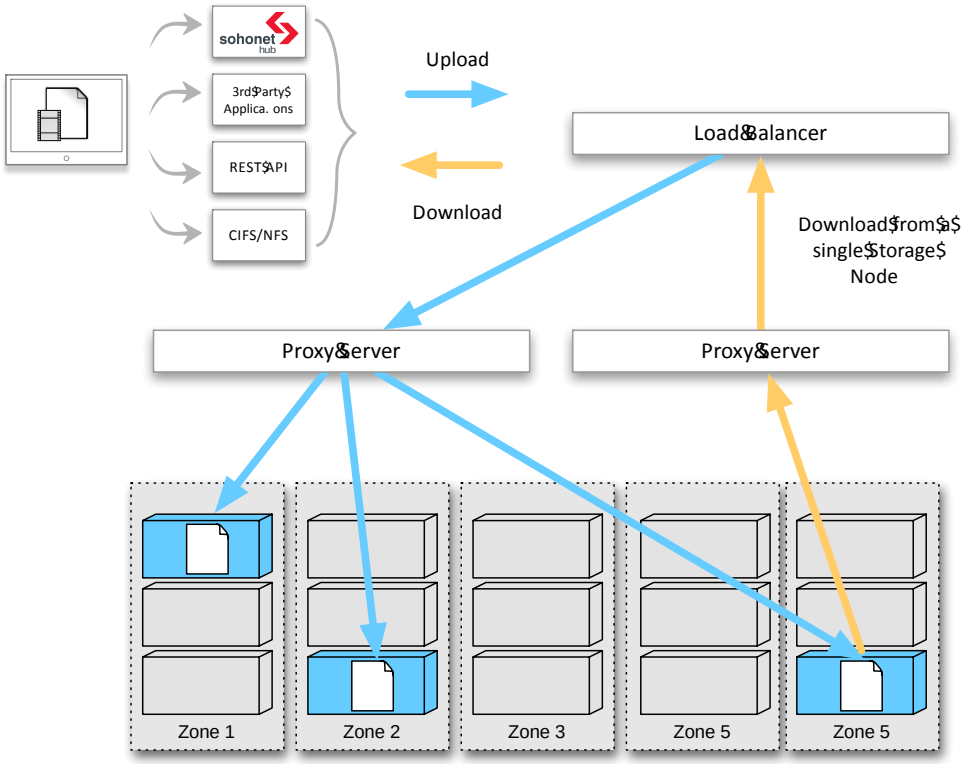
To describe how these pieces all come together, let's walk through a few scenarios and introduce the components.

Upload

A client uses the REST API to make a HTTP request to PUT an object into an existing Container. The cluster receives the request. First, the system must figure out where the data is going to go. To do this, the Account name, Container name and Object name are all used to determine the Partition where this object should live.

Then a lookup in the Ring figures out which storage nodes contain the Partitions in question. The data is then sent to each storage node where it is placed in the appropriate Partition. A quorum is required – at least two of the three writes must be successful before the client is notified that the upload was successful.

Next, the Container database is updated asynchronously to reflect that there is a new object in it.



T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Download

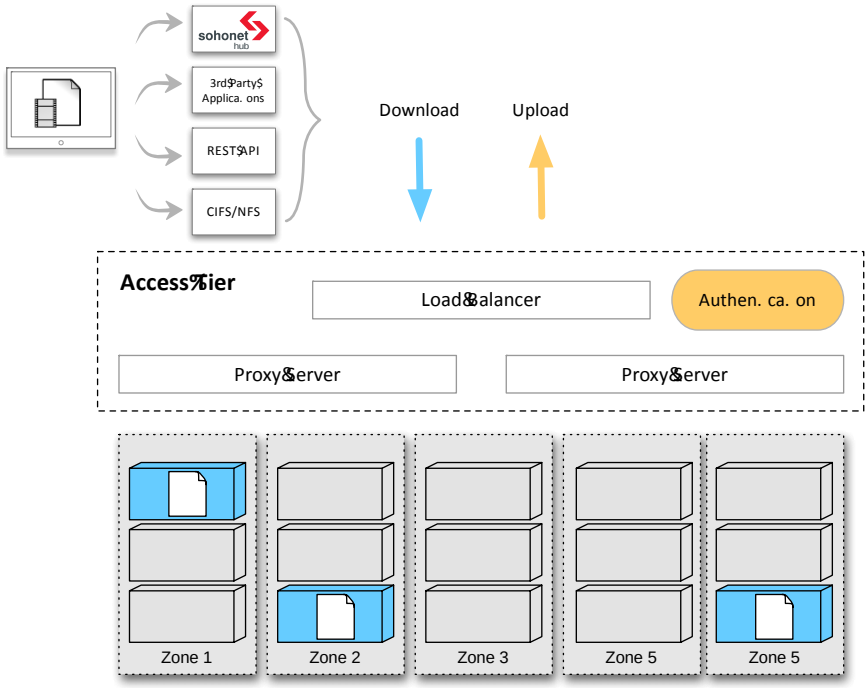
A request comes in for an Account/Container/Object. Using the same consistent hashing, the Partition name is generated. A lookup in the Ring reveals corresponding storage nodes contain a Partition. A request is made to one of the storage nodes to fetch the object and if that fails, requests are made to the other nodes.

Sohonet Media Storage Cluster Architecture

Access Tier

Sohonet Media Storage segments off an "Access Tier". This tier is the "Grand Central" of the Object Storage system. It fields incoming API requests from clients and moves data in and out of the system.

This tier is composed of front-end load balancers, ssl-terminators, authentication services, and it runs the (distributed) brain of the object storage system — the proxy server processes.



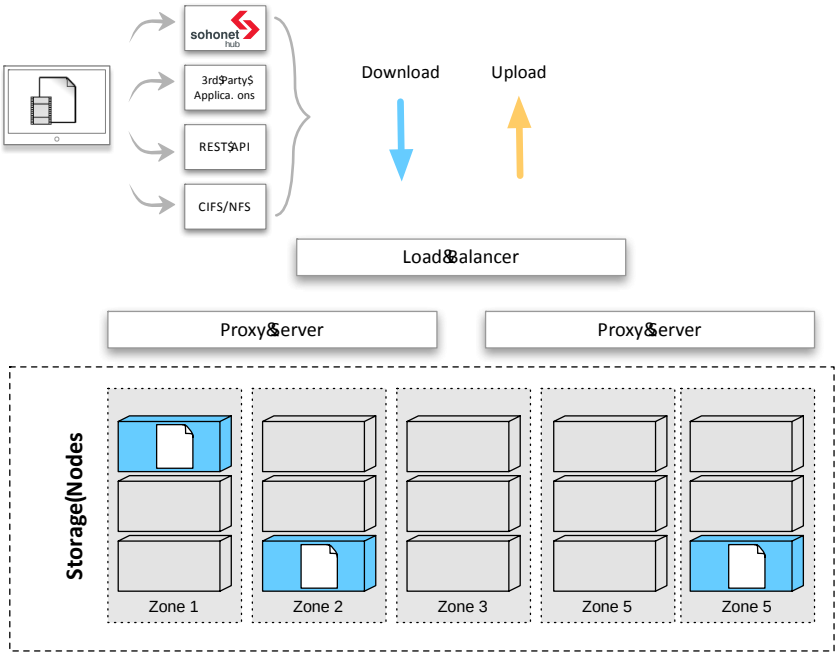
Having the access servers in their own tier enables read/write access to be scaled out independently of storage capacity. As this is an HTTP addressable storage service, a load balancer is incorporated into the access tier.

Typically, this tier comprises a collection of 1U servers and they use a moderate amount of RAM and are network I/O intensive. As these systems manage each incoming API request the IO is via two high-throughput (10GbE) interfaces. One interface is used for 'front-end' incoming requests and the other for 'back-end' access to the object storage nodes to put and fetch data. Users can use one or more of four interfaces, Sohonet Hub, 3rd party software that supports OpenStack, Sohonet Media Storage API, Sohonet Media Storage CLI and 3rd party hardware providing protocol translation. The table on page 8 highlights the key differences and use cases for each interface.

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Storage Nodes

The next component is the storage servers themselves. Generally, most configurations should have each of the five Zones with an equal amount of storage capacity. Storage nodes use a reasonable amount of memory and CPU. Metadata needs to be readily available to quickly return objects. The object stores run services not only to field incoming requests from the Access Tier, but to also run replicators, auditors, and reapers. Object stores can be provisioned with single gigabit or 10 gigabit network interface depending on expected workload and desired performance.



Considerations

To achieve apparent higher throughput, the object storage system is designed with concurrent uploads/downloads in mind. The network I/O capacity (1GbE, bonded 1GbE pair, or 10GbE) should match your desired concurrent throughput needs for reads and writes.

Actual transfer rates between your facility and the Sohonet Media Storage can run up to line speed i.e. 100Mbps, 1Gbps and 10Gbps with a typical daily transit of 1TB, 10TB and 100TB respectively. Obviously, this does depend on 1) current utilization of your connection to the Sohonet Media Network, 2) facility local/on-site disk performance.

For saturated SMN connections, it is recommended that customers upgrade their SMN connection to ensure that storage transfers do not effect pre-existing connection and networking experience. Please speak to our technical sales team to discuss the various networking options available i.e. network shaping or additional network connection.

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Why is Sohonet Media Storage relevant?

Sohonet Media Storage is Scalable

The system uses a shared-nothing approach and employs the same proven techniques that have been used to provide high availability by many web applications.

Sohonet Media Storage is Extremely Durable

Sohonet Media Storage is architected to withstand hardware failures without any downtime. To achieve this level of durability, objects are distributed in triplicate across the cluster. A write must be confirmed in two of the three locations to be considered successful. Auditing process run to ensure the integrity of data. Replicators run to ensure that a sufficient number of copies are in the cluster. In the event that a device fails, data is replicated throughout the cluster to ensure that three copies remain.

Another feature is the ability to define failure zones. Failure zones allow a cluster to be deployed across physical boundaries, each of which could individually fail.

Sohonet Media Storage is based on Open Standards and Open Source Software

As an open standards based system, Sohonet Media Storage provides the following benefits to its users:

1. **No vendor lock-in** – As an open source project, with open interfaces.
2. **Large ecosystem** – With the large number or organizations and developers participating in the OpenStack project, the development velocity and breadth of tools, utilities and services for Sohonet Media Storage will only increase over time

The source is reviewed by many more developers in comparison to proprietary software. The key benefit is that potential bugs also tend to be more visible and more rapidly corrected than for proprietary software. In the long term, “open” generally wins. Sohonet is a contributor to the open source code for OpenStack and we have written media specific optimization code as part of the development of Sohonet Media Storage.

Sohonet Media Storage uses Similar API to AWS S3

Access to the Sohonet Media Storage object storage system is through a REST API, which is similar to the Amazon.com S3 API and compatible with the Openstack API. This means that (a) applications that are currently using S3 can use Sohonet Media Storage without major re-factoring of the

application code and (b) applications that like to take advantage of both private and public cloud storage can do so as the APIs are comparable. Reusing of storage access source code needs to take into account the Sohonet security policies and utilization within the Sohonet Media Network.

Since Sohonet Media Storage is comparable with public cloud services, developers & systems architects can also take advantage of a rich ecosystem of commercial and open-source tools, which is available for these object storage systems.

Sohonet Media Storage uses Industry-standard Components

Under the hood, Sohonet Media Storage uses and built on proven components that work in large-scale production environments, such as rsync, MD5, sqlite, memcache, xfs and python. Sohonet Media Storage runs on Linux distribution, which is different from most other storage systems, which run on proprietary or highly customized operating systems.

From a hardware perspective, Sohonet Media Storage is designed ground up to handle failures so that reliability on the individual component level is less critical. Hardware quality and configuration are chosen to suit the tolerances of the application and the ability to replace failed equipment.

Sohonet Media Storage is privately deployed Infrastructure As-a-Service

For organizations uncomfortable storing their data in a public cloud, Sohonet Media Storage is an excellent alternative which allows you to retain control over network access, security, and compliance. Cost is also a major factor for bringing cloud storage in-house. Public cloud storage costs include per-GB pricing plus data transit charges, which can become very expensive.

The network latency to public storage service providers may also be unacceptable. Sohonet Media Storage provides lower-latency access to storage, as required by many applications. It is not practical to transfer large media datasets over the Internet. Hence, the Sohonet Media Storage is fully integrated within the Sohonet Media Network that is private, secure and proven global network solution. With dedicated SMN connections we can move data line speed albeit at 100Mbps, 1Gbps or 10Gbps. This effectively means we can move 1, 10 and 100TB of data per day.

For the above reasons, organizations can use Sohonet Media Storage to extend in-house storage systems that has similar durability/accessibility properties and is compatible with the suites of tools available for public cloud storage systems.

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Summary

Sohonet Media Storage provides extensible, secure and flexible storage that provides additional and supplementary storage capability to existing infrastructure via a private and secure Sohonet Media Network connection, transferring at 100Mb/s, 1Gb/s and 10Gb/s facilitating typical high I/O performance of 1, 10 and 100TB per day.

Sohonet Media Storage uses the combination of performance disks, chassis hardware and open source software to provide a balance between capacity, flexibility and price and has been optimized for the demands of the media industry to ensure sustained utilization and quality of service. We provide storage as a service, that has the benefits of near-line storage, but the scale of traditional tape based archives. Effectively, Sohonet Media Storage is referred to as tier-2.5. Essentially, Sohonet Media Storage provides all the performance, flexibility and security benefits of tier-2 storage with the scale of tier-3 storage, allowing immediate access to off-premise storage.

Typical uses of Sohonet Media Storage are backup, parking, short-term archive, alternative to LTO, backoffice backup, transition storage between ingest and post-production, shadow storage and replacement of other portable media. This provides customers the ability to utilise Sohonet Media Storage as remote storage that is designed for write many, read many and modify none.

Over the past 18 months, we have provided storage as a service for a variety of production, post-production and industry association projects ranging from 1TB back up to many 100's of terabytes for full feature production and digital negative storage.

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

APPENDIX A – API and CLI

Sohonet Media Storage API Commands - The Basics

The Sohonet Media Storage has a REST-ful API and uses HTTP verbs (PUT, GET and POST) to signal the requested action. A Sohonet Media Storage URL looks like this:

```
https://storage.sohonet.com/account/container/object
```

Sohonet Media Storage 's URLs have four basic parts. Using the example above, these parts are:

- Base: Sohonet Media Storage .example.com/v1/
- Account: An account is determined by the auth server when the account is created.
- Container: Containers are namespaces used to group objects within an account
- Object: Objects are where the actual data is stored in Sohonet Media Storage . Object names may contain /, so pseudo-nested directories are possible.

To get a list of all containers in an account, use the GET command on the account:

```
GET https://storage.sohonet.com/account/
```

To create new containers, use the PUT command with the name of the new container:

```
PUT https://storage.sohonet.com/account/new_container
```

To list all object in a container, use the GET command on the container:

```
GET https://storage.sohonet.com/account/container/
```

To create new objects with a PUT on the object:

```
PUT https://storage.sohonet.com/account/container/new_object
```

The POST command is used to change metadata on containers and objects. When planning a Sohonet Media Storage deployment, the first step is to define the application use cases, workloads and functional requirements determining how your Sohonet Media Storage is configured and managed by Sohonet.

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Sohonet Media Storage CLI Commands - The Basics

The command line usage for swift, the CLI tool is:

```
swift (command) [options] [args]
```

Here are the available commands for swift.

```
stat [container] [object]
```

Displays information for the account, container, or object depending on the args given (if any).

```
list [options] [container]
```

Lists the containers for the account or the objects for a container. -p or -prefix is an option that will only list items beginning with that prefix. -d or -delimiter is option (for container listings only) that will roll up items with the given delimiter, or character that can act as a nested directory organizer.

```
upload [options] container file_or_directory [file_or_directory] […]
```

Uploads to the given container the files and directories specified by the remaining args. -c or -changed is an option that will only upload files that have changed since the last upload.

```
post [options] [container] [object]
```

Updates meta information for the account, container, or object depending on the args given. If the container is not found, it will be created automatically; but this is not true for accounts and objects. Containers also allow the -r (or -read-acl) and -w (or -write-acl) options. The -m or -meta option is allowed on all and used to define the user meta data items to set in the form Name:Value. This option can be repeated.

Example: post -m Color:Blue -m Size:Large

```
download —all OR download container [object] [object] …
```

Downloads everything in the account (with —all), or everything in a container, or a list of objects depending on the args given. For a single object download, you may use the -o [—output] (filename) option to redirect the output to a specific file or if "-" then just redirect to stdout.

```
delete —all OR delete container [object] [object] …
```

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom

Deletes everything in the account (with —all), or everything in a container, or a list of objects depending on the args given.

Example: swift -A https://auth.api.example.com/v1.0 -U user -K key stat

T+44 (0) 20 7292 6900 sales@sohonet.co.uk www.sohonet.com
Sohonet Limited, 5 Soho Street, London, W1D 3DG United Kingdom