Object Matrix
Object Storage Overview

Version 1.3.1, Nov 2014, Jonathan Morgan

Contents

1 Overview

In this Digital Information age appropriate storage systems are required to store and make available data.

Over 2 trillion objects are estimated to be stored on Amazon S3's object storage platform and Microsoft claim 8.5 trillion objects are stored in their Azure platform. IDC and EMC project that data will grow to 40 **zettabytes** by 2020; a 50-fold growth from the beginning of 2010. Computer World states that unstructured information might account for more than 70%-80% of all data in organizations.

Object vendors include Bycast, CleverSafe, DataDirect Networks (WOS), EMC (Centera, Atmos, ViPR), HDS (HCP), HP (HP OpenStack), IBM, NetApp (StorageGRID), Nirvanix, **Object Matrix**, and Scality. Cloud service vendors include Amazon (AWS S3), Google (Google Cloud Storage) and Microsoft (Microsoft Azure).

- What's the difference between **traditional storage systems** and **Object Storage**?
- When does Object Storage make sense over **filesystems**?
- What architectures of Object Storage are there?
- What is **Unstructured Data**?

This whitepaper seeks to set out the differences in Object Storage over traditional storage solutions, the differences between selected architectures of Object Storage and to provide examples of where Object Storage provides the most relevant storage solution, providing examples in media workflows.

2 History

2.1 The Need

2.1.1 Scale as a Driver

There has been a revolution in how data is stored and accessed because more data is being stored than ever before, a greater proportion of that data is unstructured data and the way data is accessed and used has changed.

Thinking back to the 1960's through to even the past few years, the ubiquitous filesystem has been the dominate force behind online[1] data storage.

But a filesystem is limited in multiple ways. A filesystem:

- essentially provides a single structured view of data[2]
- identifies data through labels that are not globally unique
- typically either doesn't support metadata at all, or at least doesn't support it in a manner that encourages distributed search across multiple computers, doesn't encourage sharing of that metadata between applications due to their proprietary interfaces, and doesn't share that metadata through the lifetime of that data (**tiers of storage**)
- has limited scaling (traditional large filesystem solutions (**SAN / NAS**) make great strides to try to overcome the inherent bottleneck of a single metadata controller, normally through deployment of several **highly-coupled** metadata controllers, but yet still have a reputation for corruption, volatility, and limited scalability)

What the filesystem couldn't do in terms of metadata handling, relational databases were added to achieve. But again, inherent architecture limitations mean that relational databases:

- have limited scale
- complexity in set-up, maintenance and upgrade requiring specialist skills and – and as databases are clustered and requirements are susceptible to corruption in disaster scenarios

Modern needs can include ability to store trillions of objects, have completely scalable architectures, have worldwide availability, be simple to maintain and upgrade, and to provide metadata handling without the need for additional databases … to name but a few requirements.

Filesystems simply cannot cope with such requirements.

---

[1] Online as opposed to offline (on a tape / optical) data storage
[2] Ad-hoc bolt-ons can include virtual file links and alternative vitualised top-down views but inherently a filesystem is still a top-down architecture

Databases are complex to set-up, use and maintain. They also don't scale well beyond a certain point.

2.1.2 Economics as a Driver

Consider that an organisation has collected several petabytes of information through the past few years. Historically, that data would have either been never stored or perhaps stored on media that is overwritten. Or perhaps that data would have been stored onto removable media such as USB drives or data tape.

Now consider that there is an opportunity value of analysing or accessing that data, e.g., to perform a data analysis of customer interactions, or to provide a media clip for re-usage.

If the *opportunity value exceeds the total cost of storage and retrieval* then it makes economic sense to keep that data.

Increasingly, in this Google age of instant data access, organisations are finding ways to monetise their information histories, and increasingly, the demands are for instant access to that information.

It all comes down to that algorithm and does the *opportunity value exceeds the total cost of storage and retrieval.*

Whilst physical hardware costs continue to fall (in terms of $/Terabyte) labour costs have continued to rise. Efficiencies of storing, finding and managing data have therefore become more about that "4th dimension" cost… time: management time, time to discover and time to store/retrieve than the physical media costs. Hence the growth of need for massively scalable storage solutions within private organisations. Sometimes this is labelled as Private Clouds.

2.2 Solution History: Grid Computing through to Object Storage

Grid Computing in many ways was the predecessor and is closely related to Object Storage.

The term *grid computing*[3] originated in the early 1990s as a metaphor for making computer power as easy to access as an electric power grid. In 1999 the book "The Grid: Blueprint for a New Computing Infrastructure" was published (Morgan Kaufman Series). Grid computing combines computers from multiple administrative domains to reach a *common goal, to solve a single task*, and may then disappear just as quickly.

This is achieved by *a Grid*. "A Grid is a system that:

1. Co-ordinates resources that are not subject to centralised control…
2. …using standard, open, general-purpose protocols and interfaces…

[3] According to http://en.wikipedia.org/wiki/Grid_computing

3. …to deliver nontrivial qualities of service"[4]

Clustered computing is *a grid* that typical runs within a local area network. Clusters tend to be used for IO intensive operations where high network connectivity bandwidth is required.

Now for Object Storage. Object storage was proposed[5] at Carnegie Mellon University's Parallel Data Lab as a research project in 1996. Research by Garth Gibson, *et al.* on Network Attached Storage Disks. The project promoted the concept of splitting less common operations, like namespace manipulations, from common operations, like reads and writes, to optimize the performance and scale of both.

Object Storage naturally fits on top of a Grid or on top of a Cluster. Except where explicitly stated, we discuss where Object Storage is used on a Cluster architecture.

Object Storage has come to market prominence over the past dozen or so years. 2002, EMC purchased a Belgian company FilePool through whom they developed and launched Centera. In many ways this can be considered "Object Storage v1". Adopted by a reported 3,500 customers, the product was labelled as Content Addressable Storage (CAS), meaning that the data contents defined the globally unique identifiers of the data objects stored within. Whilst meeting a large amount of criteria for an Object Storage solution, it provided only limited metadata services, limited scalability (due internal data structures and to **nodes** being overly highly coupled) and limited performance via all data needing to be sent/received through head nodes.

Newer object storage solutions have generally dropped the concept of CAS objects, preferring rather to use content-independent geographically unique identifiers (GUIDs). Many companies have been heavily funded to push forward the concepts of object storage, to break down the barriers of scale and to provide various **data services**. We look at various definitions and classifications of object storage in Architecture.

[4] Globus ToolKit 4, Morgan Kaufmann, 2006.
[5] According to http://en.wikipedia.org/wiki/Object_storage, however this claim is highly open to debate

3 Architecture

There is a commonality in “object storage” characteristics but there are also widely varying implementations of object storage solutions that might make one better than another e.g., archive of many large data objects whilst another solution might be better for, e.g., simultaneous random access and update of data in multiple geographically dispersed locations.

3.1 General Architecture

Commonly desired characteristics of Object Storage systems are:

1. **Support the capabilities of Grid Computing:**
    a. **Ease of management**. Centralised control.
    b. Standard, open, general-purpose protocols and interfaces.
    c. Deliver non-trivial data services
2. **Support the desired capabilities of Mass Data Storage:**
    a. **Scale.** E.g., to multiple Petabytes / billions of objects.
    b. **Preservation.** E.g., Data shouldn’t be lost due to hardware failures. Often this is achieved through self-healing strategies.
    c. **Accessibility.** Being able to find the data that is required and to be able to retrieve that same data at a total cost, measured in time and manpower, that is less than the opportunity value of that data. Systems should be adaptable to new workflows.
    d. **Good TCO**. To keep **Total Cost of Ownership** (**TCO**) low compared to alternatives. I.e.:
        i. Soft costs: support costs (internal / external)
        ii. Hard costs: purchase price, maintenance price (in the future moving data from one tape / disk to a newer one, etc.), power, space, etc.
        iii. To keep Opportunity cost of data usage lower than TCO.
    e. **Long-term strategy.** E.g. (1), **Future Skills**. Avoid undue skills requirements – employees and procedures will change and complex systems will be neither understood nor adaptable. E.g. (2), avoid hardware obsolescence issues.
3. **Support generic Objects:**
    a. **Globally unique identifiers** for objects (GUID’s)
    b. **Data**. Store and retrieve object data.
    c. **Metadata**. Store user and content extracted metadata and to augment that data with environmental metadata (date/time stored etc). Allow search of that metadata.
    d. **Policies**. Allow for various data storage policies such as number of instances of data to keep, location of objects, mutability of objects.

The architecture of an object storage solution consists of:

Fabric: The collection of individual computers, storage devices, CPUs, databases, etc., providing the object storage solution. For the sake of parlance we define a node as one storage location of which the Fabric may contain many.

Connectivity: Consisting of communication and security modules.

Data services: For internal (one node), intra-cluster (between nodes within a cluster), inter-cluster and external (to third party computers) usage.

3.2 The Software Architecture

The underlying software architectural decisions made when defining an Object Storage solution will have a fundamental affect on the suitability of the implemented solution for certain tasks over other tasks. E.g., the Amazon S3 architecture may be highly suited to storing trillions of objects with global access, whereas the EMC Isilon solution is suitable for usage with a high-speed filesystem interface and Object Matrix's MatrixStore is a hybrid of the two. If a solution is designed to be highly loosely-coupled then it can cope well with various speeds and efficiencies of nodes whereas if a solution is highly coupled it may be able to obtain higher levels of quality of service with I/O throughputs. A solution with global dispersement of data into local object storage caches may not work very efficiently if simultaneous object data updates are required when compared to an architecture where that keeps data management centrally.

No object storage solution is necessarily the *best* – each is designed to solve particular requirements and therefore each is unique.

3.2.1 Object Storage Organisation

IDC MarketScape: Worldwide Object Based Storage (OBS) 2013 Vendor Assessment compares various solutions under the following criteria:

Data organisation, Persistent Data Stores, Storage Services and Delivery Model.

Object Matrix has its own definition of data organisation in:

Table 1: Classifications of Data Storage Organisation.

3.2.2 Object Storage Comparison

Table 1: Classifications of Object Storage architectures shows the classifications of object storage solution that will be used to compare object storage solutions.

All solutions compared assume that they have a global namespace.

Table 1: Classifications of Data Storage Organisation

Type	Features	Pros	Cons
geo-dispersement e.g. Amazon Web Services S3, Microsoft Azure, Google Cloud Storage.	Objects may be accessed from all over the world. Typically HTTP data transmission protocols.	Geographically handle / distribute instances of the data to where they are being accessed from. Typically capable of storing trillions of objects. Typically strong at being loosely-coupled / work with various hardware generations. No special client-side software required.	Typically feature external data analytics. Typically varied quality of service on bandwidth. Typically slower than solutions with client-side transmission protocol installations.
CAS e.g., Caringo, EMC Centera.	Data is identified by a digest generated from the data contents.	Typically have data de-duplication features built-in. Typically (but not necessarily) have regulatory compliance features (such as data immutability guarantees).	Not typical where data needs to be updated (because updating an object changes the digest and therefore changes the ID of the object).
Object based File Systems e.g., Lustre, Hadoop HDFS [6]	Metadata stored to metadata servers, data to data servers. File system view of data	Scalable, fast (in some scenarios) file system	Complex set-up, installation and maintenance. Complex structures (metadata databases) must be maintained.
Erasure Code	Objects stored into the	Can be less hardware space consuming than	Can cause extra CPU load to destruct/reconstruct

[6] Hadoop is not strictly object-based but is used in conjunction with Object Storage solutions such as Scality.

Solutions e.g., CleverSafe, Isilon, Scality, Amplidata	solution are split using erasure codes to ensure redundancy in a RAIN architecture	solutions where data instances are kept whole.	data, in particular during random access file system updates. Solution runs at the speed of the slowest node so typically all nodes should be the same configuration in order to be able to give a quality of service. Scaling typically requires a new set of nodes (if the erasure code requires 8 locations and the original 8 locations are full, then a complete set of new nodes must be purchased).
Standard Object Storage e.g., Object Matrix MatrixStore	Objects stored whole in multiple instances across a RAIN architecture.		

Table 2: Storage Comparison Chart

Storage Name	Object Matrix MatrixStore	Scalable NAS/SAN	EMC Isilon	Scality	Amplidata	Quantum Lattus	Caringo, etc.	S3 etc	Avid ISIS 2000[7]
Data Organisation Type	Object Storage	File System	Erasure codes	Erasure codes	Erasure codes	Erasure codes	CAS	Geo-Dis	Proprietary
Scale									
Entry Level Solution	36TB	1TB…	72TB	-???-	-???-	216TB ??[8] Only 126TB usable	-???-	1 byte	120TB

[7] http://www.avid.com/US/products/ISIS2000/specifications
[8] http://www.quantum.com/products/bigdatamanagement/lattus/index.aspx - Datasheets

Scale	1 Billion objects. 10's Petabytes.	10 Million's files. 1's of Petabytes.	Billions of objects. 10's Petabytes.	Trillions of objects. 10's Petabytes.	Billions of objects. Exabytes.	Billions of objects. 10's Petabytes.	Billions of objects. 10's Petabytes.	Trillions of objects. Exabytes	10 Million files. Up to 1.2PB.
Loosely-Coupled	☑	☒	☒	☑	☑	☑	☑	☑	☒
Multi-Tenancy	☑	☐[9]	☑[10]	☑	☑	☑	☑	☑	☒[11]
Inherent HSM Support	Yes, allows objects to be stubbed but the metadata to be searchable	☒	☒	☒	☒	☒	☒	☒	☒
Delivery Model									

[9] Typically requires additional software to organise.
[10] Isilon has Access Zones which are distinctly authorized zones only available to authorized users. The underlying architecture is however a single namespace.
[11] Does provide "workspaces"

File System Access	Fuse, SMB, FTP	NFS, CIFS, AFP, FTP, Hadoop	SMB, NFS, Kerberos, NTLM, FTP, SSH, HTTP, iSCSI, Fibre Channel, NDMP, Hadoop	Fuse, NFS, CIFS, AFP, FTP, Hadoop	NFS/CIFS[12]	NFS/SMB (see footnotes)	POSIX compliant	Various 3rd party solutions	CIFS, FTP
API	Proprietary Java / C APIs, Proprietary Management API	☒[13]		Sproxyd, S3, REST, CDMI, Cinder API	REST, S3, iRODs, .NET	REST, S3, iRODs	REST, S3	S3	☒
Data Protection									
Protection Schemes Supported	Multiple RAID groups, replication, H/W RAID for speed	Multiple RAID groups, replication add-on, HW or SW RAID	Reed Solomon (N+M data protection)	N+M data protection, replication	Erasure codes (+50% on data size)	Erasure codes (+50% on data size)	N+M data protection, replication	??	Multiple RAID groups, H/W RAID for speed
Single Point of Failure	None	Maybe	None	None	None	None	None	None	None

[12] Access is through an external company gateway: http://maldivica.com/ or http://www.bridgestor.com/English/Products/Coronado_NAS_Gateway.html. Scality uses StorNext (on top of Amplidata) which Object Matrix would expect would give better performance, scalability and enterprise integrations.
[13] Some file systems have some proprietary APIs for, e.g., adding metadata to a file but such metadata tends to be standalone

Business Rules Support (Regulation Compliance)	WORM+, full auditing	☒	☒	☒	WORM options[14]	WORM options	WORM+, full auditing	☒	☒
Management									
Easy Expansion	Yes and h/w independent	No, very h/w dependent	Yes but much h/w dependency . Large minimum expansion sizes.	Yes and h/w independent No indication about minimums or h/w types.	Yes and h/w independent No indication about minimums or h/w types.	Yes and h/w independent No indication about minimums or h/w types.	Yes and h/w independent	Yes	Yes - max 1.2PB - h/w must match, 120TB minimum expansion
Built in Metadata Search	☑	☒	☒	☑	☑	☑	☑	☑	☒
Media Workflows									
Avid Interplay integration[15]	☑	☒	☒	☒	☒	☒	☒	☒	☑

[14] Additional features can be achieved with http://www.qstar.com/solutions/business-need/compliance/
[15] Some 3rd party add-ons, e.g., http://www.nltek.com/ can be used against generic hardware

EVS Certified[16]	☑	☑	☑	☒	☒	☑	☒	☒	☒
Wide range of media partner workflow integrations	☑	☑	☑	☒	☒	☑	☒	☒	☒
Extract Metadata from content[18]	☑	☒	☒	☒	☒	☒	☒	☒	☒

Other Solutions:

CleverSafe: Erasure code based scalable storage solution in many ways comparable to Scality.

[16] Object Matrix is being certified in 2014 for Nearline workflows.
[17] Primarily via StorNext integration.
[18] Coming in 2014.

4 MatrixStore Differentiators

Key MatrixStore strengths vs competition include the following:

1. **Media workflow focus**. It means the workflows we support and continue to add are all about media. Our strong presence and history in the market gives us a lead on that front. Free tools that come with the solution are also media focused, such as DropSpot for data ingest which allows additional metadata to be captured.
2. (Q4, 2014) Metadata enrichment through in-place content analysis. The enriched metadata is made available to all connecting applications.
3. Media company business model. Object Matrix models its support model and pricing on media workflow companies. Its staff are knowledgeable about media workflows and the demands of the industry.
4. Focus on making multi-year data storage easy: easy to expand with one node at a time and with the latest hardware.
5. Strong business rules support, such as auditing file changes. Strong security model.
6. Inherent HSM support is a great feature, although other solutions will have workarounds / use external 3rd party apps at additional cost.
7. Related to the above *Media focused business model*: OM solutions are both configured in appropriate sizes for the media industry and are plug and play appliances. Furthermore, OM sells Quattro units (from 4TB) and Mini units (from 1TB) to aid workflows that require replication from base stations to a data storage hub.

MatrixStore weaknesses against most other options:

Missing an S3 type interface – this would allow additional 3rd party tools to be used with MatrixStore, such as CyberDuck.

Requirement for RAID hardware (which adds extra cost).

The "Unknowns"

In media sales – nothing is a bigger differentiator that the workflow and if the storage can support it. For instance, StorNext might be well proven with EVS with one backend storage but add that to Lattus and then you might not have the performance to match the needs of the customer. Everything is about proven workflows.

Add to that the above advantages and MatrixStore is a compelling offering within the industry.

5 Object Storage in Media Workflows

Blog Article by Mark Andrews:

As a concept, object storage is not new – as a company, Object Matrix has been developing it for over 11 years. But it is recently starting to get traction in the media and broadcast industry being slated as the next big thing.

So what is Object Storage, and why is it good for media workflows?
Traditional storage relies on a file system interface – that is you present your network attached storage (NAS) as a file system interface such as a drive letter on Windows or as a volume on Mac or Linux. But traditional file systems have scalability issues – an upper limit on file numbers (typically millions) and as you approach this limit then performance becomes an issue. Object Storage does not rely on a file system to manage the content under its control and so in theory there is no upper limit on file numbers. This in turns allows you store petabytes and beyond of storage with no loss in performance. An object is in fact just a digital asset such as a media file.

And there are additional benefits. A file system does not allow you to store metadata with content – it's a limitation. However object storage does allow this making it searchable and intelligent about its own content without the need for a separate media asset management system. Or when using MAM's they can be configured to archive and protect metadata as well as the media files themselves. It is also possible to manage different objects with different data management rules such as number of copies and replication rules.
Facebook and Amazon S3 are built on Object Storage principles – Amazon S3 has trillions of objects under its management which it provides in an online cloud type environment. Object Storage must allow for loosely coupled inter-dependencies – in other words having different users and applications being able to access the content without impacting on each other. The MatrixStore architecture virtualises multiple nodes of physical hardware to enable it to be used securely by multiple users, and with minimal impact upon each other. Also, the flexibility of an object storage architecture means that multiple storage policies can be enforced, e.g., for how long objects are stored, how many instances of data are kept and the location of those objects.

So why object storage for media and broadcasting workflows?

Simply put, the media world has large storage expectations. Media files, rushes, sequences, finished projects and ongoing transcoding and repurposing requirements not only dictates a platform that can manage big volumes of data, but also one that can be performant enough to serve up content in real time. LTO tape is a traditional archive medium of choice, but with its limitations, media organisations such as broadcasters, VoD companies, advertising agencies and post-production companies are acquiring additional disk based "nearline" storage solutions for storing media and other digital files as their main media repository. Disk allows random access, is quick and performance is not limited by the number of tape-drives. Disk based Object Storage takes that further by allowing a very expandable and scalable solution that can be searched, is highly resilient, easy to use and can be delivered with commodity hardware that is easy to support. And all at a price point that is compelling.

Additional storage benefits of a disk based system such as the MatrixStore from Object Matrix:

- Self-Healing Functionality in the event of any hardware failure providing 99.999% uptime
- No "Lock in" to specific hardware – as you scale (as that is the point!) over time, grow your storage with the latest and greatest storage technology to expand your platform
- Authenticity check with "checksum", such as MD5 or Adler32, to ensure a bit-for bit copy when ingesting content from another platform
- Workflow Integration – proven integration with various technology platforms such as Avid and their Interplay MAM/DAM platform, Adobe, Apple, Cantemo,

CatDV, Harmonic, Glookast, MOG, Cambridge Imaging Systems, Aspera, Signiant, Masstech as well as an array of other transcoding, MAM and play out partners. Inter-dependency as described above allows sharing of applications on the same physical hardware without impacting on the security and data management needs of the other workflows.

- Various client tools including a 'virtual' file system for Windows, Mac or Linux (MXFS), an SMB interface, or DropSpot, our ingest and search tool. A file system can be assigned to each workflow and mounted as a separate 'logical' file system from the same physical hardware.
- You can have a very scalable single file-system to petabytes and beyond if you desired
- Easy to use and administer
- Easy to add storage and grow (In less than 1 minute) with re-balancing of content for improved performance
- Full support

6 Glossary

Term	*Definition*
API	Application programming interface.
Avid	www.avid.com Industry leading manufacturer of software (e.g., editing tools) and hardware (e.g., ISIS data storage) solutions in media workflows.
Data Services	Traditional storage simple stores the data and possibly given metadata. Object Storage typically has intelligence (CPU power) in the storage servers, so can add additional services such as metadata extraction, self-healing, replication services, distributed databases for searches, security, APIs, etc.
Data Storage Policy	See Policy.
De-Duplication	In whole object de-duplication: if the data of an object / file is an exact match to the data in a 2nd object / file then a single instance of the data can be kept with two references to that instance. Other schemes also exist (e.g., block level de-duplication, partial object de-duplication). De-duplication is common in storage servers storing e.g., email data that has multiple instances of the same attachment, but less common in media workflows where duplicate data is less frequent or are removed by external tool (e.g., Avid Interplay).
Erasure Codes	An erasure code[19] is a forward error correction (FEC) code for the binary erasure channel, which transforms a message of k symbols into a longer message (code word) with n symbols such that the original message can be recovered from a subset of the n symbols. The fraction $r = k/n$ is called the code rate, the fraction k'/k, where k' denotes the number of symbols required for recovery, is called reception efficiency.

[19] http://en.wikipedia.org/wiki/Erasure_code

	Put another way: an object is divided up into N + M parts where N represents the number of parts that are required to rebuild the object and M is the number of locations that can be "lost" whilst still being able to make up the original file. Typically, all parts are of an equal length (*L*) and N * *L* equals the original length of the file. Thus, M**L* is the data storage overhead. Normal drawbacks of erasure codes are that CPU power is required to deconstruct data for storage or to reconstruct data for retrieval. This can make some operations, such as random updates of data files slower than would otherwise be the case.
EVS	EVS (www.evs.com) are the leading software suppliers of solutions for live replay such as in sports coverage.
Exabyte	1 Exabyte = 1000 Petabytes, or 10^{18} bytes.
External Data Analytics	Traditional storage and most Object Storage solutions[20] require data to be read from the storage devices to a server with CPU power / software algorithms to analyse that data. This requires data to be transferred out of the storage server, network bandwidth and CPU power to make the data transfer.
FileSystem	Filesystems provide a top down hierarchical view of the data that they contain. Typically the structure of the filesystem is contained with one (or more) metadata controllers and files are served (or written through) those controllers to block based storage devices.
Hardware Obsolescence	Traditional storage and some Object Storage solutions may be expandable in terms of capacity but require the hardware used to expand the cluster to be of the same or similar generation as the hardware when the cluster was purchased. This can be an extremely limiting factor.
Highly-Coupled	Opposite of Loosely-Coupled.
Media Workflows	The storage of data that is video, audio in nature.

[20] Notable: MatrixStore avoids the need to externalize the data for analytics in some circumstance since data can be processed in place by the direct attached CPU.

	Traditional media companies are broadcasters, film makers etc, but actually most large organizations have media workflows, e.g., for CCTV, company meetings, audio recordings etc.
Metadata	Metadata[21] is data about data. Metadata can be tags that are added by the user, but that metadata can also be augmented by the examination of the data that was stored. That process is called Metadata Extraction or Content Analysis. Metadata is typically put into a Metadata Database.
Metadata Extraction	See Metadata.
Metadata Database	Some storage solutions (e.g., the XFS filesystem) will simply put any metadata into a key-value database that can be examined by the user through an API on a location by location basis. Other more advanced solutions will compile the metadata into a distributed database such that the database can be searched using more complex search terms and across the entire data set.
N+M data protection	See Erasure Codes.
NAS	See SAN.
Nodes	A node, within the context of this document, is a computer server with CPU, network connectivity (both to other nodes and to externally attached clients) and storage capacity.
Object Matrix	www.object-matrix.com
Object Storage	Data is stored together with its metadata and policies that control how the data is kept (1 instance, 2 instances, etc).
Petabyte (PB)	1 Petabyte = 1000 Terabytes, or 10^{12} bytes.
Policy	A data storage policy is a set of instructions to the storage manager (e.g., the Object Storage system) that defines how the object should be stored. Parameters might include (depending on the system) the period of

[21] http://en.wikipedia.org/wiki/Metadata

	immutability of the object, the number of instances of the object that should be kept, the geo location(s) of that the object should be kept in.
Private Clouds	In the context of this document a private cloud to a storage solution where from the users perspective data is simply checked in and checked out without concern about where that data is actually kept, and from a management perspective it is virtually management free.
Quality of Service	Quality of services refers to the storage solution being able to guarantee one or many uninterrupted stream(s) of data storage or retrieval. This is particularly important in some media workflows where, for instance, a video is being played out to a watcher without buffering, or where a stream of data is being ingested live from a camera.
Redundant Array of Independent Nodes (RAIN)	Redundant array of independent nodes[22] (RAIN) is a disk subsystem that provides distributed data storage and protection in network architecture by integrating inexpensive hardware and management software. RAIN is designed to offer scalable and reliable network-attached storage (NAS) by combining off-the-shelf distributed computing and commodity hardware with sound management software. It is designed to improve on the shortcomings of non-redundant NAS systems. The concept of RAIN is derived from redundant array of independent disks (RAID), which is a similar system that is implemented at the disk level. Redundant array of independent nodes may also be called redundant array of inexpensive nodes.
Regulatory Compliance	Governments and organisations have from time to time created laws or rules that define how data must be kept, found, audited and deleted. Object Storage systems (in particular) can aid compliance to those rules by providing data services related to the mutability, auditing, lifeline and deletion of data.

[22] http://www.techopedia.com/definition/1106/redundant-array-of-independent-nodes-rain

SAN	SAN (Storage Area Networks) and NAS (Network Attached Storage) are generally seen by users as a shared filesystem to which they typically connected to via internal networks of the organisation or via VPN. Typically the SAN or NAS comprises of storage devices and a filesystem metadata controller.
Self-Healing	Within a RAIN architecture, should a storage controller or node realise that a storage location is unreachable for a period of time then it might elect to recover the data that is contained within the offline storage location. It achieves this by locating a good instance of the data on an online storage location (or in the case of erasure codes, it might regenerate the fragment of the object that was contained within the offline node).
Tiers of Storage	Sometimes different types of storage are described as tiers of storage, in particular where the first tier of storage is typically an area where the data is volatile and perhaps (in media workflows) being edited, the second tier of storage is referred to as "nearline" and can be a scalable archive of data, and the third tier of storage, which is referred to as "archive" or "deep archive" is data that is being kept for the long term, on disk or on tape. Through the lifetime of a piece of data it may be moved from one tier of storage to the next.
Total Cost of Ownership (TCO)	TCO[23] analysis was popularized by the Gartner Group in 1987. The roots of this concept date at least back to the first quarter of the twentieth century. Many different methodologies and software tools have been developed to analyse TCO. TCO tries to quantify the financial impact of deploying an information technology product over its life cycle. These technologies include software and hardware, and training. Object Matrix has calculated a full TCO model for data storage: http://www.matrixstore.net/2010/02/23/a-living-tco-model/

[23] http://en.wikipedia.org/wiki/Total_cost_of_ownership#Computer_and_software_industries

Traditional Storage	Filesystem based storage, typical with data stored in block-based storage devices, and having metadata controllers.
Unstructured Data	Data that is not organised in a pre-defined data model. In reference to Object Storage, the following data types are typical of those labelled as unstructured data: text heavy data, video assets, audio assets, streams of analytical results. Data that, e.g., resides in a database, would typically be considered structured data.
Workflows	The movement of data through its lifecycle within an organisation, including but not limited to, editing, transcoding, adjoining, addition of metadata etc. Generally many different software tools will touch and use a piece of data throughout its lifetime, and each of these form a part of the workflow.
Write Once Read Many (WORM)	A storage technology that allows data to be written to a device after which time the data becomes immutable (including undeletable). The data may be read many times. An example of such a technology are compact disks (CDs) however a centralised storage location may also make data immutable.
WORM+	WORM+ refers within the context of this document to a storage system that can make a piece of data WORM like for a pre-defined period of time, after which the data becomes mutable again.
Zettabyte (ZB)	1 Zettabyte = 1,000 Exabytes or 1,000,000 Terabytes, or 10^{21} bytes.

Appendix A – Miscellaneous additions

CAP Theorem

The CAP theorem is sometimes used to describe data consistency and integrity policies of distributed RDMS platforms and therefore lends itself well to describing Object Storage platforms.

The CAP theorem states that it is impossible for a distributed computer system to simultaneously provide all three of the following guarantees:

> Consistency, Availability and Partition Tolerance.

Consistency is the ability of all nodes, regardless of updates or deletes, to see the same data at the same time.

Availability is a guarantee that every request receives a success or failure.

Partition tolerance is the ability of the Object Storage to continue to operate despite failure of part of the system or lack of communication between nodes (also known as split brain).

Geo-distributed Object Storage solutions will tend to offer strong partition tolerance and lower levels of consistency guarantees, whereas, Object Storage solutions such as MatrixStore will tend offer greater consistency guarantees and lower partition tolerance. All such systems have workarounds and offer sub-guarantees to the CAP theorem, e.g., MatrixStore will redirect a client to the “larger side” if a single node is in a split brain and is accessed.

Online IDC Reports

Scality:

http://info.scality.com/rs/scality/images/IDC_Marketscale_OBS_VendorAnaysis.pdf

Amplidata/Lattus:

http://amplidata.com/wp-content/uploads/2013/11/Amplidata-IDC-MarketScape-2013.pdf

http://www.slideshare.net/QuantumCorp/introduction-to-quantums-lattus-for-wide-area-storage