



## D5.3 Architecture for VSRI

---

**WP 5 – Software Architecture for Digital Preservation**

Delivery Date: 31/03/2013

Dissemination Level: Public

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

<b>Deliverable Lead</b>		
<b>Name</b>	<b>Organisation</b>	<b>e-mail</b>
Rodger Mooney	Intel	rodger.mooney@intel.com

<b>Contributors</b>		
<b>Name</b>	<b>Organisation</b>	<b>e-mail</b>
Mykola Galushka	SAP	mykola.galushka@sap.com
Michael Nolan	Intel	michael.nolan@intel.com
Phil Mondor	Intel	philip.l.mondor@intel.com
Carlos Coutinho	CMS	carlos.coutinho@caixamagica.pt
Luís Veiga	INESC-ID	luis.veiga@inesc-id.pt
Paulo Ferreira	INESC-ID	paulo.ferreira@inesc-id.pt

<b>Internal Reviewer</b>		
<b>Name</b>	<b>Organisation</b>	<b>e-mail</b>
Daniel Simon (PCC)	SQS	daniel.simon@sqz.com
Anja Bachmann (Internal)	KIT	bachmann@teco.edu
Roxana Belecheanu	SAP	roxana.belecheanu@sap.com

## Disclaimer

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law. Copyright 2013 by Intel, INESC-ID, CMS and SAP.

D5.3_M24_Architecture_For_VSRI	Dissemination Level: Public	Page 2
--------------------------------	-----------------------------	--------

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

## Table of Contents

<b>TABLE OF CONTENTS .....</b>	<b>3</b>
<b>LIST OF FIGURES.....</b>	<b>6</b>
<b>LIST OF TABLES.....</b>	<b>8</b>
<b>LIST OF ACRONYMS.....</b>	<b>9</b>
<b>1 EXECUTIVE SUMMARY .....</b>	<b>10</b>
<b>2 INTRODUCTION.....</b>	<b>11</b>
2.1 BACKGROUND .....	11
2.2 MOTIVATIONS.....	11
2.3 OBJECTIVES .....	11
2.3.1 <i>Link of the Deliverable 5.3 to other TIMBUS deliverables</i> .....	12
<b>3 ARCHIVAL TECHNOLOGIES AND ARCHITECTURES .....</b>	<b>13</b>
3.1 BACKGROUND .....	13
3.2 INFRASTRUCTURES AND ARCHITECTURES.....	14
3.3 BUILDING THE INFRASTRUCTURE FOR IT BUSINESS PROCESSES.....	15
3.4 INTRODUCTION TO HARDWARE COMPONENTS .....	16
3.4.1 <i>Physical Disk</i> .....	18
3.4.2 <i>Tape</i> .....	23
3.4.3 <i>Tape versus Disk</i> .....	26
3.5 CACHE .....	27
3.6 DISK ARRAYS & REDUNDANCY .....	28
3.6.1 <i>RAID – Redundant Array of Inexpensive Disks</i> .....	28
3.7 LOGICAL STORAGE STRUCTURES.....	29
3.7.1 <i>Logical Storage Structure Overview</i> .....	30
3.8 FILE SYSTEMS.....	32
3.8.1 <i>Unix/Linux File Systems</i> .....	32
3.8.2 <i>File Sharing Protocols</i> .....	34
3.9 STORAGE INTERFACES.....	36
3.9.1 <i>Storage Interface and Connectivity Overview</i> .....	36
3.10 BACKUP AND RECOVERY (BAR) .....	42
3.10.1 <i>Backup and Recovery Architecture</i> .....	43
3.11 CONNECTIVITY & REDUNDANCY .....	45
3.11.1 <i>Fibre Channel Fabric</i> .....	46

D5.3_M24_Architecture_For_VSRI	Dissemination Level: Public	Page 3
--------------------------------	-----------------------------	--------

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

3.11.2	<i>Zoning</i> .....	47
3.11.3	<i>Other forms of Redundancy &amp; High Availability</i> .....	47
3.11.4	<i>Securing the Archival Environment</i> .....	49
3.12	DATA DE-DUPLICATION .....	51
3.13	SOFTWARE COMPONENTS .....	55
3.13.1	<i>Archival Methodologies</i> .....	55
3.13.2	<i>Storage Hierarchy</i> .....	58
3.13.3	<i>Storage Abstraction</i> .....	59
3.13.4	<i>Data Backup and Restore Methodologies</i> .....	61
3.14	ALGORITHMS FOR DATA STORE AND RETRIEVAL .....	66
3.15	STORAGE MODELS.....	67
3.15.1	<i>SNIA Shared Storage Model</i> .....	68
3.15.2	<i>Storage Model Architectures</i> .....	70
3.16	SUMMARY .....	79
<b>4</b>	<b>SYSTEM REQUIREMENTS .....</b>	<b>80</b>
4.1	INFRASTRUCTURE CONSIDERATIONS.....	80
4.2	RISKS TO DIGITAL PRESERVATION .....	81
4.2.1	<i>Replication and Data Integrity (Fixity)</i> .....	82
4.2.2	<i>Migration of Media</i> .....	82
4.2.3	<i>Migration of Technology</i> .....	82
4.2.4	<i>Emulation</i> .....	83
4.2.5	<i>Sustainability</i> .....	83
4.2.6	<i>Redundancy and Heterogeneity</i> .....	83
4.3	FUNCTIONAL REQUIREMENTS .....	83
4.3.1	<i>Physical Storage Management</i> .....	83
4.3.2	<i>Archive Capacity &amp; Performance</i> .....	85
4.3.3	<i>Archive Security</i> .....	88
4.3.4	<i>Archive Protection</i> .....	91
4.3.5	<i>Archive Supportability</i> .....	92
<b>5</b>	<b>INFRASTRUCTURE ARCHITECTURE FOR BUSINESS PROCESS PRESERVATION.....</b>	<b>95</b>
5.1	THE BUSINESS PROCESS HARDWARE ARCHITECTURE FLOW.....	95
5.1.1	<i>Business Process</i> .....	95
5.1.2	<i>Primary Storage</i> .....	96
5.1.3	<i>Secondary Storage</i> .....	96
5.1.4	<i>Network and Storage Interface</i> .....	96
5.1.5	<i>Backup Meta-Database &amp; Backup Hardware Infrastructure</i> .....	97
5.1.6	<i>Business Process Archive Architecture</i> .....	97
5.1.7	<i>Redeployment Architecture</i> .....	98

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

5.1.8	<i>Storage Abstraction Layer</i> .....	99
5.1.9	<i>Offsite Protection</i> .....	100
5.1.10	<i>Relationship of D5.3 Architecture to D5.5 Architecture</i> .....	102
<b>6</b>	<b>OUTLOOK AND CONCLUSIONS</b> .....	<b>103</b>
<b>7</b>	<b>REFERENCES</b> .....	<b>105</b>
<b>APPENDIX A: SUPPLEMENTARY ARCHIVAL TECHNOLOGY MATERIAL</b> .....		<b>108</b>
A.1	FORMS OF HARD DISKS .....	108
A.2	LTFS EDITIONS .....	108
A.3	TAPE PERFORMANCE AND CAPACITY .....	110
A.4	DISK CACHING DISKS (DCD).....	111
A.5	RAID LEVELS .....	112
A.6	OBJECT-BASED STORAGE.....	114

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

## List of Figures

Figure 1: Relation of D5.3 to the rest of TIMBUS .....	12
Figure 2: Total Worldwide Archive Capacity by Media Type .....	13
Figure 3: Typical back-end infrastructure components.....	14
Figure 4: Example Infrastructure Architectures .....	15
Figure 5: Building IT Business Processes .....	16
Figure 6: What about legacy interoperability?.....	16
Figure 7: Hardware Obsolescence.....	17
Figure 8: Archive Building Block Components.....	17
Figure 9: Maximum HDD Areal Density.....	19
Figure 10: Longitudinal versus Perpendicular Recording .....	19
Figure 11: Hard Disk Track.....	20
Figure 12: Evolution of IT (Dublin October 2012).....	22
Figure 13: Storage Media of the Future <sup>11</sup> .....	23
Figure 14: Sequential versus Random access.....	24
Figure 15: Worldwide Archive Data in Petabytes.....	25
Figure 16: DRAM Cache Operation.....	28
Figure 17: Block Level Storage.....	30
Figure 18: File Level Storage.....	31
Figure 19: The UNIX File System .....	33
Figure 20: CIFS Architecture .....	34
Figure 21: NFS Architecture.....	35
Figure 22: Fibre Channel Communication System .....	37
Figure 23: Example SAN Architectures .....	38
Figure 24: USB Adapter Connectors .....	40
Figure 25: Intel’s new Thunderbolt™ Interface .....	41
Figure 26: Where were NASA’s ‘lost’ Lunar Landing Images? .....	43
Figure 27: Symantec NetBackup Architecture .....	44
Figure 28: Location of NetBackup Catalogues.....	45
Figure 29: Basic MESH Fabric Architecture Example.....	46
Figure 30: Specialisations of the basic MESH Architecture .....	46
Figure 31: High Availability.....	47
Figure 32: Fibre Fabric Dual Paths Configuration.....	49
Figure 33: Data De-duplication.....	52
Figure 34: Data De-Duplication Methodologies Example <sup>34</sup> .....	53
Figure 35: SW Components of the D5.3 Reference Architecture.....	55
Figure 36: CAS Based Storage.....	57
Figure 37: Active Archive Environment .....	58

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

Figure 38: Intel IT’s Data Storage Infrastructure .....	59
Figure 39: Data Multiplexing .....	64
Figure 40: Backup Selections Window .....	65
Figure 41: Cluster of OSD.....	67
Figure 42: SNIA Shared Storage Model v2 <sup>42</sup> .....	69
Figure 43: SNIA Shared Storage Model (for tape application) <sup>42</sup> .....	69
Figure 44: SAN Block Storage .....	70
Figure 45: NAS Storage Architecture .....	71
Figure 46: SAN Architecture .....	72
Figure 47: SDLT (Inside View with Leader tape) .....	73
Figure 48: Super DLT Tape .....	74
Figure 49: LTO Tape Bar Code <sup>44</sup> .....	74
Figure 50: Inside an Automated Tape Library .....	75
Figure 51: Tape-Based Archive Solution.....	76
Figure 52: Disk-Based Archive Solution with Data De-Duplication .....	77
Figure 53: Openstack – Archive Storage.....	78
Figure 54: Object Based Swift storage Solution .....	79
Figure 55: Scalable, Resilient archive features as defined in NEC paper <sup>46</sup> .....	81
Figure 56: Stanford University Risks to Digital Information .....	82
Figure 57: Technical Components constituting business process .....	95
Figure 58: Client/Server Storage .....	96
Figure 59: Connectivity Methods .....	97
Figure 60: Archive Protection Environment .....	97
Figure 61: The Business Process Archive .....	98
Figure 62: The Redeployment Environment.....	99
Figure 63: The Storage Abstraction Environment .....	100
Figure 64: Offsite Protection .....	101
Figure 65: LTFS Partition.....	109
Figure 66: Relationship between LTFS, the O/S and the tape hardware .....	110
Figure 67: Tape capabilities and data transfer rates.....	111
Figure 68: RAID 0 .....	112
Figure 69: RAID 1 .....	112
Figure 70: RAID 1-E .....	113
Figure 71: RAID 5 and RAID 5-E .....	113
Figure 72: Openstack Folsom Architecture .....	115

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

## List of Tables

Table 1: Securing the Archive .....	50
Table 2: De-duplication Ratios.....	54
Table 3: Retention Periods .....	64
Table 4: Flexibility .....	84
Table 5: Reliability .....	84
Table 6: Storage Interoperability.....	84
Table 7: Data Migration .....	85
Table 8: Zoning .....	85
Table 9: Performance .....	86
Table 10: Network Capacity.....	86
Table 11: Primary Storage .....	87
Table 12: Scalability .....	87
Table 13: Location Suitability .....	88
Table 14: Physical Storage Port Security .....	88
Table 15: Remote Interface Access .....	89
Table 16: Remote Access.....	89
Table 17: Authorised Access.....	90
Table 18: Secure Data Transportation.....	91
Table 19: Archive Protection .....	91
Table 20: Offsite Protection .....	92
Table 21: Operating Systems.....	92
Table 22: File Systems .....	93
Table 23: Hardware Systems .....	94
Table 24: D5.3 to D5.5 Mapping.....	102
Table 25: Openstack Cloud Components .....	116



<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

## List of Acronyms

10GBe	10 Giga Byte Ethernet
ATL	Automated Tape Library
API	Application Programming Interface
BaR	Backup and Recovery
CIFS	Common Internet File System
DAS	Direct Attached Storage
DCD	Disk Caching Disk
DP	Digital Preservation
EB	Exabyte
FCoE	Fibre Channel over Ethernet
GB	Gigabyte
GUI	Graphical User Interface
HAMR	Heat Assisted Magnetic Recording
HBA	Host Bus Adapter
HDD	Hard Disk Drive
iSCSI	Internet SCSI
I/O	Input/Output
IO Ops	Input/Output Operations
JBOD	Just a Bunch Of Disks
LAN	Local Area Network
LTFS	Linear Tape File System
LTO	Linear Tape Open
LUN	Logical Unit Number
M/s	Milliseconds
MAC	Media Access Control
MB	Megabyte
NFS	Network File System
NTFS	New Technology (NT) File System (Windows file system)
NTLM	New Technology (NT) Lan Manager
OS	Operating System
PMR	Perpendicular Magnetic Recording
RAID	Redundant Array Inexpensive Disk
RAM	Random Access Memory
ROM	Read Only Memory
RPM's	Revolutions Per Minute
SCSI	Small Computer Standard Interface
SDLT	Super Digital Linear Tape
SMB	Server Message Block (also known as CIFS)
SNIA	Storage Network Industry Architecture group
SSD	Solid State Drive
TB	Terabyte
UDP	User Datagram Protocol
UFS	Unix File System
WP	Work Package
ZB	Zettabyte

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

## 1 Executive Summary

The data stored in the digital universe is continuing to grow rapidly. According to the World Internet Statistics group, internet penetration remains at just 34% worldwide so there is ample reason to believe that the growth in data volumes can continue, at least for several more years. As it does so, the amount of data being produced in the world is rapidly increasing. IBM's big data website<sup>1</sup> has said that an incredible 90% of the data in the world today was created in the last 2 years, which is essentially the period since TIMBUS began. While it is true that not all of this data needs to be preserved, it is reasonable to assume that the percentage of it which does is also experiencing rapid growth, a fact backed up by the digital archive section of TIMBUS deliverable D2.2. While storage costs have never been cheaper than at present, it is also true that the volume of data requiring storage has also never been larger so the decreases in the cost per gigabyte of storage are being rapidly offset by the equally rapid growth in the need for storage capacity. Kryder's law appears to be broken<sup>2</sup> and regardless of the 2011 floods in Thailand which affected hard disk prices worldwide, the assumption that disk costs will fall to offset the demand for storage may no longer be valid.

Against this backdrop, the primary motivators for deliverable D5.3 are:

- Address the gap in knowledge in the digital preservation community as identified by deliverable D3.3 surrounding the technical storage technologies. This deliverable therefore goes into detail to provide a reader with ample material covering the basic storage concepts and technology developments which they may not be aware of and which are relevant to long-term storage solutions.
- The primary goal of the deliverable is to present the TIMBUS long-term preservation architecture which is required to support the tools under development in WP6 and thus aid with preserving and re-deploying business processes.
- The storage industry is still evolving. This deliverable covers current developments and presents an architecture based on that. The architecture is therefore based on industry best practices today. As there are no further deliverables relating to an updated version of the infrastructure architecture, the TIMBUS partners have taken the decision that the D5.3 architecture will continue to be evolved through-out the remaining time of TIMBUS as part of Task 6.7. In this way, task 6.7 will become a feedback loop where advances in the storage industry, or knowledge gained from the implementation and operation of the architecture will have the opportunity to be documented and disseminated. In addition to being pragmatic, this approach will also allow any reviewer feedback in year 2 to be addressed by task 6.7 in year 3.

This deliverable presents the TIMBUS infrastructure reference architecture for long term archives.

<sup>1</sup> <http://www-01.ibm.com/software/data/bigdata/>

<sup>2</sup> <http://blog.dshr.org/2012/10/storage-will-be-lot-less-free-than-it.html>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

## 2 Introduction

### 2.1 Background

As our online world grows each day so does the amount of data we produce. Virgin Atlantic's IT director David Bullman recently stated that their new 787 fleet will create a half terabyte of data for each flight<sup>3</sup>. The majority of these will stem from processes running on board the aircraft during inflight.

In a study conducted by IDC in 2010, digital archive growth will reach data amounts of 35 ZB (Zettabytes) by 2020<sup>4</sup>. We have a real challenge on our hands to distinctively meet our wider objectives which are to provide short term backup and restore, long to medium term preservation and disaster recovery while providing an infrastructure to preserve and store business processes and supporting software and technology stack.

Meeting our objectives requires storage subsystems that can support software processes which exhumate environments to be redeployed at any point in time in the future. In doing so we must consider the reliability of this infrastructure and the transparency of the hardware to enable long term preservation.

Many would argue that we are already doing this today. With reference to TIMBUS deliverable D2.1 many storage vendors today offer sustainable solutions for long-term storage of digital content such as documents, images, video, audio files. The focus of TIMBUS is to preserve the software environments which constitute business processes. This will require us to preserve metadata describing the hardware, software stack and the software environmental context.

### 2.2 Motivations

We understand that providing a digital archive for long term retention is a process routinely performed today within many organisations with many vendor based solutions in place. The primary motivator for Task 5.3 is to provide a comprehensive storage architecture which addresses the hardware and software requirements for the long-term preservation of complex digital objects such as different data types, entire business processes and the execution environment they run in. This was identified as an area where more knowledge is needed by the TIMBUS training requirements. In addition, to offer this solution along-side static archives to complement the redeployment of business processes.

### 2.3 Objectives

The main objectives of task 5.3 is to define an archival storage architecture to support a digitally preserved environment which is driven by the requirements and use case scenarios defined in work package 7, 8 and 9. The storage architecture is also aligned to the other TIMBUS deliverables, particularly D5.5 and D6.4.

The main objectives of Task 5.3 (M24) are:

- a) Identify and state the technical architecture and infrastructure components required to preserve a business process.

<sup>3</sup> [http://www.networkworld.com/news/2013/030713-boeing-787s-to-create-half-267470.html?source=nww\\_rss](http://www.networkworld.com/news/2013/030713-boeing-787s-to-create-half-267470.html?source=nww_rss)

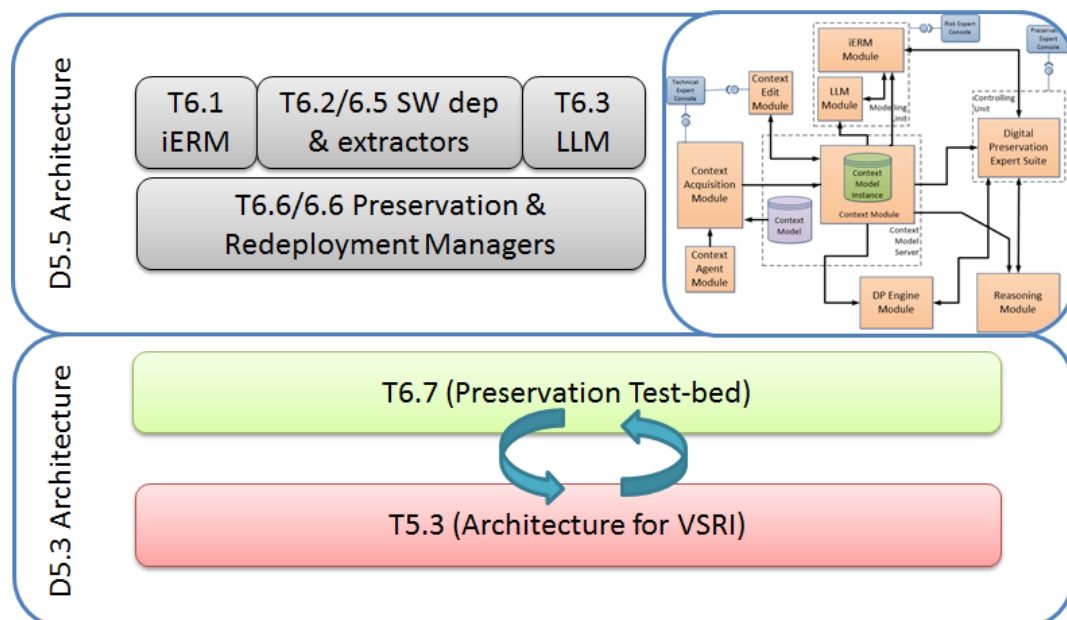
<sup>4</sup> <http://www.insic.org/news/A&S%20Roadmap.pdf>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

- b) Identify and state the technical architecture and infrastructure components required to re-deploy a business process.
- c) Identify and state the technical architecture and infrastructure components required to migrate a business process which has been preserved.
- d) Algorithms to uniquely identify the components contained in a stored business process and the level of exhumation that is deemed to be acceptable.
- e) Meta-data to describe the underlying data components that constitute a digitally preserved business process.
- f) Meta-data to describe the underlying hardware and software components required to support an exhumed environment.
- g) Storage mechanisms that allow the exhumation of the required components in a hardware & software neutral manner.

### 2.3.1 Link of the Deliverable 5.3 to other TIMBUS deliverables

Figure 1 below shows the high-level relationship of the work carried out in D5.3 with the other parts of TIMBUS. The software architecture components as defined in the D5.5 architecture and under development in work-package 6 sit at the top of the stack. It is the intention of the consortium to continue the development of the infrastructure architecture, and the test-bed, throughout the project lifecycle and to report this out as part of T6.7. This will be done wherever deemed necessary, for example to keep the project informed about external market advances in technology or best practice or to implement optimisations or new knowledge learned as TIMBUS continues on its execution lifecycle.



**Figure 1: Relation of D5.3 to the rest of TIMBUS**

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

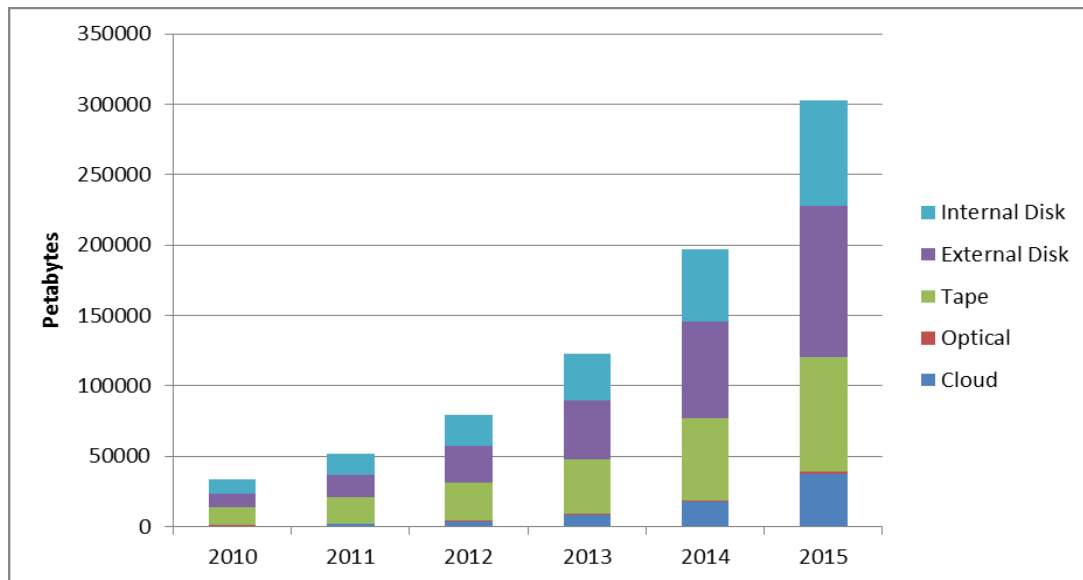
### 3 Archival Technologies and Architectures

This section of the deliverable will go in to some detail on the technologies used to implement archives and highlight their relevance to TIMBUS. The purpose of the material is to provide a technical context for the work in D5.3 and establish state-of-the-art in this field.

#### 3.1 Background

Every compute environment relies on a form of primary or main storage to write, house and read data which has been housed there by a business process running on a front end hosted compute environment such as a personal computer or server. The only known way to keep this data is to digitally archive to another location typically hosted on a storage frame or tape based solution. Digital archiving has become a high focus area for many IT planners and IT departments mainly due to the huge growth in data. In many ways this situation is contradictory as we attempt to control the growth of data within our organisations while ensuring we can maintain availability of this data for long term retention and re-use. To add to this already complex challenge many organisations face the prospect of maintaining this necessary data on high end expensive infrastructure which leads to high costs of ownership for IT There is a real risk of being consumed by a digital avalanche if the right tools are not developed and implemented to manage our digital archives.

As shown in Figure 2, there is growth across all media types in archival hence, opportunities for many technologies and approaches to play a role in a tiered archive implementation such as disk, internal or external, tape, removable disk, optical, and cloud resources.



**Figure 2: Total Worldwide Archive Capacity by Media Type** <sup>5</sup>

<sup>5</sup> ESG Report, 2010, "Total Worldwide Archive Capacity by Media Type" 2010.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

According to Mark Peters from ESG *Enterprise Strategy Group*, the amount of data stored on tape continues to surge as massive and growing digital archive demands compel users to seek practical and economic ways to address escalating data volumes. Peter expects worldwide tape archive capacity to grow more than six fold over the next five years from just under 13 Exabyte's in 2010 to over 81 Exabyte's by 2015, organisations need smart strategies and tools that deliver extreme storage efficiencies while meeting stringent data reliability and availability requirements.

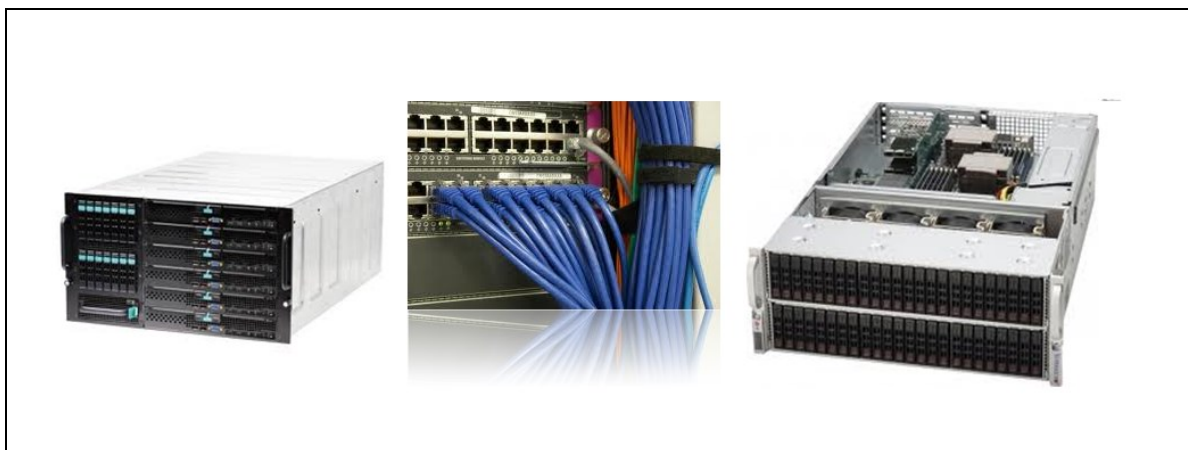
In a survey completed by NDSA (National Digital Stewardship Alliance)

- **87%** of respondents are responsible for their content for an indefinite period of time.
- **64%** of respondents are planning to make significant changes in the technologies in their preservation storage architecture in the next three years.
- **74%** of respondents report a strong preference to host and control their own technical infrastructure.
- **51%** of respondents are considering or already using a cloud storage provider to keep one copy of their content.

There are two main components of a digital preservation environment, the hardware elements and the software elements. The hardware components are the physical building blocks of the archive architecture. The software components allow us to manipulate the data, protect the data and access it when required.

### 3.2 Infrastructures and Architectures

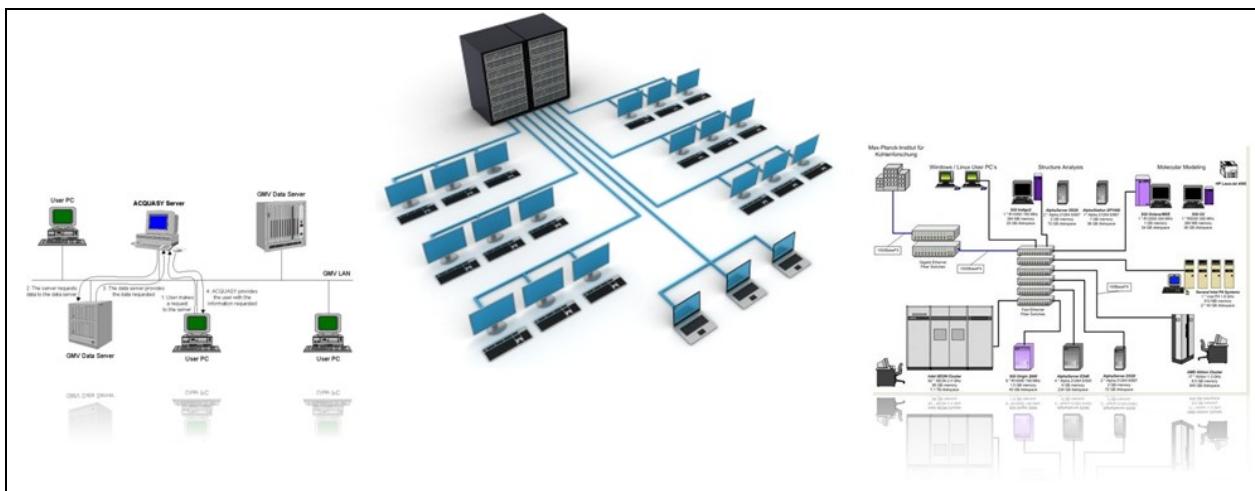
Infrastructure is defined as the constituent hardware components that are used to run IT process upon. Figure 3 below shows some typical components ranging from blade servers on the left, to network switches in the middle to storage servers on the right. Each infrastructure component is capable of performing a specific or sometimes a specialist task.



**Figure 3: Typical back-end infrastructure components**

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

Infrastructure components can be thought of as resources which provide capabilities. They are simply building blocks which allow users to provide the underlying compute, storage and network that may be required by a particular business process. On their own, individual infrastructure components can solve basic problems such as the immediate need to provision an application to an organisation. It is usually necessary to implement an infrastructure architecture with all but the most basic of IT systems. Figure 4 below shows some example infrastructure architectures. Infrastructure architectures are the result of a design process whereby components and resources are configured and interconnected in a specific way to solve a particular challenge, for example to scale a business process so that it can support thousands of users or to provide redundancy in the event of hardware failure.



**Figure 4: Example Infrastructure Architectures**

### 3.3 Building the infrastructure for IT Business Processes

As depicted by Figure 5, IT departments respond to the needs of businesses by purchasing infrastructure components, designing the underlying infrastructure architecture and deploying the suite of applications required by the business to perform a necessary task or to respond to a business opportunity. In many cases, the software developers, the application support and the infrastructure engineers work together to find the most cost effective solution which will deliver what is needed in an expedient manner. The emphasis is on meeting the immediate business need. Some emphasis is usually given to disaster recovery and typically the system will be connected to an existing Backup and Recovery (BaR) environment or one will be designed and deployed for it. Very little, if any, emphasis today is given to the long-term consequences to the business in the event that the data contained within that business process is not available in the future. The TIMBUS project is seeking to raise this awareness and deliverable D5.3 focuses on educating the reader about the considerations involved in the underlying hardware components before presenting the TIMBUS reference architecture for business process preservation and redeployment in section 5. To fully understand the infrastructure architecture proposed by deliverable D5.3, section 3 of this deliverable must detail the infrastructure components. Long-term archival considerations are discussed across components, starting with the physical media and building the layers on top of this such as caching, disk redundancy, file systems, storage models and interfaces and archival software services (such as data de-duplication). All of these have been taken in to consideration in the design of the TIMBUS infrastructure architecture.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Building IT Systems.. Everything is a choice..**

TIMELESS BUSINESS

“ Which server platforms will best meet my needs? “

“ How do I architect my IT system? ”

Is success simply any combination which meets the requirements and performs well enough to keep our users happy?  
..shouldn't we expect more?

timbusproject.net © 2013

**Figure 5: Building IT Business Processes**

### 3.4 Introduction to Hardware Components

When presented in hardware terms, the issues that TIMBUS is trying to solve are caused by legacy interoperability. As depicted in Figure 6 from the TIMBUS training day in Dublin in October 2012 evolution seems to be a natural part of our world. In nature this happens at a barely perceptible rate. In the IT world though, the pace of change is phenomenal and this is what motivates the TIMBUS challenge.

... *rapid* IT evolution causes problems

Not likely to happen !!

timbusproject.net © 2013

**Figure 6: What about legacy interoperability?**



<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

If we look at some of the common storage media readily available today, as shown in Figure 7, most people will rapidly recognise the scale of the problem. We often don't pause to consider how heavily we depend on the hardware interfaces, and the storage media which store our data. As we enter 2013, newer and smaller devices continue to be developed with the higher storage capacities that we demand as consumers. But there is a long tail of legacy devices that are being rapidly pushed out of existence.

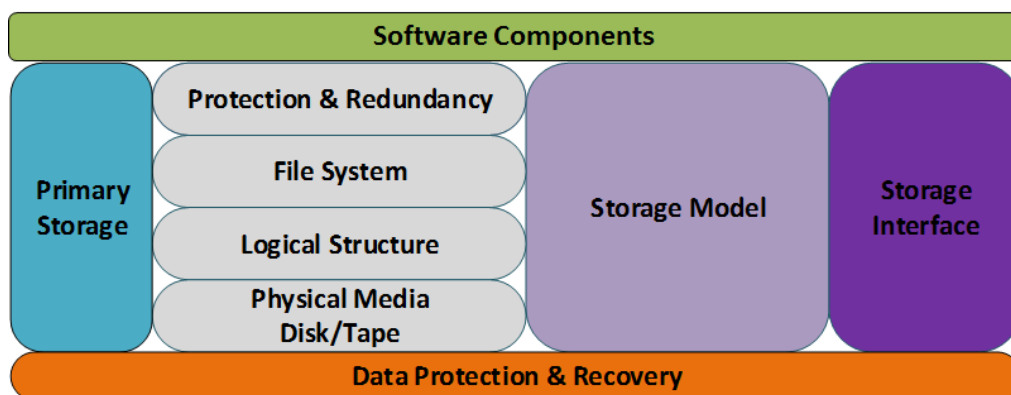
**Storage media today.. the not-so-obvious problems**

TIMELESS BUSINESS ⏪ ⏩

- Different types of disks, tapes, VTLs, CD/DVD, flash, etc
- Optical disks, flexible magnetic => proper handling and storage?
- Physical form factor of media and rendering device
- Software used to read old data formats is a big concern; is complex
- Both SW and media can be proprietary to not only the protocol used, but also to the vendor used.

**Figure 7: Hardware Obsolescence**

Figure 4 previously illustrated some general examples of infrastructure architectures. In the introduction to the hardware components used in archival systems, it is a good idea to have an overall picture of how they fit together into a typical archive-specific architecture. Figure 8 below depicts the building blocks of the hardware components involved in such an archival architecture. The remaining sections of chapter 3 of this deliverable will examine these components, starting with physical media and redundancy, then moving to logical storage structures, file systems, storage models and interfaces and finishing up with software components.



**Figure 8: Archive Building Block Components**

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

The primary block, and the starting point for the work in D5.3, centres on the physical media. When data is passed to the storage sub system, it will initially pass through the primary storage environment and then onto disk. Depending on the archive methodology this starting block will either be a disk based solution or a tape based solution.

### 3.4.1 Physical Disk

The most fundamental, and common, storage media in use today for archival are disk and tape. A hard disk is a very common source of storage. It is usually abbreviated as HDD (Hard Disk Drive) and they are common in many compute devices today such as a PC, laptop, tablet, mp3 player and the list continues. It is also worth stating that users rely heavily on hard disks for their digital archive and this is a major component of the hardware storage model.

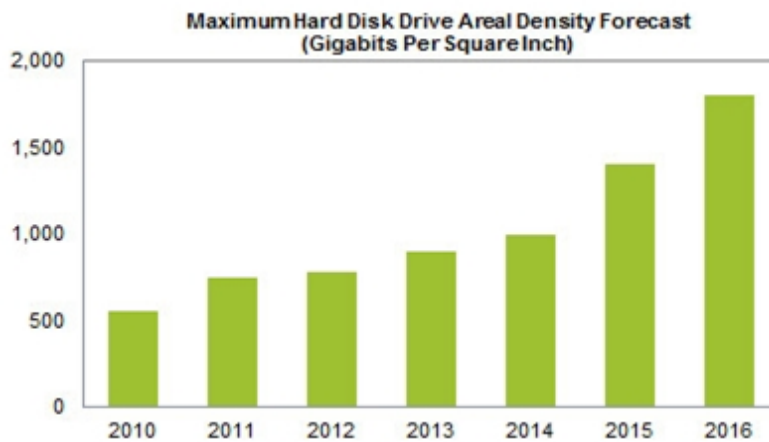
A hard disk is simply a magnetic disk produced from metal and covered with a magnetic surface used to record on. Some hard disks can be easily removed from a device or they may be permanently secured in the device. Every media type has its advantages and disadvantages and hard disks are no different. The problematic nature of hard disks, which this document will discuss, is an important consideration when designing a long-term infrastructure archival architecture. Some systems have the ability to easily remove a hard drive from a system and this is an important benefit as you can recreate the lost data, assuming that you have data redundancy enabled on your disks. Disks can also be a single point of failure if no data redundancy is enabled which means disk can be a high risk element if no other form of data redundancy or data protection is enabled. Eliminating single points of failure is an important focus in architecting any IT system, but particularly so in the case of a long-term archive. Disks are highly vulnerable to *head crash* which is a physical destruction of a hard disk. Misalignment or contamination with dust can cause the read/write head to collide with the disk's recording surface. The head crash destroys all or part of data on the disk.

Hard disk capacity plays a major role in the cost and way we manage the masses of data we are collecting and retaining. The market analysis in deliverable 2.2 relates the huge growth that is happening in data volumes. It is generally deemed that we take for granted the fact that our hard disks will continue to grow in capacity every year to meet that demand. The diagram below illustrates the growth since 2010 and predicts by 2016 we will see HDD's with a capacity of up to 2000 GB per square inch. This is a far reach from the first hard disks which stored 4.4MB across fifty 24 inch disks. The physical form-factors of hard disks today are 2.5 inch and 3.5 inch. Those form factors today enable typical hard disk capacities that range from a conservative 500GB to as much as 4TB in size. But of course, there is no guarantee that the form factor we commonly recognise today will be around in the future and as we have pointed out, today's 2.5 and 3.5 inch disks are a lot different to the 24 inch disks which first appeared when this technology was created.

Bearing that in mind, it would not be possible to undertake the work in D5.3 without considering what the future might offer in relation to storage technologies (covered in section 0) and HDD capacity (covered here). It is clear that the current magnetic disk cannot sustain the growth levels beyond 1.5TB per square inch as the bits stored become unstable even with small heat fluctuations. This is known as bit flipping. A new technology called HAMR which stands for Heat Assisted Magnetic Recording involves the heating of very small areas to enable data to be written and is more stable at normal temperatures. HAMR could ena-

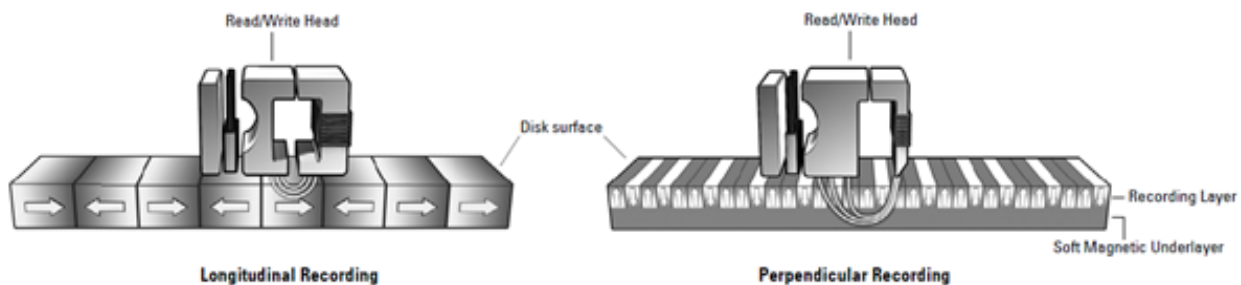
<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

ble up to 60TB drives on the 3.5 inch standard desktop hard disk<sup>6</sup>. It is expected that we could see commercial implementations of HAMR disk drives by 2015.



**Figure 9: Maximum HDD Areal Density<sup>7</sup>**

Another alternative to HAMR which is deemed to be prohibitively expensive is PMR, which stands for perpendicular magnetic recording. Figure 10 below illustrates how PMR works. PMR is an evolution from the historical longitudinal magnetic recording method. PMR differs from longitudinal as when the disk is being magnetised with bits they are aligned vertically as opposed to horizontally thus enabling more data to be stored on the disk.



**Figure 10: Longitudinal versus Perpendicular Recording<sup>8</sup>**

### 3.4.1.1 Disk Performance

A key indicator when looking at hard disks is the capacity and speed at which the drive can write at. There are four main elements to consider when we discuss disk performance. These are:

- Controller overhead.
- Seek time.

<sup>6</sup> <http://www.insic.org/news/A&S%20Roadmap.pdf>

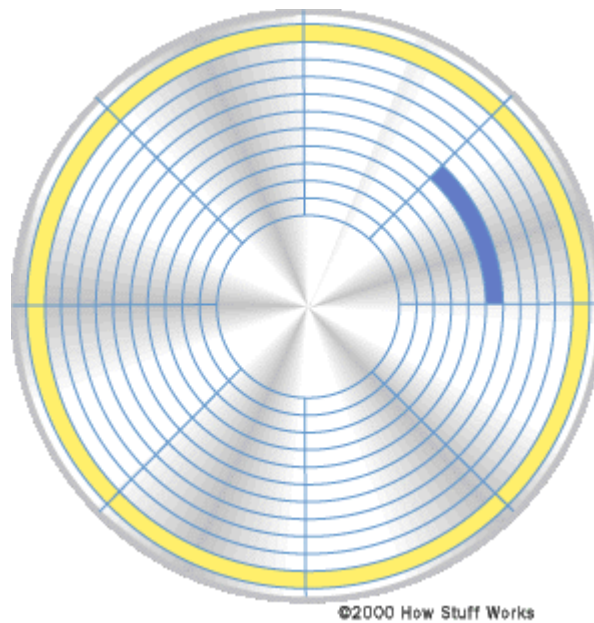
<sup>7</sup> HIS iSuppli Research, May 2012

<sup>8</sup> <http://www.wdc.com/wdproducts/library/other/2579-701185.pdf>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

- Rotational latency.
- Data read/write time.

Capacity plays a part in disk performance as the larger the disk, the longer the seek time will be. This can be compensated for by spinning the disk media at higher speeds. To help understand this, Figure 11 below illustrates how the disks surface is organised into tracks and sectors which are stored on a platter. The tracks are identified as concentric circles while the sectors are pie shaped. Below you can see that a track is marked in yellow while the sector is marked in blue. When you perform a format on a disk the tracks and sectors are created.



**Figure 11: Hard Disk Track<sup>9</sup>**

**Controller overhead** relates to when a read/write is received on a controller the time spent relates to the controller overhead which is typically 1~2 m/s.

**Seek time** refers to the moving of the read/write head.

**Rotational Latency** refers to when the read/write head is in position it is then placed on the track and then waits for the target sector to rotate. A typical rotation is 7200 RPM's.

**Read/Write Time** relates to data transfer between the disk head and disk media.

The reason these are important in the TIMBUS architecture is that form factors, capacity, seek time and performance all feed into cost. Small form factor disks (2.5 inch) with high capacity and performance are the most expensive. 3.5 inch disks have the highest available capacities and are generally cheaper than their 2.5 inch counterparts, but they are larger in size. When implementing an archival infrastructure for long-term business process preservation these tradeoffs can be used to provide different “tiers” of storage

<sup>9</sup> <http://static.ddmcdn.com/gif/hard-disk-track.gif>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

where data which is less frequently accessed can be stored more cost effectively. Data tiering is explained later in section 3.13.2 of this deliverable.

### 3.4.1.2 SSD - Solid State Drives

Solid state drives are commonly discussed today as being at the leading edge of hard drive technology and this section considers some aspects of their performance which relate to their suitability to long-term archival. Unlike the electromechanical types of hard drives, the SSD drive has no moving parts and utilises NAND flash memory as its storage element. According to the industry solid state drive technology provides faster system performance than magnetic media such as what we have discussed so far in this chapter. As there are no moving parts the risks of failure are dramatically reduced in normal scenarios such as when they are used in laptops or tablet devices. They are deemed to be greener as they use much less power than magnetic drives and are more responsive making them more efficient<sup>10</sup>.

Storage performance is usually measured in input/output per second or IOPS. The greater the IOPS the better your application will perform. Many SSD designs use the parallel ATA or PATA design. Some of the newer designs incorporate SATA which is serial interface which can allow for up to 250Mbps of data. For example Intel's SSD which incorporates native SATA with 10 parallel NAND flash channels claims to fully utilise the bandwidth in addition to using NCQ which stands for native command queuing which allows up to 32 command operations.

In essence the SSD has some key advantages over traditional hard drives as they offer much faster access to your data. You do not have to wait for head seeking or spinning to occur. SSD enjoys a much faster random access performance. SSD's are low risk compared to the mechanical hard drive with moving parts. An SSD consumes 20% less power than an electromagnetic hard drive. Another important statistic is mean time between failures (MTBF). An SSD has an MTBF of 1.2 million hours compared to that of a HDD with an MTBF of 600 hours.

SSD's are really having an impact when it comes to the growth of virtualised apps, increased demand for content, the huge growth in data repository and the demand for real time information.

Clearly disk is at a critical stage in its life cycle, next generations will need to provide higher capacity and increased performance to match the requirements needed from customers. The major disadvantage of SSD's is that their cost per GB of storage is much higher than that of traditional magnetic disks and this may deter many from choosing SSD's for archival purposes along with the fact that archives by their very nature do not require the highest levels of IO performance. As the technology matures though, there is no reason to think that they won't become competitive in price to other storage media and if that happens, more organisations will consider SSD usage for their archive infrastructures.

The option of Solid State Drives offer archives a faster and more efficient disk based solutions compared to other disk based formats. More information relating to forms of hard disks such as internal and externally attached disks can be found in Appendix A.1 Forms of Hard Disks.

---

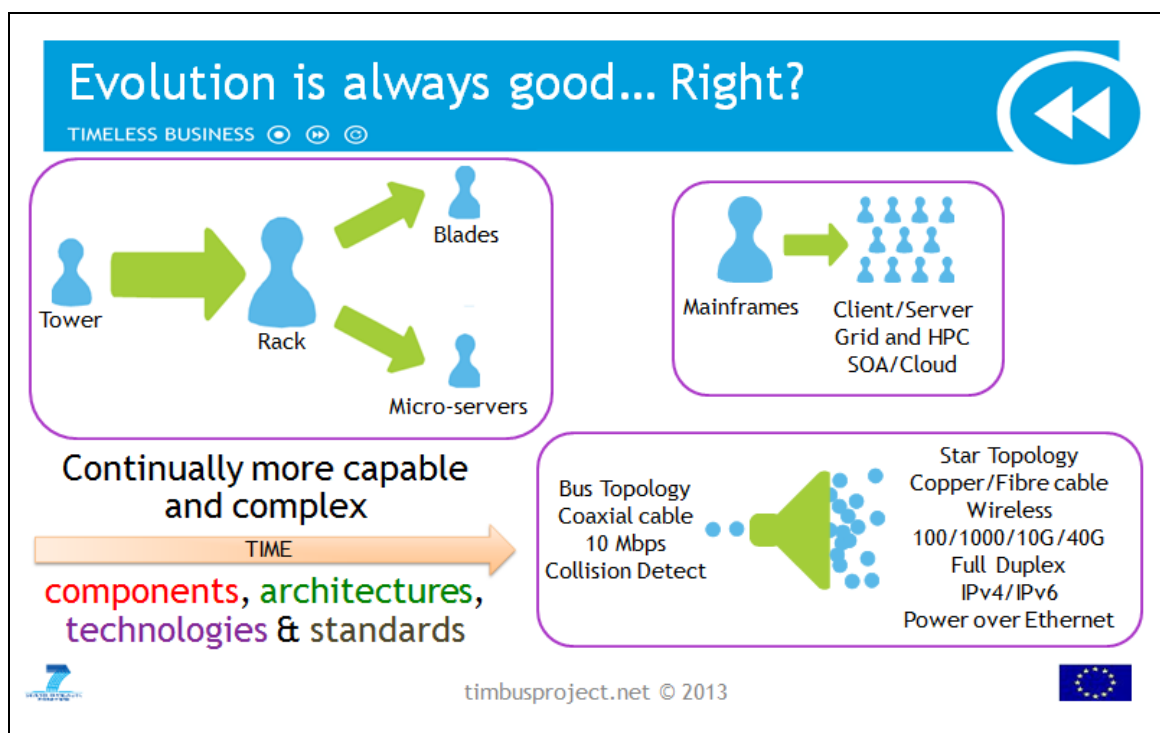
<sup>10</sup> [http://www.imation.com/Global/enUS/Scalable%20Storage/Products/InfiniVault%20Storage%20Appliances/ESG\\_Marke\\_Report\\_Logic%20and%20Value%20of%20a%20Tiered%20Archive.pdf](http://www.imation.com/Global/enUS/Scalable%20Storage/Products/InfiniVault%20Storage%20Appliances/ESG_Marke_Report_Logic%20and%20Value%20of%20a%20Tiered%20Archive.pdf)

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

### 3.4.1.3 Future Disk Drives

This section of the deliverable considers what the future may hold in terms of disk technologies. This is an important consideration when designing the archive infrastructure in D5.3.

Figure 12 below is a slide presented by Intel at the TIMBUS training event held in Dublin in October 2012. It is relating the rapid evolution of technologies in several vectors, namely form factors (tower, rack, blades, etc.), architectures (mainframe, client-server, HPC, etc.) and in technologies and standards. Some times these changes are disruptive and replace the previous technology almost completely; an example of this might be the replacement of collision-detect networking with full duplex. In other cases, there may be a long period of co-existence; a good example of this is IPv4 and IPv6.



**Figure 12: Evolution of IT (Dublin October 2012)<sup>11</sup>**

The problems that TIMBUS is trying to address have their roots in the fact that it is the rapid pace of technological change which causes IT hardware, software and file formats to rapidly become legacy. While we cannot be certain what the future will bring, we can at least extrapolate from the past that rapid IT evolution is going to continue, at least for the short to medium term future. Bearing this in mind, another of the slides presented at the TIMBUS training event in Dublin is shown in Figure 13. It takes a brief look at some of the possible storage media of the future. These range from research prototypes such as DNA disks which may or may not turn out to be viable and cost effective as commercial products to ones such as Helium, Holographic and Quartz/Glass based media which seem to be much more near term.

<sup>11</sup> <http://www.youtube.com/watch?v=ulbmURmkv6w>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Storage media tomorrow.. who knows?**

TIMELESS BUSINESS

- Helium HDDs (2012-13?)**  
 40% capacity increase, 23% less power. 3.5 inch form factor, 10 yrs of development by Western Digital subsidiary HGST
- Holographic Storage**  
 1TB on a CD/DVD media format
- Genetic hard drives** (Harvard)  
*" Scientists have found a way to store an entire textbook in the code of DNA -> billions of GB's per gram of media "*  
 => cost of the DNA sequencer and other instruments  
*" currently makes this impractical for general use "*
- Permanent quartz glass (2015?)**  
 Prototype developed by Hitachi  
 Form microscopic layers of dots inside quartz to serve as binary dots  
 40MB per square inch (equivalent to CD)  
 Withstands high temperature, water, etc.  
 Can be read by human eye with optical microscope  
 Potential lifetime of several hundred million years (assuming the concept of binary code is still understood then!)

timbusproject.net © 2013

**Figure 13: Storage Media of the Future** <sup>11</sup>

The slide is very interesting because it makes individuals concerned with long term preservation consider that their existing storage media (primarily magnetic disks, solid-state disks or tape) could become legacy if any of the technologies in Figure 13 become disruptive. The consequence of this is that long-term archive architectures need to be designed based on the assumption that the media will change. To this end, D5.3 proposes a storage abstraction layer to act as a POSIX interface and insulate the curation functions of the archival process from the back-end storage medium required to store the data. This is detailed in section 3.13.3 of this deliverable.

We can conclude that disk drives which are the foundation of our infrastructure systems today have many internal elements which determine how they reads and write data. Disk is consistently developing into the archive area and while there are some interesting potential technologies on the horizon, disk remains a main stream media for data storage. Disk is clearly at a cross roads in relation to its future roadmap but one thing is for sure, the industry will ensure disk becomes larger and faster which ever route it decides to take and cost per gigabyte, rather than high performance is always going to be a consideration of long-term archive infrastructures.

### 3.4.2 Tape

Tape is very widely used in archival today. It has been shifting from its historical role of providing a short term backup solution for disaster recovery to a more main stream role in data storage and active archives according to Information Storage Industry Consortium. Magnetic tape was first used to record computer data in 1951.<sup>12</sup> We are still using tape today and will continue to use tape long into the future as part of a

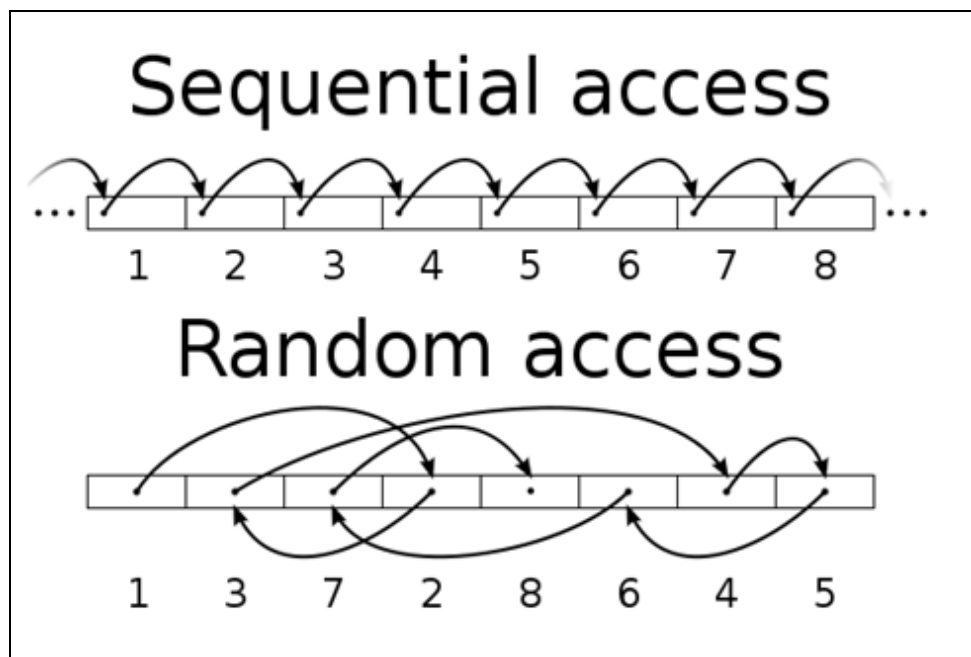
<sup>12</sup> [http://www.cratsindia.com/b/magnetic\\_tape\\_storage.pdf](http://www.cratsindia.com/b/magnetic_tape_storage.pdf)



<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

cloud based long term archival solution<sup>13</sup> and because it is such a popular archival medium this section of the deliverable covers the aspects that are needed to understand the D5.3 architecture which is detailed in section 5 of this deliverable.

Tape uses a linear approach to storing data in long parallel tracks that run the entire length of the tape. Data is written to tapes in blocks with inter-block gapes between them. Data is always appended to the end of the file and never overwrites files like disk. Tape also differs from disk as with tape, data is accessed sequentially as opposed to randomly. This has implications for the time it takes to find a file on a tape because it is not a random access media. As illustrated below in Figure 14, this is different from random access which means that data stored on the media can be accessed arbitrarily.



**Figure 14: Sequential versus Random access<sup>14</sup>**

Figure 15, below, makes a strong case for tape usage into the future showing huge growth for this media in cloud and non-cloud situations. As explained in the market assessment in TIMBUS deliverable D2.2, new players such as Amazon have begun to offer long-term archival solutions in the cloud in 2012 but these solutions only address the issues of bit-preservation. Typically tape is treated as a third line of recovery by migrating or taping out from disk after a period of time to reduce the risks associated with disk storage such as hardware failure, natural disaster and human error. Tape is also used to address cost issues, as tape is less costly than disk in terms of both initial capital expenditure and on-going operating expenditures. The low energy usage of tape makes it a cost efficient and greener solution.

<sup>13</sup> <http://www.horison.com/21stCenturyTape.pdf>

<sup>14</sup> [http://en.wikipedia.org/wiki/File:Random\\_vs\\_sequential\\_access.svg](http://en.wikipedia.org/wiki/File:Random_vs_sequential_access.svg)



TIMBUS	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

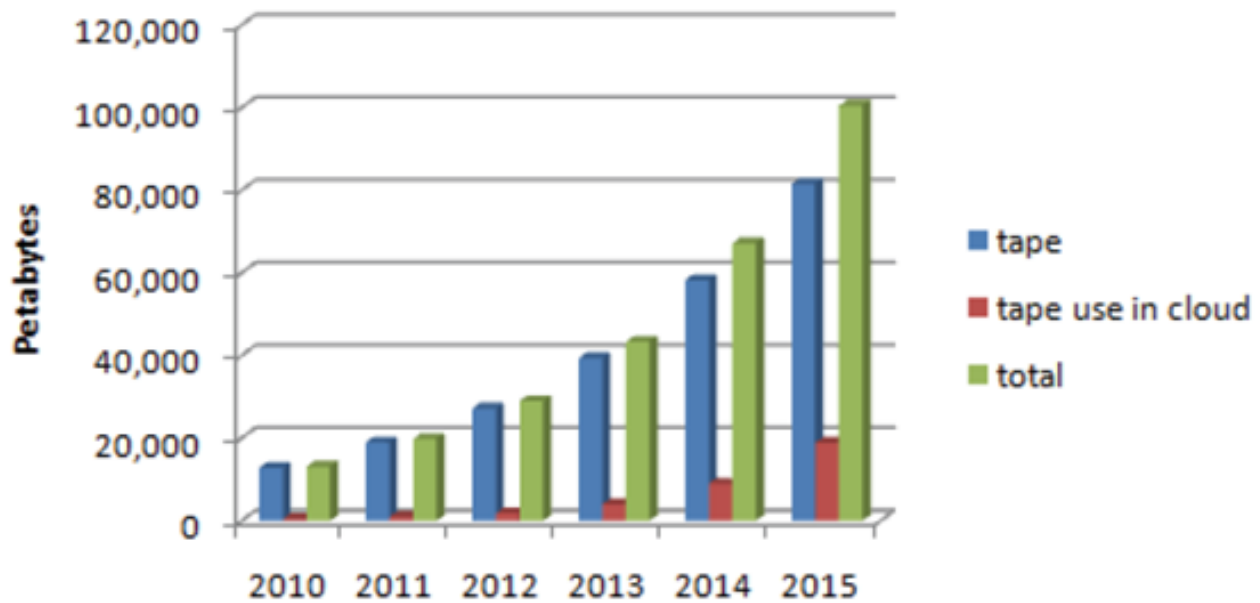


Figure 15: Worldwide Archive Data in Petabytes<sup>15</sup>

Tape technology has continued to advance impressively in the past few years, and despite forecasts that tape drive and media revenues and unit shipments may decline somewhat, tape is expected to retain a very significant share of worldwide stored data.

### 3.4.2.1 Storing data on tape

A file on tape is stored using a very basic system of numbers. This basic approach has developed over time into a more advanced file system called Linear Tape File System (LTFS) and organisations are adopting this tape based approach for their long term preservation needs. This section explains why LTFS is such an important development, especially to organisations using tape for long-term archival and not just disaster recovery.

Today there are a number of tape formats available. This fact alone creates a problem for long-term archival as these formats become legacy over time. Linear Tape-Open (LTO) format is the most common in use today. LTO drives offer a high capacity and high performance rate on media. In addition to this LTO-5 and next generation LTO-6 offer Linear Tape File System (LTFS) capability. This is highly relevant to long-term archival. LTFS was developed by IBM and released in 2010. It is a self-describing format where data in the form of files and metadata describing the files can be stored in a hierarchical directory structure.

The reason this is relevant to TIMBUS is to relate to the reader that tape formats can also become legacy but steps such as ensuring that the tape backup system uses mechanisms like LTFS can greatly enhance the survivability of data because a separate database storing all this information is no longer required as the LTFS format is self-describing. This allows an LTFS volume to be mounted and its content accessed with full use of the data without the need to access other information sources. Data can be passed between sites and applications using only the information written to an LTFS Volume. Files can be written to, and read

<sup>15</sup>[http://www.storagenewsletter.com/images/public/sites/StorageNewsletter.com/articles/icono8/insic\\_tape\\_roadmap\\_540.jpg](http://www.storagenewsletter.com/images/public/sites/StorageNewsletter.com/articles/icono8/insic_tape_roadmap_540.jpg)

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

from, an LTFS Volume using standard POSIX file operations. Older LTO formats do not have this capability and hence a separate database must be maintained to ensure that a meta-data record of what is available on that tape is available into the future. Some additional background information is included in appendix A.2 LTFS Editions.

Media load and file access times are very important. The media load time refers to the amount of time it takes a cartridge to insert and the drive to mark itself ready. In section 0 we discussed how solid state drives SSD have increased performance of hard disks. A similar story has been unfolding with tape. Linear Tape File System LTFS, some would argue has brought tape back from the dead. File access time is the time between the point when the drive receives a read command on a file and when the drive actually begins the read activity. LTO-5 offers a load time of 12 seconds and an average access time of 62 seconds. This is an average as the file could be in the middle or the end of the tape. These seek-times are good enough to allow an end user to access files on a tape as if it were a disk device. Additionally, LTFS enables drag and drop actions from tape to disk which was not previously possible. When these capabilities are combined, it means end users can run any application designed for disk files against tape without concern for the fact that the data is physically stored on tape. More information on tape is available in appendix A.3 Tape Performance and Capacity.

The view that LTFS has brought tape back from dead is entirely debatable as tape has many other qualities which would have ensured its survival into the future. Some facts about tape need to be stated. While tape has been around for over 60 years, today's tape is much improved. Consider that the first LTO generation, LTO-1, held just 100GB of data and had a maximum throughput of 20MB/sec. Today's latest LTO-5 generation operates at 140MB/sec and holds 1.5 TB uncompressed, a speed unsurpassed by all but the fastest and expensive enterprise hard disk drives. The next generation of LTO -6 will provide even greater capacity and throughput and will also be a major element in the Linear Tape File System solution.

The other strong point of tape is that tape is built for backups and archives. It is not designed to be randomly accessed or with a focus on high speed as is the case with hard drives but rather its design goals are longevity and reliability. LTO has a bit error rate of about  $10^{16}$ . This means that the un-correctable bit error rate of tape is 10,000 times lower than disk and an order of magnitude higher than enterprise SAS/FC disk, and two higher than SATA. Tape drives also read just after they write to prove it was written correctly. Finally, the magnetic medium on which data is stored is more stable than disk.

### 3.4.3 Tape versus Disk

There are many discussions on-going in the archival industry surrounding tape and disk. It is a fair assessment that much of this debate is one sided and is designed to promote one side over the other.

As we stated before, tape was designed for backups and archive. Disk has many other functions. Tape is now moving into these areas. What is true is both disk and tape has moved extensively into the data protection and archiving area. Both disk and tape have a strong existence in the long term preservation area.

Vendors on both sides of the fence will play up the faults of each medium. Tape has some challenges to overcome when compared directly to disk, a notable one being the wear and tear on a tape device called *shoe shining*. This occurs as tape writes data in blocks, once the flow of data slows or stops so does the writing of data and the tape drive stops, then starts, then stops again. Ideally the tape drive would be writing continuously. Getting the data to the tape as quickly as the tape can digest it is a real challenge. To

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

overcome this various methods have been used alone and in combination to cope with this difference. A large memory buffer can be used to queue the data. The host can assist this process by choosing appropriate block sizes to send to the tape drive. There is a complex trade-off between block size, the size of the data buffer in the record/playback deck, the percentage of tape lost on inter-block gaps, and read/write throughput.

The reality is both mediums play an important role and complement each other. A prime example of this is multi-streaming data from disk to a tape to improve backup times and better use resources. Disk staging is used by Familysearch.org to manage their backup loads. Many organisations today use both to handle current and aging data.

By updating the perspectives on tape and disk, organisations are free to use disk and tape to their best advantage. Disk gets to data fast. Tape stores data securely, for a long interval, and cost effectively. New technologies make it easy to access any data in a tape library. We use disk for data the organisation frequently uses. For cases as disparate as e-discovery, rapid data retrieval and online structured data, disk is hands-down the best choice.

You can cut costs using de-duplication technology, and migrate data efficiently to less expensive storage using the right storage application. Use tape for less used data. You can still access the data through active archive applications and file system front end applications like LTFS outlined in the above section. Use tape for data that has long retention periods, and always for disaster recovery and organisational continuity. It is the most secure and long-lasting storage media available, period. Disk and tape both have a big role in long term preservation. A long term preservation archive will use both disk and tape in unison to ensure the best solution is achieved.

### 3.5 Cache

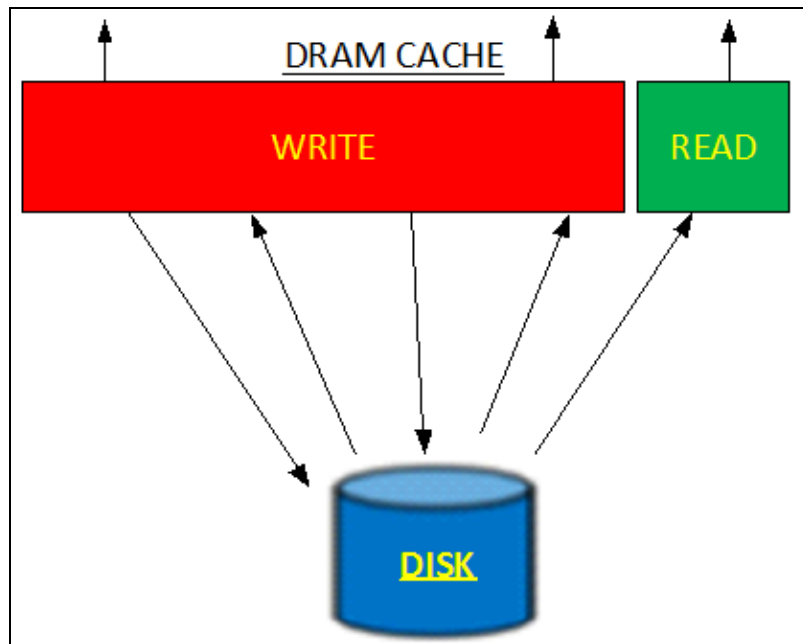
Disk and tape both lag behind the giant leaps that memory has made in recent years. Most high performance compute environments will rely on a form of primary storage to contend with the high performance applications that users have come to expect. This section of the deliverable informs the reader what they need to know about memory caching when implementing a long-term archival infrastructure.

To begin, let's clarify what primary storage means. It is often referred to as internal memory and main memory. Both are known as ROM (Read Only Memory), RAM (Random Access Memory) or Direct-Random Access Memory (DRAM). In the case of digital archiving and passing data to memory ROM is not relevant mainly due to its non-programmable structure which leaves RAM which is classed as primary storage which is a storage location that holds memory for short periods of times while the computer is running. This storage is the fastest memory in your computer and is used by applications for rapid access to data at run-time. For example, when you open a program, data is moved from the secondary storage (for example, disk) into the primary storage (memory). We will take a closer look at methods of caching during this chapter to understand how primary and secondary storage complement each other and what their relationship to long-term archival is.

All storage systems offer caching as a way to speed up the storage subsystem. Cache itself is logically divided into two regions, read cache and write cache as shown in Figure 16. The majority of cache is allocated to the write store for I/O's waiting to be written to disk. Writing to cache is always faster than writing to disk. The data in cache will eventually be sent to disk in large I/O's. Traditional caching is an

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

important consideration when implementing a long-term archival solution. Classic caches lose their contents when power is lost and because cache memory is typically used anytime data is being read or written, any data transaction which requires certainty that it completed fully must be aware of caching. Caches are present when backing up systems to tape for disaster recovery and they are present when data is being written to a long-term archive. In both these cases, it is necessary to ensure that data is not lost during the operation. More information on caching is included in appendix A.4 Disk Caching Disks (DCD).



**Figure 16: DRAM Cache Operation**

### 3.6 Disk Arrays & Redundancy

Writing to two or more disks at the same time or having the same data stored on separate disks enables the data to be recovered in the event of a disk failure without resorting to expensive data recovery techniques. Many organisations today are using disks as an alternative to tape for long-term archival infrastructures. Just like any hardware, disks can and will fail, it is therefore important to be aware of how to protect a long-term archival infrastructure against such loss and how these redundancy techniques are often a balance between an acceptable level of redundancy and cost.

#### 3.6.1 RAID – Redundant Array of Inexpensive Disks

Redundant array of independent disk (RAID) is a technology used for grouping a number of hard disks in a computer or server into an array that you can define as one or more logical drives (or LUN, logical unit number). Each logical drive for example if using Windows Operating System will appear as D:\ or F:\ appears to the operating system as a single drive. This grouping technique greatly enhances logical-drive capacity and performance beyond the physical limitations of a single physical drive. RAID also offers data redundancy as the data bits which make up individual files are combined with parity bits and spread (or striped) across many hard disk drives as opposed to a single drive. In the event of disk failures, and

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

depending on the level of RAID used, the file remains available as the parity bits can be used to reconstruct the absent data.

Many end users will not be familiar with raid as the devices they interact with (PCs and laptops) don't commonly support RAID. Generally, these devices will have a single hard drive. Back-end systems such as servers, will almost always support RAID. Before a server operating system is even installed, a system administrator must create a logical drive. Once this is in place, the server's disk controller can transfer data in parallel to many of the drives that form that RAID array. This parallel transfer enables data transfer rates many times higher than non-RAID systems. Ideally in more enhanced systems with the use of fast cache the data transfer is done when the entire stripe is full making efficient use of data striping. In essence RAID enables high performance logical disks. The ability to respond to multiple data requests provides not only an increase in throughput, but also a decrease in response time. The combination of parallel transfers and simultaneous responses to multiple requests enables disk arrays to provide a high level of performance in network environments.

There are two important factors to be understood with RAID.

1. Data striping. Used for improved performance.
2. Redundancy. Used for improved availability.

When RAID is implemented data is striped across numbers of hard disk drives. This is called the stripe unit size or element size. This unit can vary from 8, 16, 32 and 64kb. Particular environments will have specific requirements such as file and print environments enjoy a lower stripe size of 8Kb or 16Kb while transaction environments which typically move large blocks of data enjoy large sizes of between 32Kb to 64Kb. Note that the stripe size cannot be changed after the array has been created. Different applications are configured to write in particular sizes for example 4k or 8k. The data is presented using block storage in blocks sizes of 512 BYTES so if we have our RAID element set to 64Kb we would write 128 blocks as  $128 \times 512 \text{ Bytes} = 64 \text{ KB}$ . We should not confuse this with our stripe size as this will be determined by the number of disks in our raid pack and hot spare. The most common RAID levels in use today are RAID level 0, RAID 1, RAID 1 E, RAID 5, RAID 5 Enhanced and RAID 6. More information on these RAID levels is included in appendix A.5 RAID Levels .

Each of these RAID levels provides different levels of redundancy which range from none (RAID0) to full mirroring (RAID1). Each level can tolerate differing numbers of individual disk failures but the cost of increasing redundancy is that a higher number of physical disks will be needed to provide the same capacity of usable storage. When implementing disks for use in a long-term archive, it is critical to consider the RAID level which should be used. RAID0 will be the cheapest, but if a single disk fails which is part of a larger RAID0 logical disk (LUN) then all the data on that LUN will be lost. A long-term archive is unlikely to have just one copy of the data in just one location so usually there will be another copy of the data kept in another separate system. However, despite this, careful consideration should be given to allowing sufficient disk redundancy in each repository as the cost of re-creating the system in terms of time and in terms of copying the data from its alternate location will dwarf the cost of some extra physical disks.

### 3.7 Logical Storage Structures

The next part of the archival building block components (Figure 8) that we will cover are the logical storage structures which build on top of physical disks. This is an area which has become extremely interesting over the past number of years as storage vendors have looked beyond the traditional block level and file system

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

level to offer logical solutions that offer a variety of varied characteristics that differ from block and file. It is another area of infrastructure which an organisation implementing a long-term archive should be familiar with. This is relevant to any organisation using disk-based archival infrastructure solutions.

### 3.7.1 Logical Storage Structure Overview

There are three structures that are covered here, namely *Block level*, *File level* and *Object Oriented*. Each has specific characteristics that make them appealing to the storage architect. Although end users may never know what the underlying storage structure is, many software developers will understand the behaviours of each when structuring their software and this discussion will consider the advantages that these structures bring to archival infrastructure implementations.

#### 3.7.1.1 Block Level Storage

A common form of storage is block level storage. This is used extensively by Storage Area Networks (SAN's). SAN's are explained later in section 3.15.2.2 of this deliverable. Block storage enables individual formatting of file systems such as NTFS (NT File System, used in Microsoft Windows), NFS (Network File System), SMB (Server Message Block, also known as CIFS, Common Internet File System) and many others. Block storage is very common and ideal for structured data such as databases. Block level storage uses Fibre Channel, Fibre channel over Ethernet (FCoE) or iSCSI (internet-SCSI) protocols to transfer data to the target. These protocols are also explained later in section 3.9 of this deliverable. The block is used as a data storage structure where data is aligned to reside within one or more blocks. This is called blocking. The end user never actually interacts data at the block level and will not be aware of this level of detail, however in some cases it is important for the developer to understand the underlying storage mechanism. It is generally deemed that block level offers high performance against its alternatives to which we will consider.

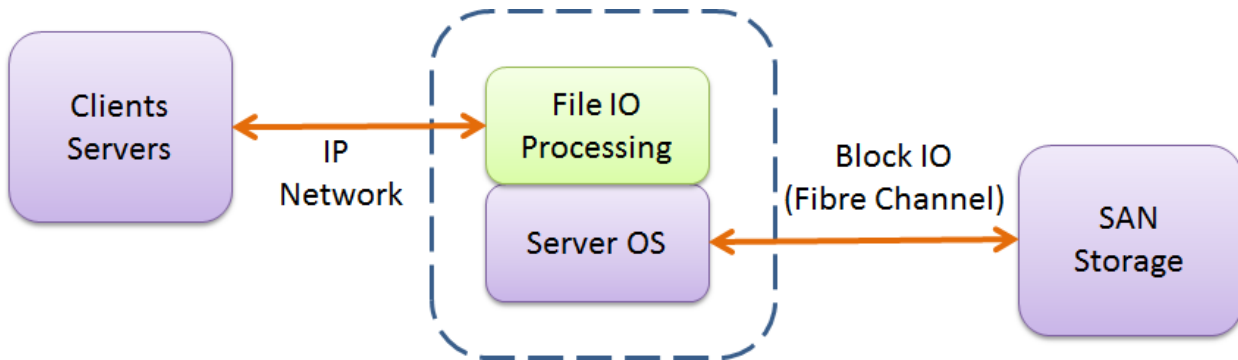


Figure 17: Block Level Storage<sup>16</sup>

#### 3.7.1.2 File Level Storage

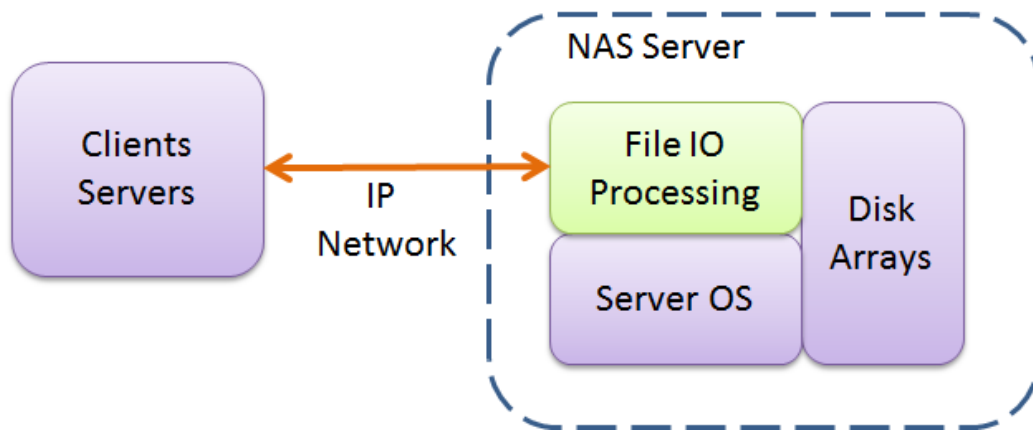
Network attached storage (NAS) is one of the architectures covered in section 3.15.2.2 of this deliverable. NAS uses file level storage. In essence network file systems reside on disk blocks. It is best described as file level sitting on top of block structure. A file access command referenced by either the file name or file handle is translated into a sequence of block access commands on the storage device.

<sup>16</sup> Shane Healy, Storage Architect. Intel Ireland.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

The difference between NAS (Network Attached Storage) and SAN (Storage Area Network) is in whether the data is transferred across the network to the recipient in blocks directly which is how Storage area network SAN manages it, or in a file data stream that was processed from the data blocks which is how NAS manages it. It is a fact that due to this translation by the file system redirector additional overhead is experienced with NAS between the file access and the block access. As a result of this it is deemed that block level offers greater performance than file level structure as it does not require this additional layer of processing.

On the other hand it must be said that there are benefits of this high layer of abstraction mainly ease of use. The built in support for Unix (NFS) and Windows (CIFS) means adding a client server to connect to the NAS is a straight forward process not requiring complex configurations such as with SAN requiring fibre connectivity from the client via a fibre channel switch to the storage array.



**Figure 18: File Level Storage<sup>17</sup>**

### 3.7.1.3 Object Storage

Object storage differs from unstructured file storage in that the objects filed are not housed in volumes or attached to a directory. Objects are simply assigned an ID number or name with detailed metadata and can be retrieved at any time.

As stated earlier in the opening of this chapter, object based storage has created a keen interest notably in the Open Cloud community given the distributed model object oriented storage provides. Openstack Swift Architecture is one such open cloud example we will take a closer look into.

Swift provides redundant, scalable distributed object storage. The distributed part means it is replicated across a number of storage nodes which are clustered. Swift is highly scalable and a durable object storage system that was designed to store large amounts of unstructured data at low cost. It is therefore highly suited to use in long-term archival infrastructures when disk-based storage is being employed. Below is a summary of Swift.

- ✓ Open source , Part of Openstack, rapidly evolving
- ✓ Distributed object storage platform

<sup>17</sup> Shane Healy, Storage Architect, Intel Ireland.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

- ✓ REST API, S3 support
- ✓ Designed for cloud storage use cases
  - Can be used for any purpose, initial use cases are glance image and snapshot storage
- ✓ Uses Commodity hardware – no custom or proprietary hardware/software
- ✓ No single point of failure, all storage and proxy nodes are redundant

CEPH is another Open Cloud logical storage solution which uniquely delivers object, block, and file storage in one unified system. The CEPH Object Store, called RADOS, is the object storage component for CEPHFS file systems, CEPH RADOS Gateways, and CEPH Block Devices.

CEPHFS, CEPH RADOS Gateway, or CEPH Block Devices can be used in an organisation’s deployment. You may also use all three interfaces with the same CEPH Object Store cluster. There’s no reason to build three different storage clusters for three different types of storage interface.

We now understand more about our archive architecture from the raw disk level and the logical storage structure. We have identified that Block level is the most common form of logical structures in use today and is used extensively on Storage Area Networks. File based systems most common with NAS are in essence a file level structure sitting on top of block structure. Finally, object based storage is less common but is becoming a fast growing structure in Open Cloud based on its flexibility and is strongly worth considering if building a disk-based archive infrastructure. There is extensive additional information and discussion included on this topic in appendix A.6 Object-based storage.

The next stage to consider is the actual file system. How do we want our files managed? A file system is ultimately where our digital object will be housed and recognisable to the end user.

### 3.8 File Systems

Each platform has a different file system which is unique to that system. Each file system has varying characteristics which make file systems efficient for some situations and inefficient for others. File systems have developed over time with some remaining while others becoming obsolete. This is another point that makes file system evolution relevant to TIMBUS as the file system itself in which your data is stored may be no longer readable by future computer systems.

A key element in file systems is possessing understanding of the underlying nature of the data. This is called metadata. All file systems store metadata along with the object itself. Examples of this include the last modified date and the location that the file is stored on the storage device

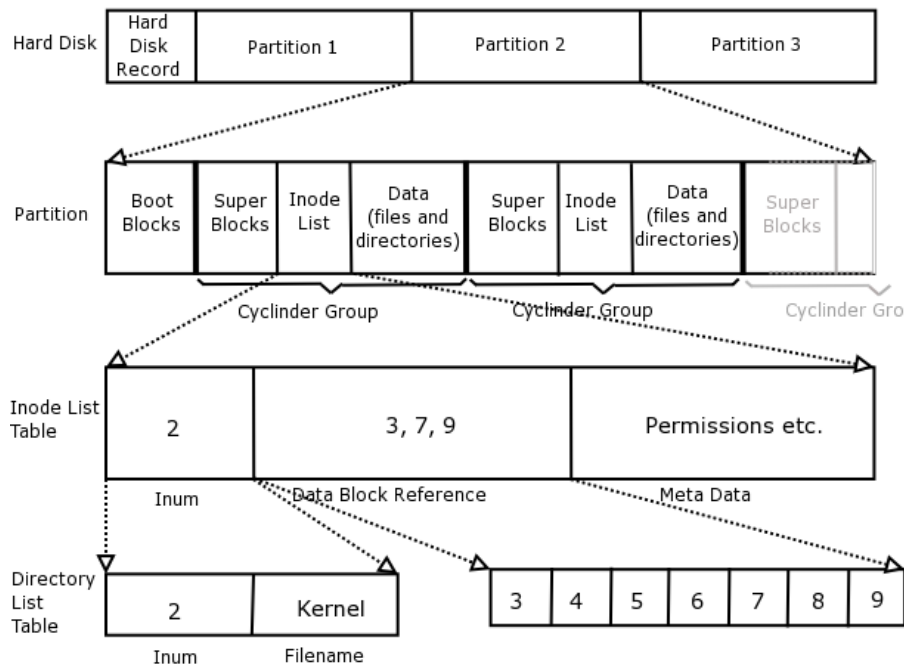
#### 3.8.1 Unix/Linux File Systems

Taking a closer look at the UNIX file system (UFS) we can identify that it is made up of many parts as detailed below in Figure 19. This level of information is in itself important metadata to consider when preserving and re-deploying current generation business processes.



<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

## UNIX File System Layout



**Figure 19: The UNIX File System**

(source: <http://www.learnlinux.org.za/courses/build/internals/ch08s04.html>)

Boot blocks are located at the beginning of the partition and are reserved to allow a partition to be bootable. Next is a superblock which identifies the specific UFS file system and other aspects relating to tuning behaviours. A collection of cylinder groups contains a backup of the superblock, the i-nodes (file indexes), file attributes (metadata) and data blocks.

To a UNIX system a file system is a device that has been formatted to store files. The exact formats by which the files are stored are not important. The UNIX system provides a common interface for all file system types. Today most Linux distributions use a journaling file system so when a disk write is performed an entry is written to a log or journal before the write to disk occurs. In some cases, this can slow down the process of writing to disk but offers some redundancy protection against sudden power outages. In other cases, it speeds up system responsiveness as multiple disk writes (for example a user saving a word processing document) can be queued up and if a particular write request is quickly superseded by another, then only 1 write to the disk (the latest one) actually takes place.

The Ext3 file system is common with Linux distributions as it supports 256 character filenames, 16TB max file system size and 2TB of max file size. Ext4 is the latest offering of file system with Linux currently, Ext4 adds 48-bit block addressing, so it will have 1 ExB of maximum file system size and 16 TB of maximum file size. It is possible to upgrade Ext3 to Ext4 without any impact to the file system. Ext4 is available with Ubuntu 9.04 as a manual partition option when installing Ubuntu. It is also supported by Caxia Magica's CM19 Linux distribution. Caxia Magica is one of the TIMBUS partners and their latest Linux distribution, CM19, is being used in the TIMBUS test bed as part of Task T6.7 which is detailed in deliverable D6.4.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

### 3.8.2 File Sharing Protocols

This section of the deliverable covers some of the common file sharing protocols in use today. These are relevant to preserving business processes for several different reasons. Firstly, there are a lot of them and a business process may use more than one of them in its operation. Secondly, just like file formats and hardware technologies, file sharing protocols are evolving constantly. New ones are likely to be developed in the future which will make today's protocols obsolete. On top of that, all of today's protocols have gone through several version iterations. Each version may have added, or dropped, support for various security protocols, such as NTLM, LanMan or perhaps to stop plain-text passwords being sent over the network. While those are worthy advancements from a security perspective, a business process archivist must be aware that the environments in which their processes run have dependencies on network and security protocols that they may not even be aware of. Lastly, file sharing protocols are directly relevant to the implementation of a long-term archive infrastructure as they may be employed to copy the digital objects we need to preserve from their source location to the preservation repository (the preservation module in the D5.5 architecture) and from there to the target re-deployment location. The following section of the deliverable relates some information on these protocols.

#### 3.8.2.1 Common Internet File System (CIFS)

CIFS is the native file-sharing protocol used in Microsoft Windows. CIFS defines a series of commands used to pass information between networked computers. The redirector, shown below in Figure 20 uses CIFS to make requests to the protocol stack of the local computer.

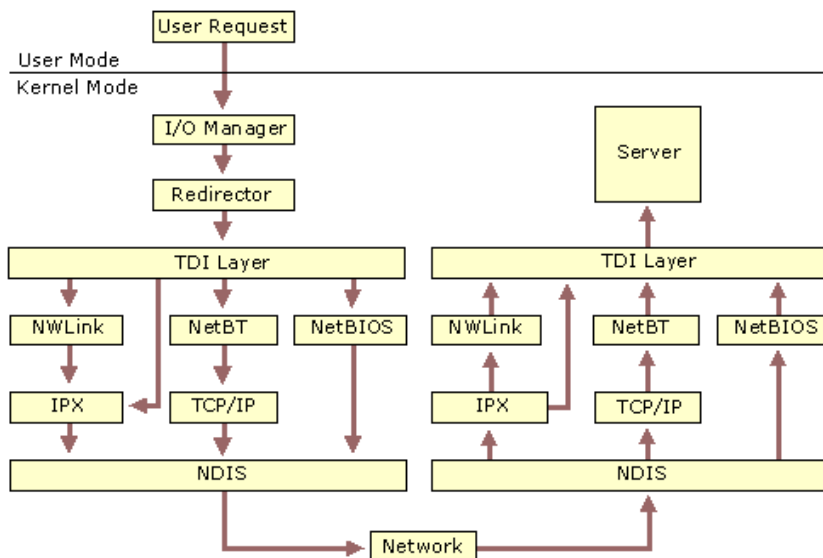


Figure 20: CIFS Architecture<sup>18</sup>

When there is a request to open a shared file, the I/O calls the redirector, which in turn requests the redirector to choose the appropriate transport protocol. For NetBIOS requests, NetBIOS is encapsulated in

<sup>18</sup> [Technet.microsoft.com](http://Technet.microsoft.com)

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

the IP protocol and transported over the network to appropriate server. The request is passed up to the server, which sends data back to satisfy the request.

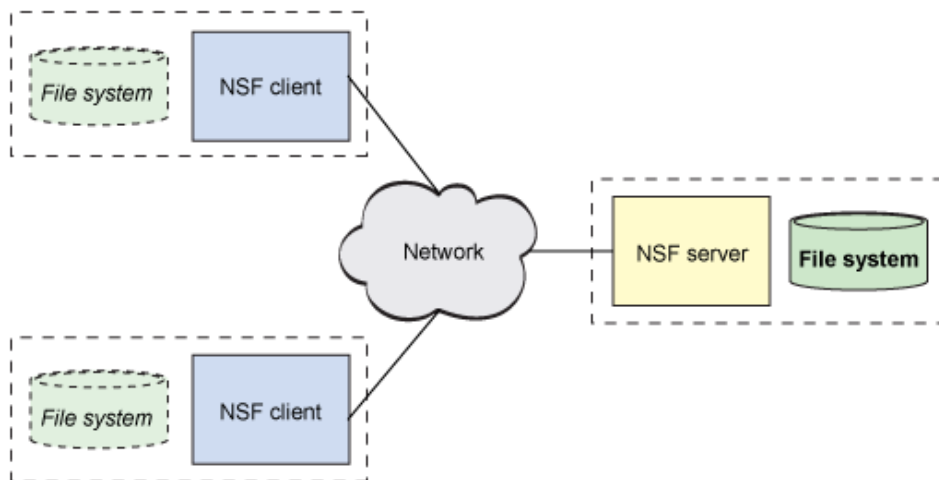
CIFS supports both anonymous transfers and secure, authenticated access to named files. File and directory security policies are easy to administer. CIFS has gone through several version changes which have generally related to security enhancements.

### 3.8.2.2 Network File System (NFS)

NFS allows a system to share directories and files with others over a network. By using NFS, users and programs can access files on remote systems almost as if they were local files.

All versions of NFS can use *Transmission Control Protocol (TCP)* running over an IP network, with NFSv4 requiring it. NFSv2 and NFSv3 can use the *User Datagram Protocol (UDP)* running over an IP network to provide a stateless network connection between the client and server.

When using NFSv2 or NFSv3 with UDP, the stateless UDP connection under normal conditions has less Protocol overhead than TCP which can translate into better performance on very clean, non-congested networks. The NFS server sends the client a file handle after the client is authorized to access the shared volume. This file handle is an opaque object stored on the server's side and is passed along with RPC requests from the client. The NFS server can be restarted without affecting the clients and the cookie remains intact. However, because UDP is stateless, if the server goes down unexpectedly, UDP clients continue to saturate the network with requests for the server. For this reason, TCP is the preferred protocol when connecting to an NFS server. NFS architecture is illustrated below.



**Figure 21: NFS Architecture<sup>19</sup>**

As we discuss the hardware elements a very important fact not yet discussed it data protection and redundancy. Block and File level storage offer little resistance against a hardware failure on a disk. With the large capacity disks available today, it is a risk that could prove costly. On large scale storage systems it is

<sup>19</sup> <http://www.ibm.com/developerworks/linux/library/l-network-fileystems/?ca=drs->

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

simply not an option to have data written to one disk and then sequentially written to another disk due to performance reasons.

### 3.9 Storage Interfaces

This section of the deliverable now considers the interconnectivity between our storage models and the remainder of the compute environment. As we discussed with storage models there are direct attached storage models and remotely connected models utilising high bandwidth links such as Ethernet and fibre channel. This section is without a doubt the most complex yet interesting part of storage architectures.

#### 3.9.1 Storage Interface and Connectivity Overview

There are a number of industry standard interfaces available enabling a device to interconnect with a storage device. An example of this is a host computer and hard drive or a storage area network connecting with a pool of host machines over a local area network. These interfaces range from serial which allows for one bit at a time to be sent in sequence, parallel enabling more than one bit to be sent at the same time. Our network environment and interfaces plays a major role in the archival architecture. In isolation we understand the importance of having a connected network. There is a growing trend of end to end convergence of data centre networks to enable transport fabrics such as Fibre Channel over Ethernet (FCoE) and Internet SCSI (iSCSI) to use Ethernet channels to communicate. We will discuss the various types of interfaces from ATA, SATA, SCSI, iSCSI, USB, Fibre Channel and Ethernet and how convergence is shaping the future.

##### 3.9.1.1 SCSI

The Small Computer System Interface (SCSI) was developed in 1986 and offers a universal parallel interface. SCSI is noted for its SCSI bus to which devices are connected along with a total of 8 although this can be extended with various combinations of SCSI available today such as wide SCSI which offer up to 15 devices. SCSI uses a controller to send and receive data and power to other SCSI enabled devices. Each SCSI device has an ID and it must be unique. For example on Wide SCSI you can have 15 devices and one controller, 0-15 IDs are used. You can have SCSI connectors with 50, 60 or 80 pins. It is important to terminate the bus with the last device.

Today SCSI still dominates the storage interface environment due to its reliability and scalability. SCSI can expect speeds of up to 320 MBps. SCSI hard disks are ideal for large enterprise applications due to their efficient mechanics. SCSI is more expensive than SATA due to the extensive design and testing process.

SCSI does have some challenges. It requires configuration to work such as ID allocation and termination. There is no standard interface. There are variations such as Fast, Ultra and Wide which all have specific requirements. The new versions of SATA are replacing a lot of SCSI drives.

##### 3.9.1.2 iSCSI

iSCSI, or *internet SCSI*, carries SCSI commands over IP networks, iSCSI is used to facilitate data transfers over intranets and to manage storage over long distances. With standard SCSI we know that when an application or user sends a data request the operating system will generate a SCSI command to interact with the storage system on disk. With iSCSI the SCSI request is encapsulated and a packed header is added for transfer over the network. On the receiving end the packet is de-assembled and the SCSI command is issued.

D5.3_M24_Architecture_For_VSRI	Dissemination Level: Public	Page 36
--------------------------------	-----------------------------	---------

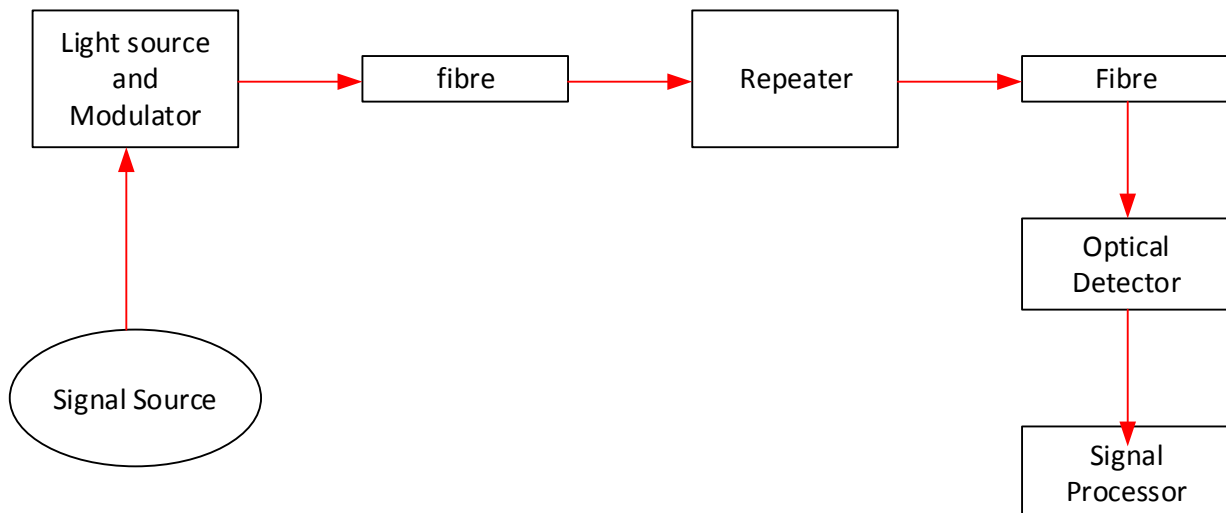
<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

iSCSI is not alone in data transmission over a LAN. As we know SAN's rely heavily on fibre fabric's to connect storage to hosts.

### 3.9.1.3 FCoE or FCIP

Fibre Channel over IP or over Ethernet (FCoE) is a protocol designed to run Fibre Channel (FC) over 10GB Ethernet environments. FCoE is ideal for environments where there is a requirement or desire to consolidate Ethernet and Fibre Channel into one environment. This obviously makes managing one switch easier than having a SAN and network switch. It is generally deemed that FCoE is more limited than iSCSI as it can only be used with existing fibre channel hardware whereas iSCSI is more compatible with existing networks and iSCSI enabled devices such as routers and switches.

Fibre Channel is highly reliable and very fast way to interconnect storage. Fibre Channel has been the catalyst for the mass adoption of SAN. With the advent of optical media for the purposes of transferring digital data, light rays are high-frequency electromagnetic waves with a short wavelength in the micron range. For the optical fibre communications, light is made to travel through a medium of glass. Figure 22 is an illustration of a fibre channel communication system.



**Figure 22: Fibre Channel Communication System<sup>20</sup>**

### 3.9.1.4 HBA – Host Bus Adaptor

Simply put a Host Bus Adaptor connects the host machine to a network or storage device. In most cases you will see HBA referred to in relation to fibre channelled environments. Similar to a network interface card NIC a HBA has a unique World Wide Name WWN similar to the MAC address on a NIC. On a host the HBA will house the incoming fibre cable from the remote fabric switch.

### 3.9.1.5 10G Ethernet

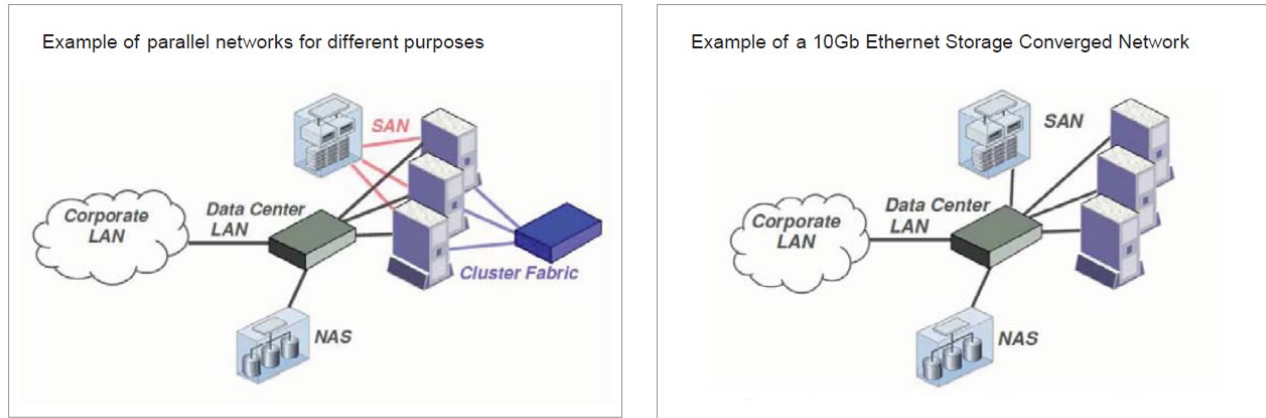
10-Gigabit Ethernet 10GBASE-T is a communication technology that offers data speeds up to 10 billion bits per second. With Increasing processing power with the adoption of multi-core processor-based server

<sup>20</sup> Rodger Mooney, Intel Ireland.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

platforms the need for greater network bandwidth and more efficient processing of network traffic has become a reality. It is expected by industry analysts that 10GB will develop into 40 GB and even 100GB in the next ten years with many 10GB physical ports sold as default on networking equipment today.

Avnet.com offer an illustrated example of the multifaceted environments that are common today such as SAN and NAS compared to an integrated 10GB network layer connecting all devices to storage.



**Figure 23: Example SAN Architectures<sup>21</sup>**

The main components of a 10 Gigabit Ethernet (GbE) solution are Host Bus Adaptors (HBA), which connect a host system to other network and storage devices. Switches are the computer networking devices that physically connect the various network segments.

10GB is built on the Ethernet technology used in most of today's local area networks, 10-Gigabit Ethernet offers a more efficient and less expensive approach to moving data on a backbone connection between networks and end to end networking. With the use of optical fibre 10-Gigabit Ethernet can replace existing networks that use ATM (Asynchronous Transfer Mode) switches and SONET multiplexers. 10-Gigabit Ethernet uses full-duplex transmission combined with multimode fibre, 10-Gigabit Ethernet will support distances up to 300 meters. With single mode fibre, it will support distances up to 40 kilometres. Smaller Gigabit Ethernet networks can feed into a 10-Gigabit Ethernet network.

### 3.9.1.6 Infiniband

Infiniband offers RDMA which is Remote Direct Memory Addressing which enables high performance IPC which is inter processor communication. This makes Infiniband a true high performance compute environment. To summarise Infiniband we can say that it offers high bandwidth, low latency and low CPU utilisation.

RDMA service passes messages across the network, between processors. Messages are passed directly between registered memory locations, without operating system intervention or data copying.

<sup>21</sup> [http://www.ts.avnet.com/uk/solution\\_guides/assets/10gb\\_ethernet\\_solution\\_guide](http://www.ts.avnet.com/uk/solution_guides/assets/10gb_ethernet_solution_guide)

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

This highly efficient communication is useful for applications such as MPI for HPC, traditional socket applications, storage applications and file systems. All through the use of specialised APIs over a common transport.

InfiniBand’s physical signalling technology has always stayed well ahead of other network technologies, allowing the greatest bandwidth of any networking protocol. InfiniBand today runs at 56 GB/s with a road map to get to 100 GB/s in one-and-a half years.

### 3.9.1.7 USB, Firewire, Thunderbolt™

While these may not be typically viewed as solutions for enterprise or large scale long-term archival infrastructures, they are none the less very relevant when considering smaller archive requirements or even personal data archival in the home. USB, Firewire and Thunderbolt, as we will see, are cost effective and are advancing rapidly, primarily to meet the demand in IO transfer speeds required by end users today. With capacities of up to 4TB for a single disk today, it becomes possible for small offices or home users to have a very cost effective and large pool of storage. These can even be deployed as shared storage solutions by plugging them in to home NAS systems and can in turn be augmented with home or small office RAID capabilities at a hardware or software level.

This market as it exists today was largely shaped when a group of 7 companies, including Intel began development of USB (Universal Serial Bus) in 1994. Today it is the most commonly known method of attaching peripheral devices such as a printer, mouse, scanner, digital camera, head phones or speakers to a PC. What makes USB even more attractive is the fact that USB is plug and play which means once your PC supports USB you can connect a device via the USB port and it will be detected and begin operation almost instantaneously. This is a dramatic difference from prior interfaces such as serial connectors which would require BIOS configurations to enable the COM ports on the PC. This was a major hurdle for PC manufacturers who needed to make PC’s more user friendly.

Just as we have seen with the industry in general, the USB standard is continuing to evolve and so has its physical interfaces. There are three forms of USB available today, USB 1.1 (released 1996), 2.0 (launched 2000) and 3.0 (2008).

USB ports are located on a PC or server along the chassis. The requirement for these ports has risen as USB has become more popular. In some cases the amount of ports on a PC, laptop or smartbook does not meet the requirement so a USB hub can be attached which allows for a greater number of ports to be available. Another item to note with USB is when a device is attached to the USB port it will often, but not always, have no requirement for a separate power supply as it takes the power (5 volts) from the USB port via the PC. This is another excellent advantage of USB devices. In the case where a device requires higher voltage (12 volts is typical) an external power supply is required or else *Powered USB* can be used by some proprietary extensions to supply 5, 12 or 24 volts.

Just like other aspects of IT systems, the connector types have become more diverse as the USB standard evolved. Smaller devices such as cameras and smart phones have required that ‘mini’ and ‘micro’ versions of the initial USB A and B-type connectors be developed. The higher performance of USB 3.0 has also brought significant physical changes to the adapter interfaces. Figure 24 below illustrates these interfaces to give the reader an feeling of how much diversity there is in the interfaces of a single standard over a period of only 16 years. 16 years is a very short time in comparison to how long many organisations need to keep business processes for.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI



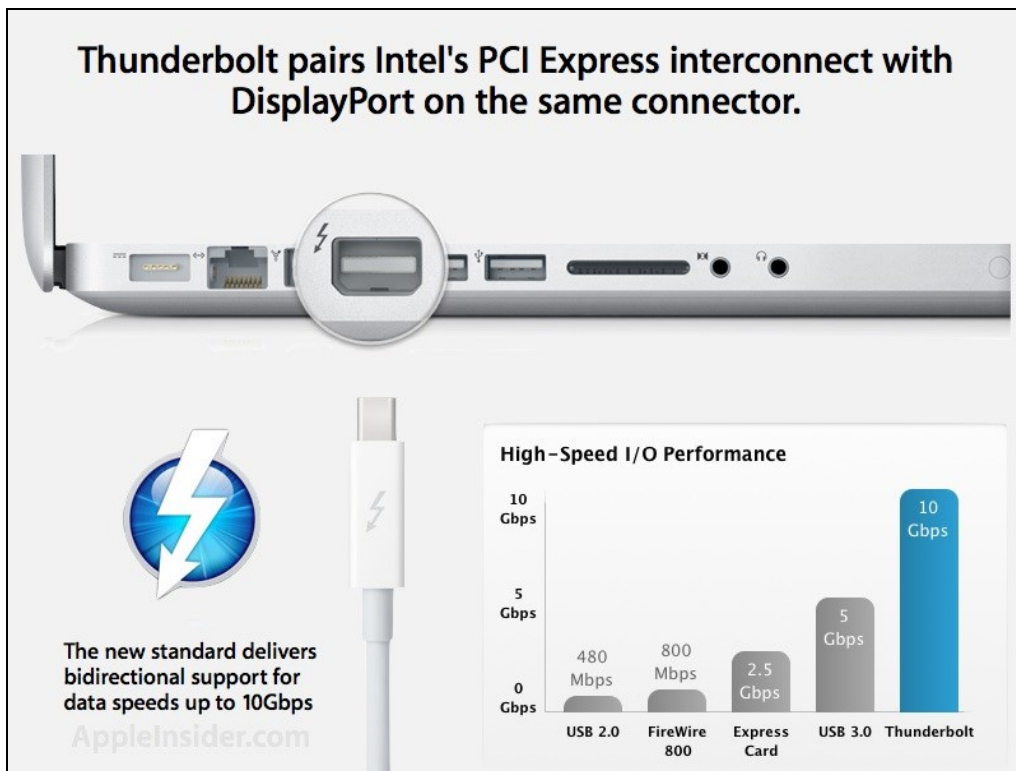
**Figure 24: USB Adapter Connectors<sup>22</sup>**

USB could be described as a disruptive technology as it has become dominant today replacing older interfaces such as RS232 which was slower and had design restrictions. However, USB itself faces stiff challenges and as a result is evolving continually. A new serial standard called Firewire (IEEE 1394) is a complementary standard developed by Apple with major releases in 1995, 2000 and 2005. Its latest version was faster than USB2.0 with speeds of 800 Mbps and it is isochronous which means the data delivery is constant. Intel however, were once again involved in the development of USB3.0 which has made a massive 5-fold increase in performance over Firewire providing 5Gbps versus 800 Mbps for Firewire. The most recent addition to this market has been another Intel developed interface called Thunderbolt™. Thunderbolt™ doubles the speed available in USB3.0 to an impressive 10Gbps as shown below in Figure 25.

<sup>22</sup> <http://www.l-com.com/content/USB-Tutorial.html>



<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI



**Figure 25: Intel's new Thunderbolt™ Interface**

### 3.9.1.8 Storage Interface Summary

When it comes to large enterprise solutions SCSI, Fibre Channel, Infiniband, iSCSI and 10GBe offer enterprise interfaces for connecting and transferring data across a storage network. Other interfaces such as SATA and USB offer internal connectivity for hard disks and other computer peripherals used in data storage.

As mentioned at the beginning of this section on storage interfaces and connectivity, traditionally data traffic from servers to centralized storage has been carried on specialised Fibre Channel networks, designed to provide maximum performance and availability. On the other hand client server traffic traverses over Ethernet LAN.

Recent innovations in Ethernet networking present the possibility of network unification through the use of a single technology for both SAN and LAN. Advances in Ethernet technology and the maturation of protocols such as iSCSI and FCoE offer deterministic performance, low latency, and constant data availability on Ethernet. An enhanced lossless version of Ethernet is at the heart of enabling FCoE, which offers current users of FC SANs a path to move to network convergence while maintaining zoning practices, skill sets, and management tools that are already familiar. On the server side, the prospect of shared redundant ports for both front-end LAN traffic as well as back-end SAN networking promises cost savings as well as an increased ability to virtualise server I/O.<sup>23</sup>

<sup>23</sup> Forrester Consulting 2011

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

Network connectivity and storage interfaces are a major component in the archival architecture. A solid understanding and appropriate implementation of these interfaces and connectivity devices is critical for a robust preservation archive.

It is clear that many possible interfaces exist to interact with your storage device. This is a rapidly evolving area if the IT industry and the choice of storage interface will have direct implications for the architecture of a long-term infrastructure for business process preservation.

### 3.10 Backup and Recovery (BaR)

Most archival solutions today are simply the same system that the organisation uses for Backup and Recovery (BaR). The problem with this is that BaR systems are not designed for long-term archival. Their primary purpose is to support business continuity through supply of a disaster recovery service or to aid business efficiency through the backup of critical data against accidental data loss or corruption. A review of BaR systems is highly relevant when considering the requirements of a long-term infrastructure for business process archival. Data loss can be very costly, particularly for organisations in the small and medium business market where the difference between survival and closure can rest on the ability to recover from a disaster. The typical understanding of this, and the area at which traditional BaR systems are aimed, is in recovery from operational system outages. The TIMBUS context is significantly different from this. In TIMBUS, the business continuity focus is on risks to the business which could be mitigated or eliminated through the preservation of a business process. For example, this could be the preservation of the design system used for dams which is one of the TIMBUS use cases. The TIMBUS preservation archive must implement best practices, or advance state-of-the-art to ensure it is protected or else we stand to lose data. At the very least, critical data loss will have a financial impact on companies of all sizes. SNIA defines a backup as a collection of data stored on non-volatile storage media for purposes of recovery in case the original copy of data is lost or becomes inaccessible. Non-volatile storage is persistent when the power supplying it is not available (for example, tape, disk or flash memory). Volatile storage is not persistent so once the power source is removed, any data is lost (for example the memory in a laptop).

It is interesting to look at what can happen when things goes wrong with archives when they are not properly managed. NASA admitted in 2006 that no one could find the original video recordings of the July 20, 1969, moon landing. NASA was eventually able to identify what had happened. The tapes were part of 200,000 tapes that were degaussed, magnetically erased and re-used to save money<sup>24</sup>. In a related story in 2008, NASA discovered 48,000 lbs of 70mm tape at an abandoned McDonalds of high resolution lunar images as shown below in Figure 26. The data was 40 years old and was beamed from the Apollo missions to Earth where they were recorded to tape. The data was highly valuable as it was the only surviving copies available and the raw images had been captured in high resolution. Recovering the data in this case required technological archaeology. There was only one functioning Ampex tape player in the world which luckily, one person on the planet who retired in 2008, still knew how to refurbish. The details and scale of these problems will differ for different organisations, but this was an interesting example of what can happen, even to a body like NASA and it is still true today that many organisations struggle to manage tape archive media over time.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI



**Figure 26: Where were NASA’s ‘lost’ Lunar Landing Images?**<sup>24 25</sup>

The backup and recovery environment will span both the hardware and software components of archive architectures. In relation to hardware the backup and recovery has many components to consider which we will now look at.

### 3.10.1 Backup and Recovery Architecture

As there are many backup and recovery solutions available today it is not possible to reflect the architecture of each one. The material in the example below is based on a common enterprise solution from Symantec called NetBackup is used to reflect the capabilities and structure of many of the enterprise backup solutions available today. The TIMBUS consortium does not endorse any particular vendor solution; rather this is presented to inform the reader of the basic components used by Backup and Recovers (BaR) systems today.

This is an important topic to cover in TIMBUS because many organisations confuse backup with archive. Although this will depend on your expectations the simple facts are, a backup is not an archive and an archive is not a backup. They both have different objectives. The backup environment should protect our archive. The backup and recovery environment underlies the hardware stack and offers recovery in the event of hardware failure, disaster, intentional attacks and human error. The backup environment is not an archive nor should it be used as one.

#### 3.10.1.1 The Master Server

Master servers manage backups, archives, and restore workflows. The master server is responsible for media and device selection. Typically, the master server contains a catalogue. The catalogue contains the internal databases that hold all information about backups, backup scheduling and system configuration. In essence the master server acts as a meta-data server, using the catalogue to maintain an index of what it

<sup>24</sup> <http://www.reuters.com/article/2009/07/16/us-nasa-tapes-idUSTRE56F5MK20090716>

<sup>25</sup> [http://www.thelivingmoon.com/47john\\_lear/02files/Lunar\\_Orbiter\\_Tapes\\_Found.html](http://www.thelivingmoon.com/47john_lear/02files/Lunar_Orbiter_Tapes_Found.html)

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

stored in the BaR environment and it does not actually control any backup data flow. This is handled by the media servers.

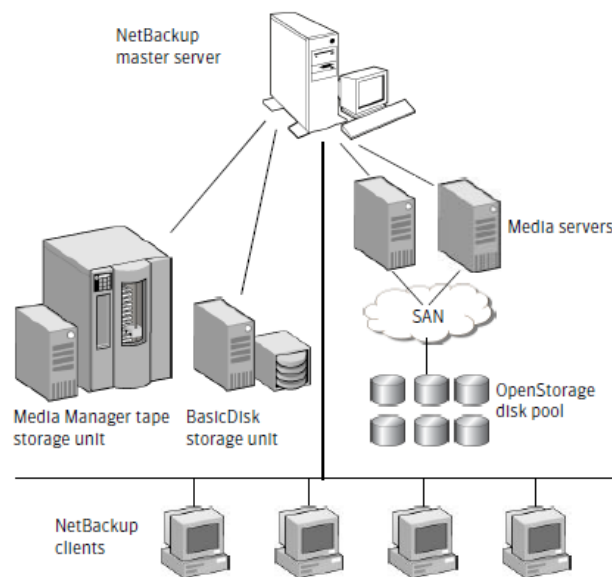
### 3.10.1.2 Media Servers

Media servers provide additional storage by allowing the BaR systems to use the storage devices that are attached to them. Depending on the requirements a media server may have a tape device connected directly via a SCSI cable or the media server maybe connected to a fibre channel fabric via an internal Host Bus Adapter (HBA) card in the media server. This connection enables storage based backups or automated tape library backups which we will discuss in the next chapter.

Media servers can also increase performance by distributing the network load. Media servers can also be referred to by using the following terms:

- Device hosts when tape devices are present
- Storage servers when I/O is directly written to disk
- Data movers (when data is sent to independent, external disk devices like OpenStorage appliances).

During a backup or archive job, the client sends backup data across the network to a BaR Media server. The Media server manages the type of storage that is specified in the backup policy. During a restore, users can browse and select the files and directories to recover. The Media server finds the selected files and directories and restores them to the requested location.



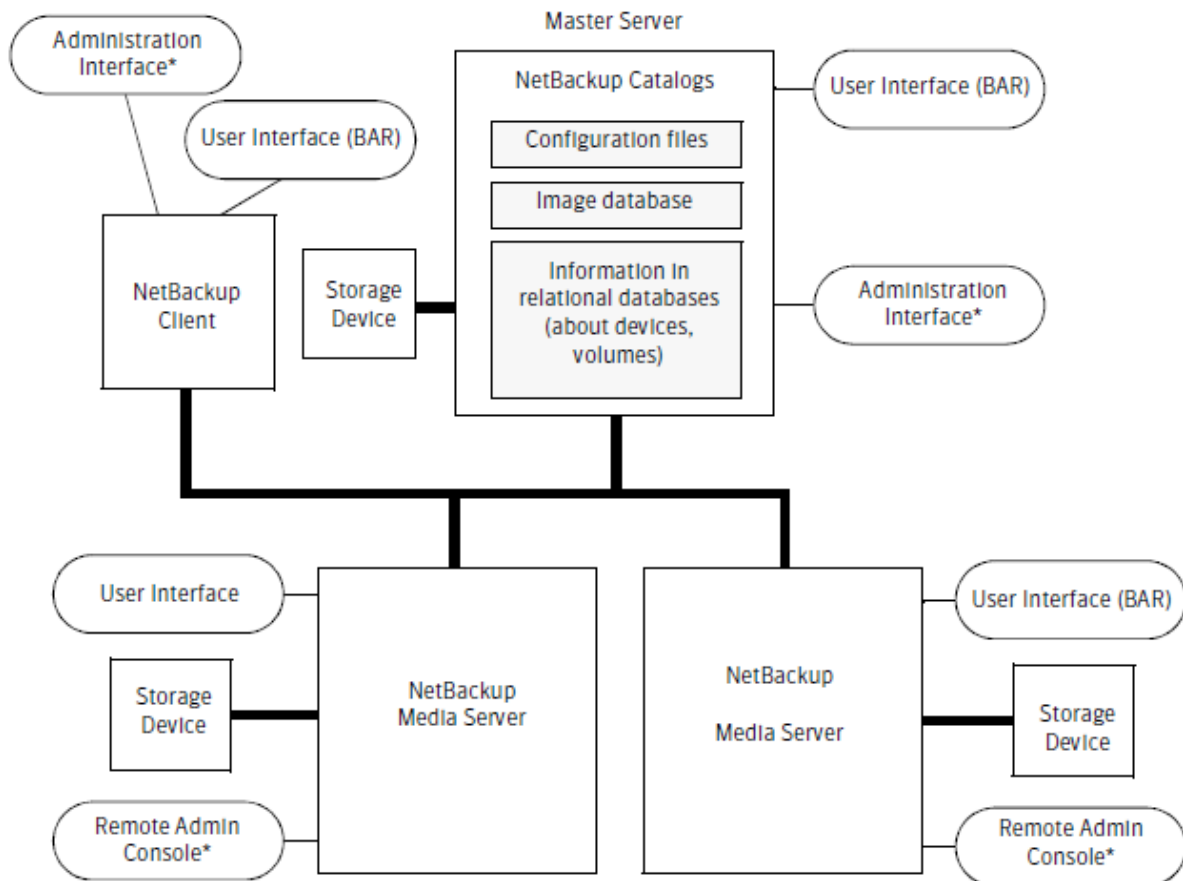
**Figure 27: Symantec NetBackup Architecture**<sup>26</sup>

<sup>26</sup> <http://www.symantec.com/business/support/index?page=content&id=DOC5159>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

### 3.10.1.3 Catalogue Management Server

The Master server is the default location for indexes and catalogues. The catalogues include the media and the volume database. The volume database contains the media usage information and the volume information that are used during the backups. The diagram below is taken from the Symantec NetBackup administration guide and outlines where the catalogues are located.



**Figure 28: Location of NetBackup Catalogues<sup>27</sup>**

Every BaR solution on the market will use similar concepts and they all maintain a catalogue or index of some type. The catalogues contain information on every client backup job. Catalogue backups are tracked separately from other backups to ensure recovery in case of a server crash. The catalogue is a critical database and must be protected. This can be done by backing up the database regularly. In the past this process was disruptive as it meant all other backups needed to be halted while the catalogue ran. Now with the use of hot database backups it's a much quicker process. A catalogue can be recovered.

### 3.11 Connectivity & Redundancy

It would not be possible to implement the TIMBUS testbed in task T6.7 without considering the practical aspects of storage connectivity. The technologies available today for this are covered in this section of the

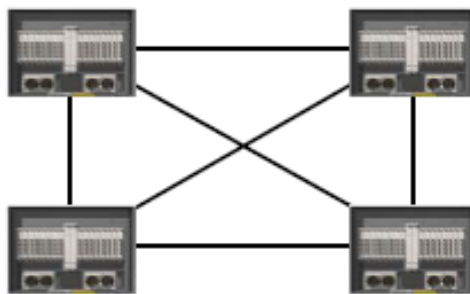
<sup>27</sup> Symantec Netbackup User Configuration Guide

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

deliverable. These connectivity methods have differing advantages and disadvantages and offer redundancy in many cases if implemented correctly. Redundancy is important in a long-term archive to avoid data loss.

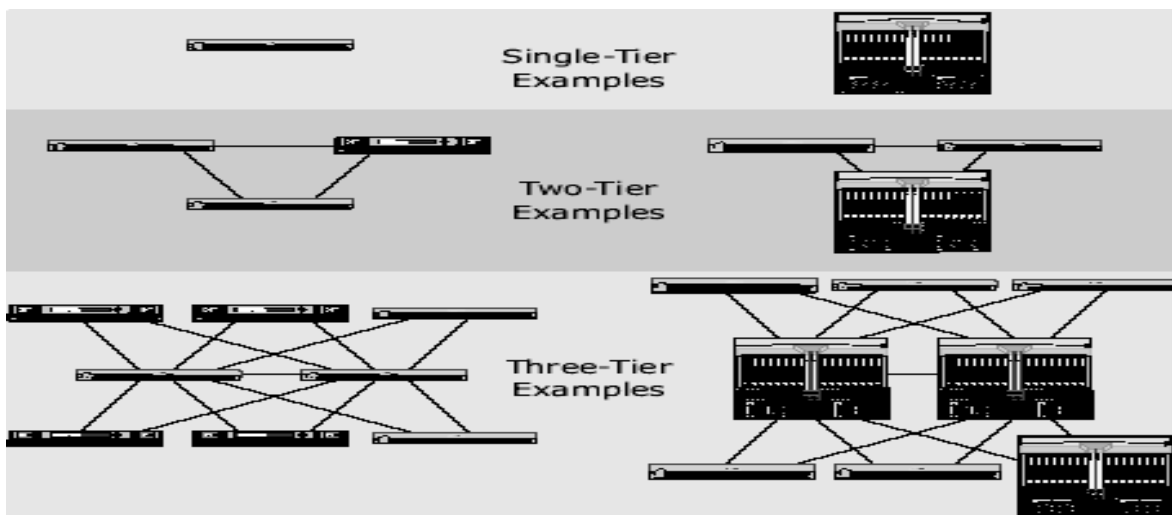
### 3.11.1 Fibre Channel Fabric

The fabric is a virtual space used by each connected node to communicate with each other via switches connected together. This fabric enables traffic to be routed between each node. Each HBA has a WWN World Wide Name which makes it uniquely identifiable. These switches can be linked together to increase capacity and efficient management and this is called inter switch linking ISL. Fabric can become a single point of failure (SPOF) so high availability and high performance approaches can be taken to mitigate this. One such approach is called the MESH fabric.



**Figure 29: Basic MESH Fabric Architecture Example<sup>28</sup>**

There are further developments to the MESH such as outlined below with tiered approaches based on requirements. All of these remain based on the general approach shown in Figure 29.



**Figure 30: Specialisations of the basic MESH Architecture<sup>29</sup>**

<sup>28</sup> *Storage Area Networks, Shane Healy. Intel Ireland.*

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

### 3.11.2 Zoning

To make sense of all these connected devices a method called Zoning is applied. Zoning is a switch function that allows devices within the fabric to be logically segmented into groups that can communicate with each other. Most zoning implementations are based on the Simple Name Service ANSI standard.

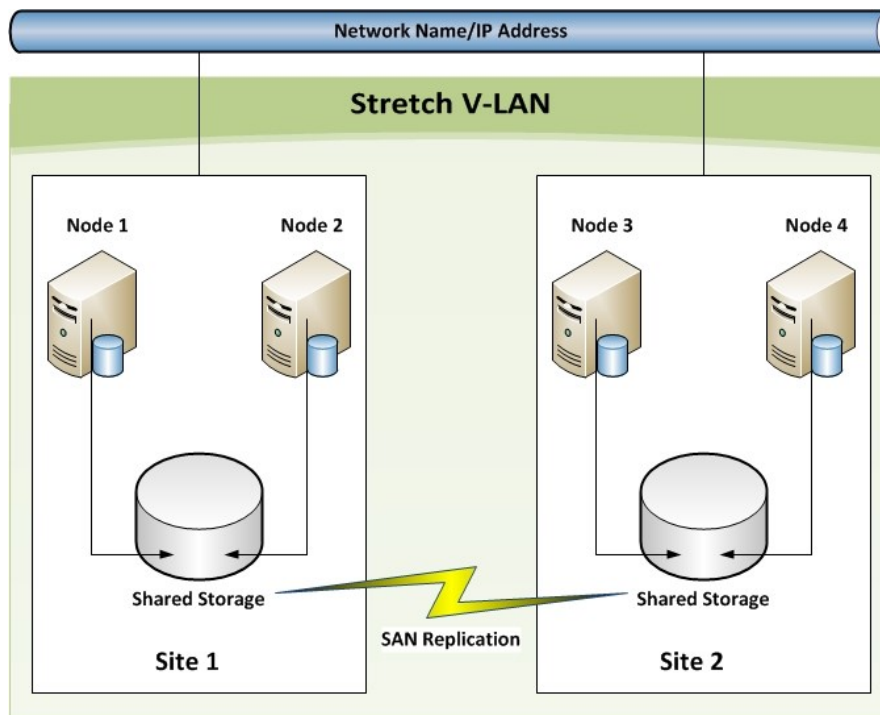
When a device logs into a fabric, it is registered by the name server. When a port logs into the fabric, it goes through a device discovery process with other devices registered as SCSI FCP in the name server. The zoning function controls this process by only letting ports in the same zone establish these link level services.

A collection of zones is called a zone set. The zone set can be active or inactive. An active zone set is the collection of zones currently being used by the switched fabric to manage data traffic.

### 3.11.3 Other forms of Redundancy & High Availability

#### 3.11.3.1 Failover Disk Clustering

Although high availability is not data redundancy it does add additional value to items such as RAID. One of the well-known high availability offerings is failover clustering.



**Figure 31: High Availability<sup>30</sup>**

<sup>29</sup> *Storage Area Networks, Shane Healy. Intel Ireland.*

<sup>30</sup> <http://www.sqlsoldier.com/wp/wp-content/uploads/2011/01/GeoCluster.jpg>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

Fail-over clustering is designed to provide very high server availability, for environments with critical service high availability is essential.

In a failover cluster, there can be two servers or more depending on the environment. The primary server provides the service in normal situations. A second server called the failover computer is present in order to run the service when the primary system fails. The primary system is monitored with active checks every few seconds called a heartbeat check. The system performing the monitoring may be either the failover server or an independent system called the cluster controller. In the event of the active system failing, or failure of components associated with the active system such as network hardware, the monitoring system will detect the failure and the failover system will take over operation of the service.

A key element of the fail-over clustering approach is that both computers share a common file system. One approach is to provide this by using a dual ported RAID (Redundant Array of Independent Disks), so that the disk subsystem is not dependent on any single disk drive. An alternative approach is to utilise a SAN Storage Area Network.

An interesting aspect of Figure 31 is that it advocates a multi-site strategy for archival hosting. That is a great option if you are a large multinational organisation. What can you do if you are not? Tom Cramer of Stanford University spoke at PASIG 2012 about the Digital Preservation Network (DPN)<sup>31</sup> that has been created by a large number of North American universities. Individual archives held by each university are replicated to designated secondary campuses. This initiative was created to protect valuable research that is generated within the universities following a fire which destroyed a lot of the research at one university. The result is that by joining forces, the North American universities have implemented a cost-effective multi-site solution to their archival hosting problems. The TIMBUS consortium is not aware of any such large scale initiative in Europe between different academic or industry organisations. The alternative options are that universities in Europe perform their own archival work or else they do what industry does and outsource their requirement for secondary data locations to third parties.

### 3.11.3.2 Dual Paths

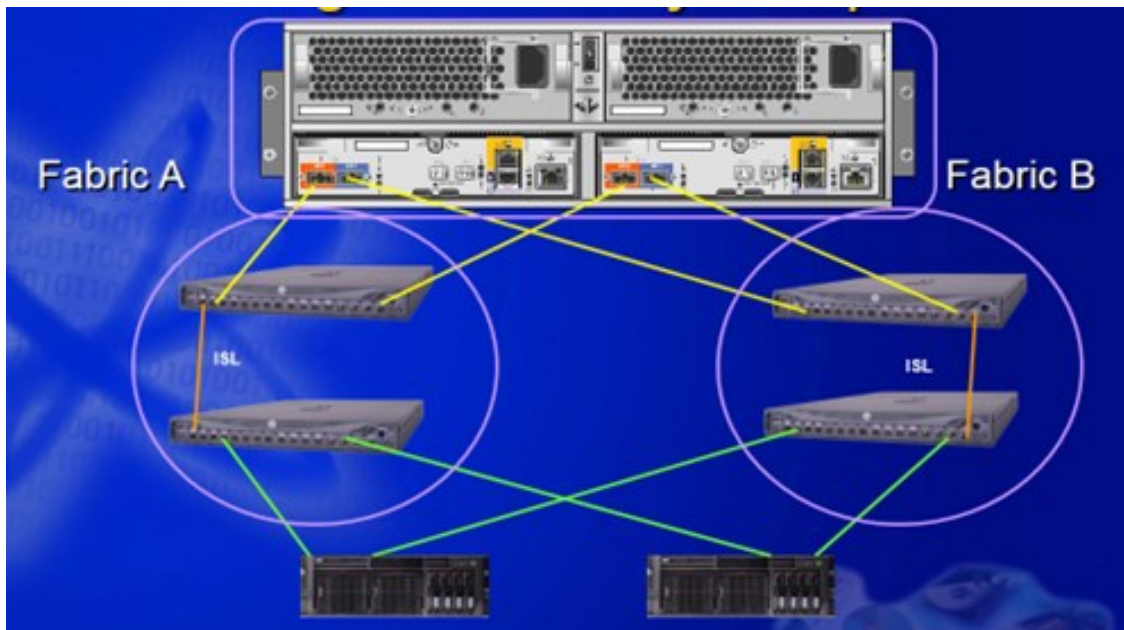
Another approach is providing high availability in the connectivity path. With this approach two host bus adaptors are used to offer dual paths to the storage network. In this case using redundant HBA's offers path failure isolation. Below is an illustration with a host with two HBA cards and with multiple connections to two different switches. This is an example of multi path.

---

<sup>31</sup> Tom Cramer, Stanford University: [https://lib.stanford.edu/files/pasig-oct2012/11-Cramer\\_DPN\\_for\\_PASIG.pdf](https://lib.stanford.edu/files/pasig-oct2012/11-Cramer_DPN_for_PASIG.pdf)



<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI



**Figure 32: Fibre Fabric Dual Paths Configuration<sup>32</sup>**

### 3.11.4 Securing the Archival Environment

Valuable business information which includes intellectual property, personal identities, and financial data is routinely processed and stored in storage arrays, which are accessed through the network. As a result, storage is now more exposed to various security threats that can potentially damage business-critical data and disrupt critical services. While it is not a large part of the scope of the TIMBUS project, nevertheless, securing storage infrastructure has become an integral component of the storage management process in traditional and virtualised data centres and it must be taken into consideration when implementing an operational archive, rather than a research prototype. Securing archives also protects against unauthorised data deletion and therefore is an intensive and necessary task, essential to managing and protecting vital information.

Storage security is the process of applying information security principles and practices within the domain of storage networking technologies. Storage security implements various kinds of safeguards or controls, in order to lessen the risk of exploitation or vulnerability in the storage network which could otherwise cause a significant impact to organisation's business. From this perspective, security is an on-going process, not static and requires continuing revalidation and modification. Storage security begins with building a framework.

The physical security of all components is an essential part of storage network security. We start with the building being a physically secure data centre with access control implemented. Controlling access to this critical hardware is important and cannot be ignored. Placing your storage in an unsecured area increases the risk of illegal access or destruction of data. Critical components such as storage arrays, switches, control stations and hosts running management applications should reside in the same data centre. With physical security implemented, only authorized users should have the ability to make physical or logical changes to

<sup>32</sup> Shane Healy Intel Ireland – Storage Architect

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

the architecture such as moving cabinets or cables. You may have access control set at the software level but in many cases hardware devices can be accessed locally.

A factor called hardening the environment relates to addressing risks such as weak passwords, default passwords, terminal server access. All default passwords on new devices need to be changed. This reduces the risk of opportunists accessing the environment with relative ease.

Risks can be external but also internal. Granting access to people who need access is the best way to ensure accountability and traceability. Under no circumstances should faceless accounts be used when data is being accessed.

It is fair to say that a fibre channel SAN is more secure than an IP based network as a FC SAN is a private environment with less nodes than an IP network. This view is probably less valid today given the growth and complexities of Storage networks.

The Fibre Channel Security Protocol (FC-SP) aligns security mechanisms and algorithms between IP and FC connections. FC-SP also addresses LUN masking and zoning, security in Fibre Channel switch port, switch-wide and fabric-wide access control, and logical partitioning of a fabric. A stronger variant of LUN masking may sometimes be offered whereby masking can be done on the basis of source FC address. It offers a mechanism to lock down the FC address of a given node port to its WWN.

Port binding limits the devices that can attach to a particular switch port and allows only the corresponding switch port to connect to a node for fabric access. Port binding does not eliminate WWPN spoofing.

Port lockdown and port lockout, restrict a switch port's type of initialisation and Persistent port disable, Prevents a switch port from being enabled even after a switch reboot.

**Table 1: Securing the Archive<sup>33</sup>**

Common Threats	Available Controls	Examples
<ul style="list-style-type: none"> <li>• Spoofing user or host identity</li> <li>• Elevation of privileges</li> <li>• Tampering with data in-flight and at rest</li> <li>• Network snooping</li> <li>• Denial of service</li> <li>• Media theft</li> </ul>	<ul style="list-style-type: none"> <li>• Strong user and host authentication and authorization</li> <li>• Access control to storage objects</li> <li>• Data encryption</li> <li>• Storage network encryption</li> </ul>	<ul style="list-style-type: none"> <li>• Multi-factor authentication</li> <li>• RBAC, DH-CHAP</li> <li>• Zoning, LUN masking</li> <li>• Storage encryption</li> <li>• IP-Sec, FC security protocol</li> <li>• Antivirus</li> <li>• Controlling physical access to data center</li> </ul>

<sup>33</sup> [www.emc.com](http://www.emc.com)

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

### 3.12 Data De-Duplication

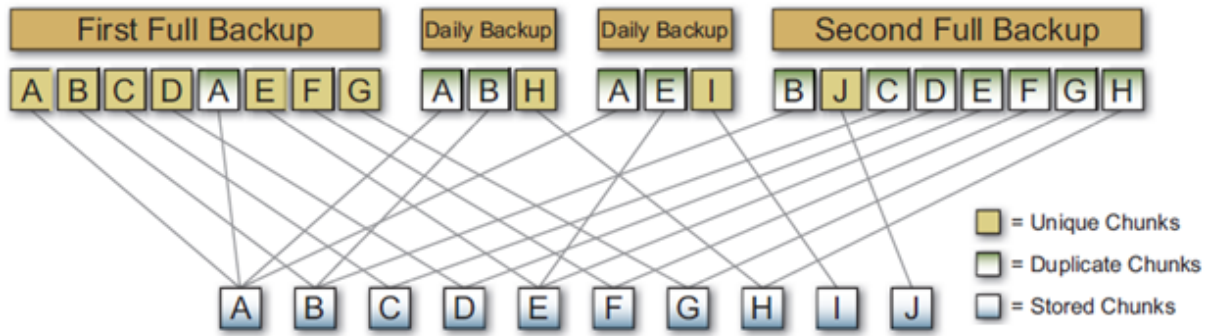
Data De-Duplication is an important optimisation that is deployed in storage architectures today. It is particularly suited to unstructured data and widely used in industry. It is not typically applied to archives today because those archives better resemble structured data systems as each digital object is likely to only exist once within the archive so there is very little to be gained in considering data De-Duplication. However, the infrastructure architecture developed in TIMBUS is designed to preserve business processes and this is a fundamental difference to the types of archives that exist today in memory institutions and academic research circles. When preserving a business process, TIMBUS will need to capture many software components and place them in a repository. These software components will include operating systems and applications. In addition, the TIMBUS solution needs to be able to preserve the business process each time the process changes. A change may be that a new version of one software package used in the business process has been deployed. The key difference therefore with an archive designed to store business processes is that it is reasonable to expect that there will be a many to many relationship between the ingested software components and different versions of different business processes in existence in the organisation over time. Most organisations naturally will try to keep IT support as simplified as possible and it is rare that an organisation will not have some standard suite of applications, or at least a common platform on which their tend to deploy their business processes. This approach is practical as the operational experience gained over time or from supporting other business processes can be applied to all business processes. An example of this would be where an organisation may prefer particular versions of Microsoft or Linux systems. It is rare that an organisation will not have some overlap and convergence in this aspect of their operations. Therefore it is likely in the data that TIMBUS will need to preserve that the same operating systems, and the same versions of applications may be used over and over and by different business processes. The repository that the TIMBUS tools will create (specifically task T6.4) will over time contain every version of every application and operating system deployed in the organisation. As new business processes start up and require preservation, only the applications and platforms that are unique and not already captured will need to be preserved. The most efficient way to cater for this would be to add some logic to the T6.4 implementation which would compare components included in a particular preservation request to those already in the TIMBUS repository and only preserve the ones that a not already in the repository. As optimisation is not a key requirement of T6.4, and certainly not one to be implemented before the basic functionality of T6.4 is in place, this requirement could instead be handled by a back-end service such as Data De-Duplication. This section of the deliverable explains more about what Data De-duplication is and what it does.

To understand how data de-duplication works and understanding of traditional backup methods is a good start. There is a high level of redundancy in the data we backup. There are many copies of the same data spread across many servers in your environment. Due to the nature of our backup rotations and retentions this data is kept in our environment for months or even years. According to SNIA de-duplication is the process of examining a data set or byte stream at the sub-file level and storing and/or sending only unique data.

If we take operating systems files are very common across multiple systems. We backup these systems every day. In some cases file versions may differ minutely from the previous version yet the entire file is backed up as there was a change to the file. Backing up redundant data increases the amount of backup storage needed and can negatively impact network bandwidth. Organisations are running out of backup window time and facing difficulties meeting recovery times due to the need to manage backup versions and a mix of backup tapes.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

*De-duplication* carves data into chunks as shown in Figure 33. These chunks are hashed and the resulting unique identifier is compared to an index of all unique identifiers. If the identifier already exists, the data is a duplicate and is replaced with a pointer to the correct unique identifier in the index. If your backup is 100GB and 20GB is redundant or duplicate data you would only store the unique 80GB. When the unique data is then compressed, you can see even more substantial disk space savings.



**Figure 33: Data De-duplication<sup>34</sup>**

*Data De-Duplication* removes any redundant data or takes unique data and backs it up to disk or tape. There are a number of approaches to Data De-Dup which are *File Level Single instance*, *Fixed Block De-Dup* and *Variable Block De-Duplication*.

**File Level** – This approach eliminates duplicated files. A pointer is placed in for the file that has been de-duped off and points to the remaining file. This approach can also be called single instance storage. This approach assesses files due for backup and compares to those already in archive or backup. An operating system is a prime example of massive file duplication. The problem here is if a file is changed even in the slightest way then the entire file is backed up again. This approach is not ideal for constantly changing environments.

**Block Level** – This approach evaluates the data stream below the file system level at the block level. As stated already the file approach has drawbacks as when a file is changed even slightly the entire file is backed up again. At the block level we have two approaches to consider, *fixed* and *variable level*.

The data stream is now broken into chunks allowing for greater de-duplication to occur. The block level approach allows us to look at the file in more detail and focus on the unique parts of the file when it has been changed. It is simply more granular. With variable block level we get even more granular and efficient by only focusing on the changed variable within a block.

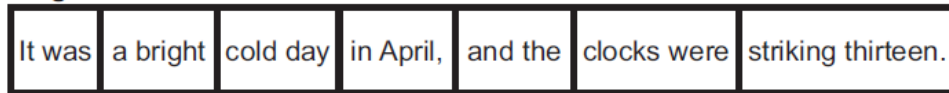
Taking a practical example of this we use an example contained in a paper produced by BakBone titled “Evaluating Deduplication Solutions” we have a sentence contained in a file. The original file is then edited with some replaced words. We can see that the fixed block approach will identify the changed blocks while the variable block with identify the actual changed instance in each block.

<sup>34</sup> [http://www.it-ps.com/assets/files/documents/BakBone\\_Evaluating\\_Deduplication\\_Solutions\\_wp.pdf](http://www.it-ps.com/assets/files/documents/BakBone_Evaluating_Deduplication_Solutions_wp.pdf)

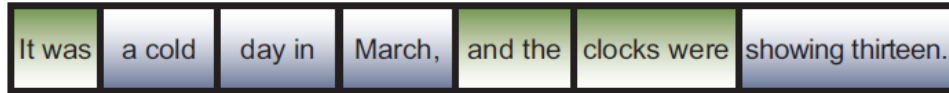
<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

“It was a bright cold day in April, and the clocks were striking thirteen.”

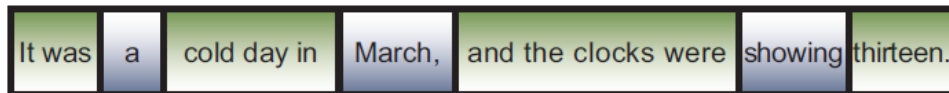
**Original File**






**Fixed Block**



**Variable Block**



-  = Duplicate Block
-  = Unique Block
-  = Original Block

**Figure 34: Data De-Duplication Methodologies Example<sup>34</sup>**

There are a number of elements to consider regarding when the de-duplication process happens and where it happens. For example if you are relying on data de-duplication to enable backups at a remote site across a limited WAN link, you don't want the de-duplication process occurring at the receiving end as it would defeat the purpose. Alternatively you may not want the de-duplication process running on a critical system which does not need the additional compute overhead.

To address these requirements there are some considerations needed. Firstly we can consider Target side or source de-duplication.

Source-side de-duplication uses a de-duplication engine that is located on the client that will perform the process of hashing the data and check for duplicates with a centrally located de-duplication index, which is typically located on the backup server or media server. Avamar from EMC is an example of this type of solution. Duplicates that are found are not transmitted across the LAN/WAN to the de-duplication storage/backup environment, but the unique blocks will be sent. The advantage of source side de-duplication is that it reduces network bandwidth because less data is sent over the network. This solves some really challenging problems for organisations with many remote offices. The standard solution would be a local backup environment at each office. Now with source-side de-duplication we can backup over the WAN.

Source-side is ideal for the remote office but there are some drawbacks. The compute is done on the client server. This can be impactful if the server is resource constraint. Also, the first backup done on the remote system will be a full system backup as everything is unique. This may lead to some long transfer times initially. You will also need to ensure that your WAN link employs some form of Quality of Server (QoS) and bandwidth utilisation to avoid saturation of the link during backup times. Quality of Service (QoS) can rate the backup traffic to junk status thus giving other more important replication traffic higher status.

Ideally Source-side is good for smaller file systems with moderate rates of change.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

Target-side de-duplication removes the de-duplication process from the client and runs the de-duplication at the target storage. The advantage here is that you do not need to worry about having clients with enough compute power to handle the hashing algorithm. With target-side de-duplication, the hashing happens at the target storage which is usually high compute power and resourced well, and then the index is checked.

If you are sending this data to the target over a congested LAN or WAN link then problems will arise. You must design your architecture with dedicated networks for this data transfer such as 10Gbe or fibre fabric connectivity.

Inline or Post Process De-duplication will occur in either inline or post process. As we outlined above at the target or source is where de-duplication can happen. Now we consider Inline which means data is de-duplicated as the de-dupe engine receives it. It is then stored on disk. This is efficient as the data is only stored after it has been reduced by de-duplication. Inline can add to the backup time, not all data is ideal for de-duplication and with inline all data is de-duplicated so this data will not be separated.

Post Process Deduplication, Using this method the data is staged prior to the de-dupe process. Obviously this means the capacity at the backup end needs to be sufficient to support the data set. In some cases the de-dupe process will start after some of the data has been copied while other solutions need all the data copied before it starts. Post Process can offer faster backups as the de-dupe process has been run before the backup starts. It is also ideal for data that does not de-duplicate well as it can be left aside during the process.

Deduplication is typically reported in a ratio, for example 12:1. The ratio is calculated as ratio = bytes in and bytes out. Ratios can be viewed as the data capacity of a system divided by its used storage capacity. If 500GB of data only consumes 50GB of storage, the de-duplication ratio is 10:1. Of course these ratios vary based on vendor calculations. Table 2 shows an example of this.

**Table 2: De-duplication Ratios**

De-duplication Ratio	Space Reduction Percentage
2:1	$1/2 = 50\%$
5:1	$4/5 = 80\%$
10:1	$9/10 = 90\%$
12:1	$11/12 = 91.67\%$
15:1	$14/15 = 93.33\%$
20:1	$19/20 = 95\%$
20:1	$29/30 = 96.67\%$

In summary, data de-duplication offers real disk space savings when it comes to reducing your backup storage. File level or single instance store will remove duplicate files in a data storage environment. Block and variable level go deeper below the file system and look at changes to the block or changes within the block. These offer really efficient savings when it comes to backup storage.

De-duplication allows you to run the de-dupe process at the client side which is ideal for remote locations with bandwidth limitations and for large systems the process can be run the storage side with high

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

compute services. De-dupe can be run before the backup called post process and while the backup is running called inline.

All of these elements leads to reduced storage, flexibility and reduced backup windows.

### 3.13 Software Components

The focus of this deliverable now shifts towards the software components of the archival architecture. This is the last section required to cover our extensive background look into archival technologies and architectures. The following illustration depicts the building blocks of the software components aligned with the previously discussed hardware components required to implement archive architectures. This section of chapter 3 will focus on archival methodologies, storage hierarchies, the concept of a storage abstraction layer, backup and recovery. Data De-duplication has been covered previously in section 3.12.

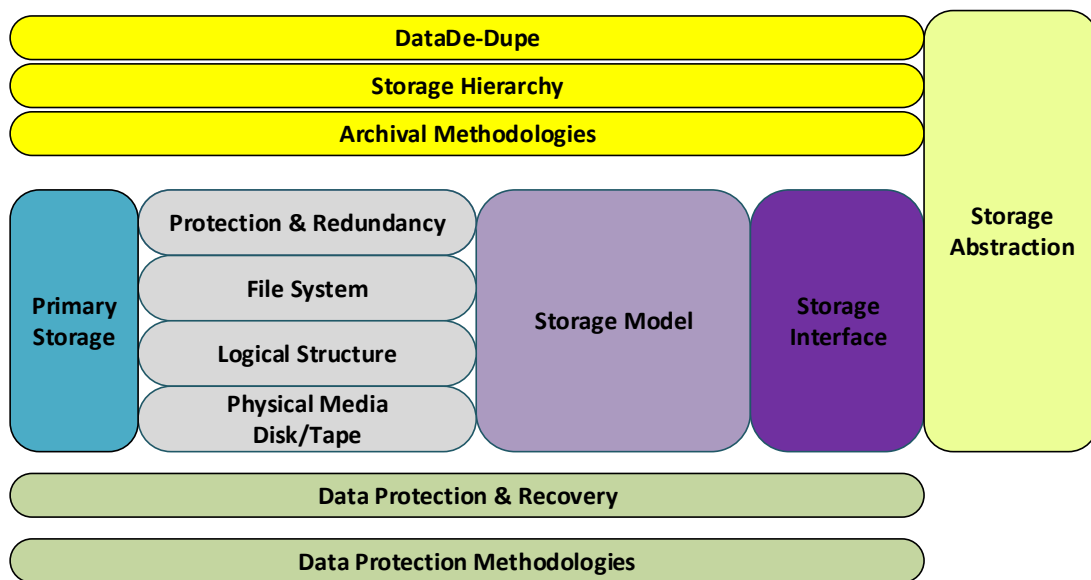


Figure 35: SW Components of the D5.3 Reference Architecture

#### 3.13.1 Archival Methodologies

As stated in the introduction of this deliverable, archiving is not a new phenomenon in data storage. Many organisations have been successfully archiving their data for many years. The TIMBUS challenge is novel because the consortium is attempting to preserve the business process environment into an archival repository in such a way that will allow it to be redeployed at any point in the future. This section of the deliverable considers some of the archival methodologies out there today and their relevance to the work of TIMBUS.

Traditional disk-based file systems such as networked-attached storage (DAS/NAS) and storage Area networks (SAN) do not have the properties desirable for archival storage. They are designed to have high performance instead of a high level of permanence, to allocate data in blocks instead of maximizing space efficiency, to read and write data instead of storing it immutably, and to provide some security but not to

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

be tamper-resistant. These storage models are on the other hand as discussed in this chapter ideal for primary based storage to which many of the business processes are executed on.

Archival data must be retained for retrieval after a period of time that exceeds the life expectancy of disk-based storage systems hardware and likely to exceed the practical lifetime of the storage system software and their interfaces. The underlying TIMBUS archive solution needs to support a Business Process Archival storage system which has the following high level objectives/capabilities:

- ✓ Significantly reduce storage cost,
- ✓ Write once, read many.
- ✓ Write once, read rarely.
- ✓ Scalable storage.
- ✓ Improved reliability (checksums, active detection, preferential replication)
- ✓ Archival storage compliance (WORM, required duration, lifecycle management).
- ✓ Lower latency than the tape systems which it replaces.
- ✓ Simple interface and design, searching capabilities.
- ✓ Accessibility across decades or centuries as well as across local or distributed systems.

### 3.13.1.1 Disk Based Archive

When discussing disk based storage, Content Addressed Storage (CAS) is our primary focus. CAS technology (CAS) is object oriented and is ideal for addressing two challenges we have with archives, long term retention and content for regulatory and legal reasons and the sheer amount of data we need to keep. The idea of an archive is this data does not change once moved off to archive, this is called fixed content. Some examples of fixed content are CAD diagrams, X-Rays and finance documents. Many of these fixed formats have legal requirements around retentions that need to be addressed by organisations. As the data grows a viable solution is needed to ensure efficient ways are applied to manage this data.

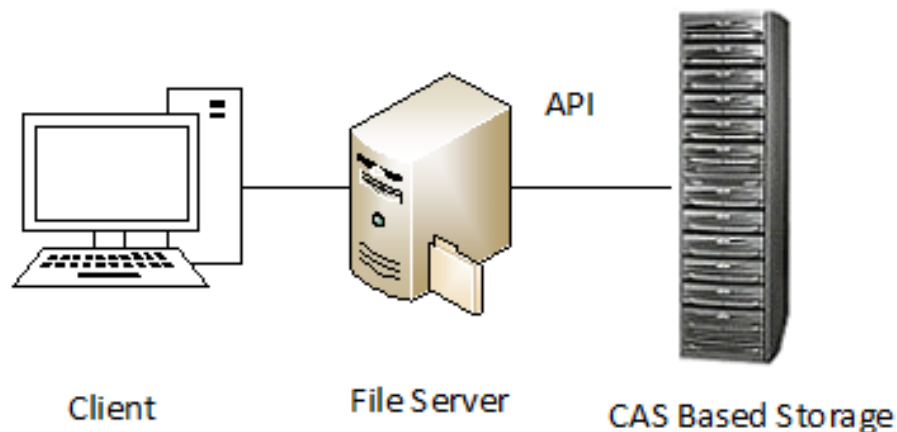
CAS based solutions often claim that tape is too old and slow to deal with this. Although we know that technologies like LTFs are changing this view point, for now implementing a CAS based storage solution offers.

A unique Content Address is created from the content. When an object has been moved to CAS for archive it receives a digital fingerprint. This is used to enable Write Once Read Many (WORM) media and to authenticate content. It protects against malicious modifications.

CAS protects the application from the physical location of the content. Single instance storage is used with CAS with multiple metadata tags which can point at the same piece of unique content. CAS offers fast retrieval compared to tapes as it maintains all content on-line. You can access your archive quickly. CAS offers technology independence which means object based systems are neutral to the storage media in use which makes it easier to migrate to new storage without disturbing the content of the archive.



<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI



**Figure 36: CAS Based Storage**

The Content Address is a unique identifier to addresses the content of a file and not its location. Content Addresses are stable and once calculated; they never change and always refer to the same content. If the content is changed then a new CA is calculated for the new content created.

*How does CAS store or archive an object?*

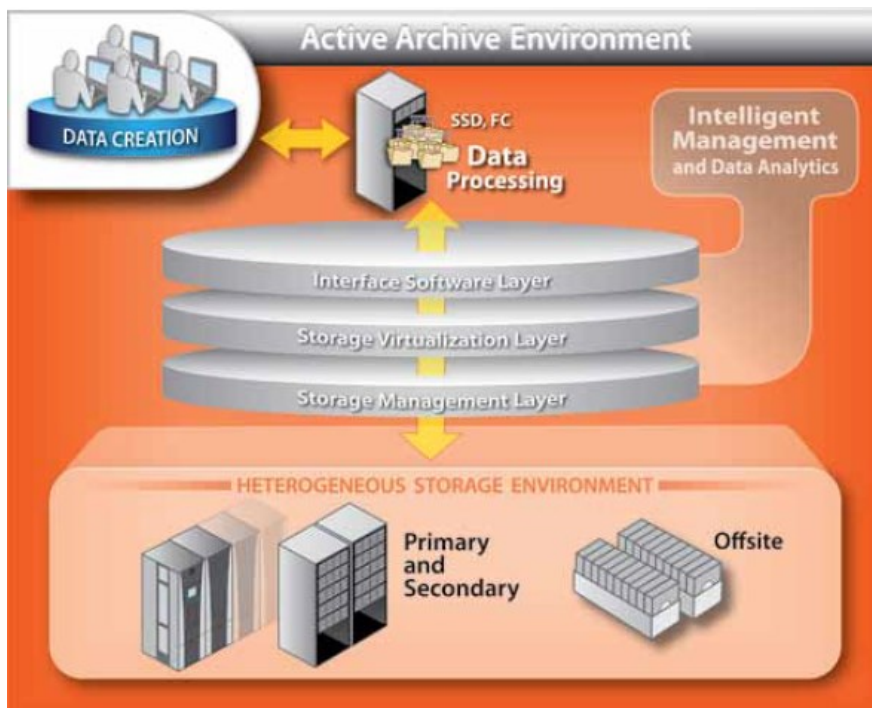
1. A user will present data at the client side to be archived.
2. A unique content address is created.
3. The new archive object is sent to the CAS storage.
4. CAS authenticates the content address and then places the object in storage.
5. The CAS will send an acknowledgement of the archive to the application that initiated the archive.
6. An Object ID is stored for future reference.

### **3.13.1.2 Active Archives**

The Active Archive Alliance launched on April 27, 2010 is a collaborative industry association formed to educate end users on the evolving new technologies that enable reliable, online and efficient access to their archived data.

With the ability to extend file systems seamlessly onto tape, active archiving is the latest solution approach that can leverage the strong suits of any storage medium. Users now have the ability to extend a file system over a myriad of different storage structures to appear as a single, logical storage volume, allowing data to reside on the most appropriate storage level.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI



**Figure 37: Active Archive Environment<sup>35</sup>**

### 3.13.2 Storage Hierarchy

The storage hierarchy relates to a process called tiered storage. If storage was free this would not exist and all our data could reside on fast efficient storage. Sadly this is not the case and we must decide on what we need to access quickly and what we can move off to cheaper storage solutions.

A more clinical answer to why we need tiered data solutions is offered by Mark Peters<sup>36</sup>. Having the right data (only) on the right storage device (always) at the right time:

- Better utilisation of all storage assets (which also means lowering floor space and power expenditure)
- Reduced management resources (despite, and because of, better alignment of data to tiers)

Tiered storage is the assignment of different categories of data to different types of storage media in order to reduce total storage cost. Categories may be based on levels of protection needed, performance requirements, frequency of use, and other considerations.

Tiered storage can be broken down into tiers 1, 2 and 3 and even more if needed. An example of Tier 1 data could be mission critical, high relevance data used regularly by the business to conduct business. You may decide to store this data on high quality storage with high levels of redundancy and performance. Tier 2

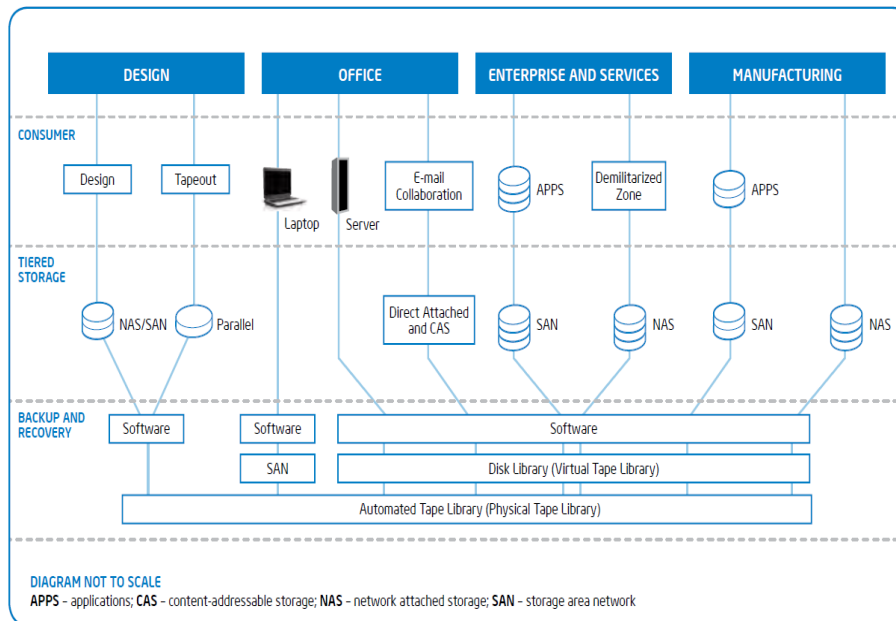
<sup>35</sup> <http://www.activearchive.com/common/pdf/ActiveArchiveOverviewBrochure.pdf>

<sup>36</sup> [http://www.esg-global.com/author/mark-peters/#cs\\_Mark%20Peters%20esg](http://www.esg-global.com/author/mark-peters/#cs_Mark%20Peters%20esg)

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

data being less critical and less used data which could be archival data is stored on less expensive, lower performance environments. Tier 3 storage may not even use redundancy to lower the costs.

A single storage solution does not always meet the business need of an organisation. Figure 38 is an illustration taken from a paper released by Intel on *Solving Intel IT's Data Storage Growth Challenges*. Intel's storage solutions differ depending on specific business applications and use cases. Some solutions apply to several business functions, while others address a specific, unique requirement of a particular business function.



**Figure 38: Intel IT's Data Storage Infrastructure<sup>37</sup>**

### 3.13.3 Storage Abstraction

Storage architectures for long term digital preservation must ensure that the current hardware model is compatible with future architectures so that an organisation can avoid vendor lock-in and we understand that future requirements in the redeployment environment may differ from what we have implemented today. As stated in section 3 of this deliverable, the redeployment environment consists of a storage ecosystem which will evolve over time and reflect current trends in storage architectures. It is reasonable therefore to assume that the back end media used in an archival solution will change over time and therefore the implementation of a storage abstraction layer can act as a common interface between the back-end media and the front-end ingest and re-deployment functions which the TIMBUS tools will perform on business processes.

<sup>37</sup> <http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/data-storage-solutions-paper.pdf>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

There are a number of potential ways to address the storage abstraction challenge such as:

- Developer chooses only APIs that have multiple independent implementations, instead of being tied into Amazon EC2 usage of Eucalyptus and Google AppEngine which are equivalent to EC2 but the challenge here is that relying on multiple implementations is risky as inconsistencies can evolve.
- A developer can choose certain API that can run on multiple clouds but not necessarily running them through the independent implementations as outlined above. The challenge here is time and cost involved in this approach and it does not always meet the desired goal.
- A separation between the application app-layer and cloud layer is deemed to be the most generic option it also requires a lot of time and is complex to maintain over time.

### 3.13.3.1 Storage Hierarchy abstraction

A paper titled “Integrated In-System Storage Architecture for High Performance Computing”<sup>38</sup> addresses the need for an abstraction layer and how one could be implemented.

CSAL stands for Cloud Storage Abstraction Layer and features *Blobs*, *Tables*, and *Queues*. It is implemented with a plug-in architecture for extensibility and currently works with Amazon EC2 and Microsoft Azure. CSAL allows an application to be portable between cloud providers and to access storage across cloud boundaries without modifying the application code. This is a significant benefit over the user-created layering as described as the third bullet above because it allows a developer to focus on application logic rather than developing and maintaining storage abstractions. It is also relevant to TIMBUS as it presents a working example of storage abstraction which is a requirement for long-term archives which can expect to experience fundamental changes in their back-end storage media over a period of decades and it is therefore desirable to limit the impact of those changes by abstracting the back-end storage from the front-end ingest and curation functions.

The paper acknowledges that there are other projects in existence today working to address the cloud abstraction challenges and states that one critical difference between the existing multi-cloud

API projects and CSAL is the idea of a unified namespace across providers and the abstraction to the application that a single cloud exists rather than a set of clouds which must be managed independently. CSAL maintains state about the namespace it manages and thus is more than an API pass through to translate calls to web services. CSAL provides the application developer with three storage abstractions: blobs, tables, and queues.

#### *Blob*

The blob abstraction is a common cloud computing storage abstraction in which data is organized into an unstructured sequence of bits that have simple get/put semantics and are intended to store potentially large amounts of data in a single object. In CSAL, blob names are only valid within a container—CSAL represents these containers only as a name. A blob’s name is only required to be unique within its container, but a single container may house many blobs. The interface to perform operations on blob containers and to get/put blobs is the *BlobStore*.

<sup>38</sup> <http://www.mcs.anl.gov/uploads/cels/papers/P2092-0512.pdf>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

### Table

CSAL’s table storage provides an abstraction for storing semi-structured data composed of sets of named attributes. A table is a named set of rows, each of which has a unique identifier called a *RowKey*. A *TableRow* is a set of *RowAttributes*, and *TableRows* within a given table need not have the same sets of *RowAttributes*. A *RowAttribute* is a name-value pair associated with a type such as: string, binary, and date. Tables are semi-structured because no schema is enforced on a table by the storage system. *TableRows* can be accessed via get/put based on *RowKey* or via a query interface that allows querying on any attribute or a set of attribute. The *TableStore* in CSAL is the primary interface to interact with tables.

### Queue

Queues abstract a first in first out storage structure with the semantics typically expected of a queue data-structure. Queues are relatively simple abstractions that provide get, put, and peek operations to retrieve messages, add messages, and view the end of the queue without altering its state respectively.

Although the above is just a quick summary of this abstraction layer solution they conclude by stating that Vendor lock-in is a difficult problem that needs to be addressed immediately because the cloud landscape is just taking shape. Standards take too long to develop, so we have presented CSAL, a vendor-agnostic abstraction layer that sits above cloud-specific APIs to provide common storage abstractions for multiple cloud platforms and to support highly portable applications even if the data itself is not as portable due to size and/or cost.

CSAL provides a set of generic storage abstractions<sup>39</sup> common to many cloud platforms. By combining a unified namespace across all supported platforms as well as high-level abstractions, CSAL allows an application to be moved with nearly no code changes and provides a foundation for multi-cloud applications that, because of mobility, can exploit the dynamic nature of the cloud landscape optimize costs and/or performance.

### 3.13.4 Data Backup and Restore Methodologies

As discussed in the hardware components section 3.2 of this deliverable, there are a number of possible methods used today for the performing archival functions. In a similar way to Data De-duplication, these are back-end services which could be quite interesting to consider their value in a TIMBUS-like infrastructure implementation whose goal is to preserve business processes rather than static data files. This section of chapter 3 is the last part of our assessment of supporting technologies available today to aid with archive management.

#### Full backup

Full backup is the starting point for all other backups and contains all the data in the folders and files that are selected to be backed up. Because the full backup stores all files and folders, frequent full backups result in faster and simpler restore operations.

It would be ideal to make full backups all the time, because they are the most comprehensive and are self-contained. However, the amount of time it takes to run full backups often prevents us from using this

<sup>39</sup> <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5708493>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

backup type. Full backups are often restricted to a weekly or monthly schedule, although the increasing speed and capacity of backup media is making overnight full backups a more realistic proposition.

Beware each full backup contains an entire copy of the data. If the backup media were to be illegally accessed, stolen or lost, the entire copy of your data could be in the hands of unauthorized persons. This is why when deciding to use a backup program to make full backups, make sure it supports encryption to protect the backed-up data.

Advantages of full backup approaches include:

1. Restore is the fastest
2. The entire backed-up data is stored in a single file

Disadvantages of full backup approaches include:

1. Backing up is the slowest compared to other backup types
2. The storage space requirements are the highest compared to incremental or differential.

### **Incremental Backup**

Incremental backup stores all files changed since the last full, differential OR incremental backup. The advantage of an incremental backup is that it takes the least time to finish. The disadvantage is that during a restore operation, each increment is processed and this could result in a lengthy restore job.

Incremental backup provides a faster method of backing up data than repeatedly running full backups. During an incremental backup, only files changed since the most recent backup are included. That is where it gets its name; each backup is an increment for a previous backup.

Advantages of incremental backup approaches include:

1. It is the fastest backup type since it only backs-up increments
2. Saves storage space compared to other types
3. Each backup increment can store a different version for a file/folder

Disadvantages of incremental backup approaches include:

1. Full restore is slow compared to other backup types because it is necessary to first perform a full backup and then apply all increments since then.
2. To restore the latest version of an individual file the increment that contains it must be found first.

### **Differential backup**

Differential backup contains all files that have changed since the last FULL backup. The advantage of a differential backup is that it shortens restore time compared to a full back up or an incremental backup. However, if you perform the differential backup too many times, the size of the differential backup might grow to be larger than the baseline full backup.

There is a significant, but sometimes confusing, distinction between differential backup and incremental backup. Whereas incremental backs up all the files modified since the last full backup, differential or

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

incremental backup, differential backup offers a middle ground by backing up all the files that have changed since the last full backup. That is where it gets its name: it backs up everything that's different since the last full backup.

Restoring a differential backup is a faster process than restoring an incremental backup because only two backup container files are needed: the latest full backup and the latest differential.

Use differentials if you have a reasonable amount of time to perform backups. The upside is that only two backup container files are needed to perform a complete restore. The downside is if you run multiple differential backups after your full backup, you're probably including some files in each differential backup that were already included in earlier differential backups, but haven't been recently modified.

Advantages of differential backup approaches include:

1. Restore is faster than restoring from incremental.
2. Backing up is faster than a full.
3. The storage space requirements are lower than for full backup

Disadvantages of differential backup approaches include:

1. Restore is slower than restoring from full backup
2. Backing up is slower than incremental backup
3. The storage space requirements are higher than for incremental backup

The above explanations of full, differential and incremental backups were sourced from [www.backup4all.com](http://www.backup4all.com).

### **Incremental Forever**

With incremental forever methodology, you only need to perform one full backup when you first set up your backup software. After this, all of your incremental uploads will be combined with the full backup once it arrives at the backup server.

This means that you only need to download a single set of backups in order to restore. This also means that your incremental backups are being appended to your most recent update instead of your original full upload. Since your last backup is always a full version, this eliminates the need for the full monthly upload.

### **Data Retention**

Once you have successfully backed up your data on tape you must ensure you can retain and access this data for the required period of time. There are many challenges to face with data retention. For most organisations keeping all our data means we can always access that file we need or that database we forgot about. The challenge as we have already stated in our introduction is the cost of maintaining primary copies of this data and now we are maintaining secondary copies and in some cases tertiary copies of this data. There are also legal and regulatory implications for keeping and not keeping data.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Table 3: Retention Periods**

Backup Type/Frequency	Retention
Daily	7-14 days
Weekly	14-30 days
Monthly	3 months
Quarterly	12 Months
Yearly	12 Months – 7 years

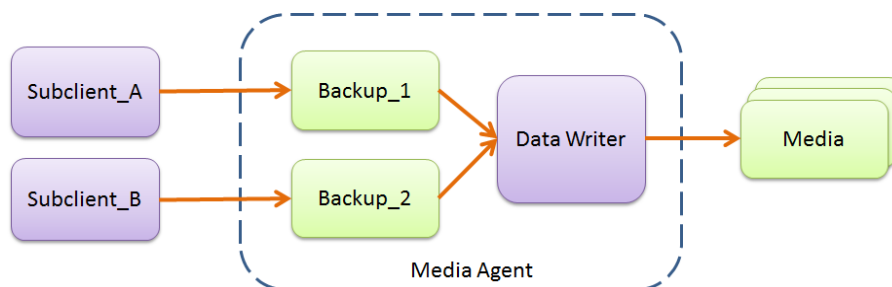
For many organisations managing a backup rotation is a real challenge. Ensuring the expensive resources purchased to manage backups are fully utilised is not an easy task. The flow of data from a source host to backup device can meet many bottlenecks along the way. There are some approaches that enable better utilisation of resources.

### Multiplexing and Multistreaming Backups

Multiplexing and Multistreaming are two mechanisms widely applied to make better use of our backup hardware. Both are opposites to each other but provide very useful functions in resource constraint backup environments.

With data multiplexing, data movers must read and write data to the same piece of media. To achieve this, these data movers are comprised of two components, data receivers and data writers. During data multiplexing, one data receiver per backup stream reads the data coming through the data pipeline. One data writer per media receives data from multiple data receivers then writes data to the media.

Figure 39 illustrates two clients, Subclient\_A and Subclient\_B, which are being backed up at the same time and their data is being multiplexed. Multiple data receivers read the data and then one data writer writes the data to a single piece of media.



**Figure 39: Data Multiplexing<sup>40</sup>**

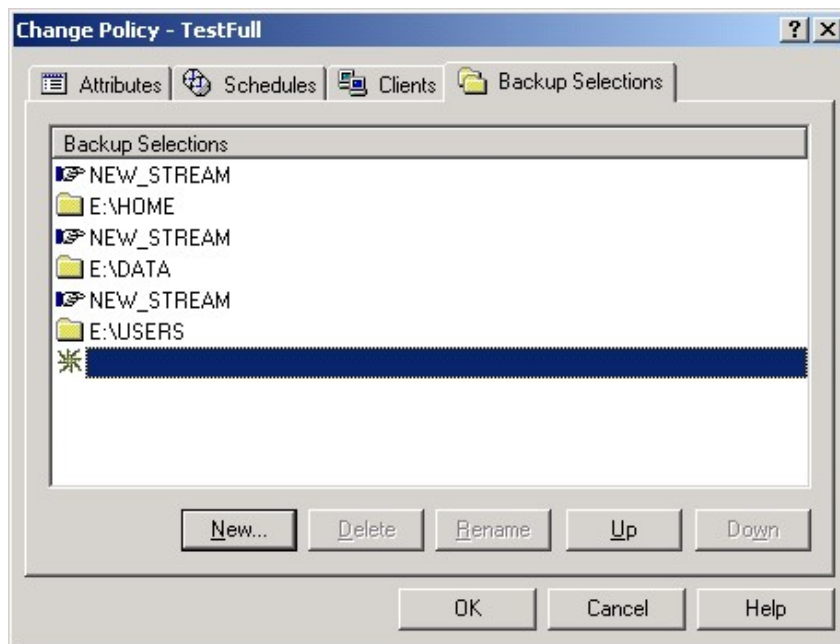
<sup>40</sup> [http://documentation.commvault.com/dell/release\\_7\\_0\\_0/books\\_online\\_1/english\\_us/images/data\\_multiplexing/data\\_multiplexing.gif](http://documentation.commvault.com/dell/release_7_0_0/books_online_1/english_us/images/data_multiplexing/data_multiplexing.gif)



<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

With data multistreaming, a client sends multiple streams of data at the same time to a data mover. Each stream of data is treated as a unique backup job. To further explain multistreaming we have to look to a standard backup job. Assume you have a Windows Server with three drives, C:\, D:\ and E:\. The standard backup would start with C:\ and progress through D:\ and E:\. This would have a specific backup duration based on the capacity. If the drives are large this backup could take time.

We could take the approach here to stream each drive letter to the data mover and tape device. Hence we would have three backup jobs running on the same server. You may also devise this to backup specific files on the same drive based on size. For example if you have a home directory on the E:\ you could backup specific folders on that drive.



**Figure 40: Backup Selections Window<sup>41</sup>**

Multiplexing is ideal for environments with limited tape resources. Multistreaming is ideal for environments with under utilisation on tape devices.

### Snapshots

There are many debates to whether a snapshot of a system is a valid backup. While storage snapshots are widely used to quickly create point-in-time virtual copies of data, they are also deemed to be valid backup solutions. This is only the case when copied to secondary storage such as a separate disk array or backup media. One possible solution in TIMBUS for capturing the source business process is to create a snap-shot of the systems which constitute it. This would ideally be done if the source business process consisted of a large proportion of virtual machines, rather than physical machines.

A snapshot represents a frozen image of a volume. The source of a snapshot is called an “original.” When a snapshot is created, it looks exactly like the original at that point in time. As changes are made to the

<sup>41</sup> [http://www.symantec.com/business/support/library/BUSINESS/ATLAS/images\\_v1/267291/multiplexing-4.jpg](http://www.symantec.com/business/support/library/BUSINESS/ATLAS/images_v1/267291/multiplexing-4.jpg)

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

original, the snapshot remains the same and looks exactly like the original at the time the snapshot was created.

The snapshot feature is very useful for performing data backups. In order to perform a consistent backup, the volume that is being backed up should not change while the backup is running. This often means the volume must be taken offline during the backup, which can be a significant inconvenience to users. With snapshots, the volume can be kept online. A snapshot of the volume is created and the backup is taken from the snapshot, while the original remains in active use.

## Replication

Replication is a necessary consideration in a long-term archive as we need to be certain that the valuable data contained within the repository is as protected as possible from loss. One way to do this is to replicate the data to another location so that more than one copy exists. Asynchronous data replication is ideal for this purpose to replicate between system instances. Once the initial mirror replica is established, only changes to index/metadata and new data segments are replicated to the target site. As a result, WAN bandwidth requirements are reduced and the amount of time to replicate data to an offsite location is reduced significantly. The need to manage tapes is now gone although tape out can still be done to complement this arrangement.

## Data Restores

There is little point in backing up data to tape if you cannot restore when needed. Sadly this fact is a major reason why many organisations lose data. A data restore can range from a single file or an entire data centre in the event of a disaster. In TIMBUS, re-deployment of the preserved business process is our equivalent to traditional data restores which happen in industry.

During a block-level backup, each backed up file is made up of a collection of blocks that define that particular file. A catalogue file is created containing a list of the backed up files, along with the individual blocks that were used for each file and the available recovery points for these files. When you need to restore a particular file, you can search your backup and select the file you want to restore and the recovery point you want to restore from. The backup solution will then collect the version of the blocks that were used for the recovery point of the specified file, and reassemble and restore the file.

We now have a better understanding of how our business process archive is protected from risks associated with data. Given the sheer amounts of data generated today much of which will move to archive, any mechanism that will reduce this data size is welcomed. One such approach is called Data De-Duplication.

### 3.14 Algorithms for data store and retrieval

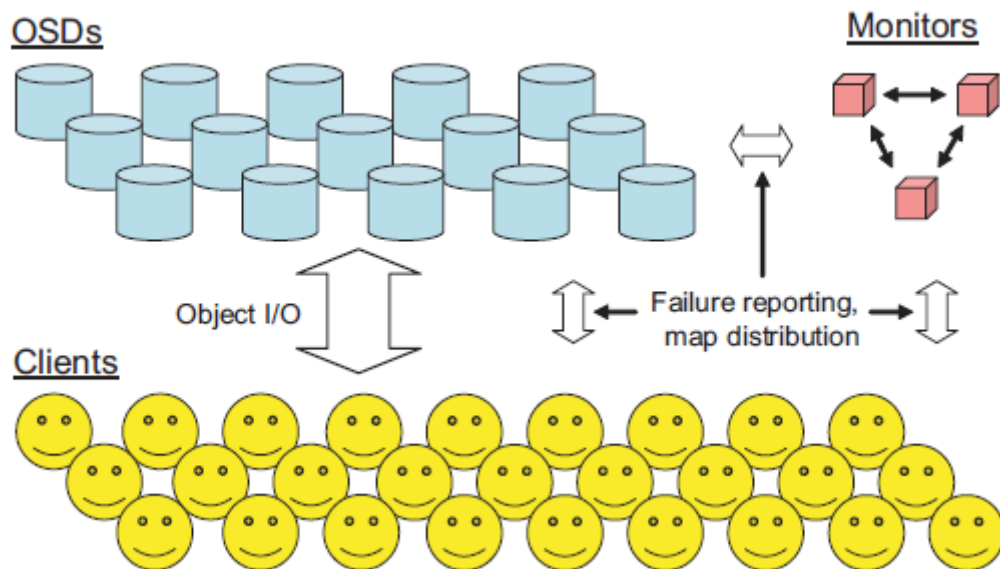
The CRUSH algorithm determines how to store and retrieve data by computing data storage locations. CRUSH empowers CEPH clients to communicate with OSDs directly rather than through a centralized server or broker. With an algorithmically determined method of storing and retrieving data, CEPH avoids a single point of failure, a performance bottleneck, and a physical limit to its scalability. CRUSH requires a map of your cluster, and uses the CRUSH map to pseudo-randomly store and retrieve data in OSDs with a uniform distribution of data across the cluster.

D5.3_M24_Architecture_For_VSRI	Dissemination Level: Public	Page 66
--------------------------------	-----------------------------	---------

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

CRUSH maps contain a list of OSDs, a list of ‘buckets’ for aggregating the devices into physical locations, and a list of rules that tell CRUSH how it should replicate data in a CEPH cluster’s pools. By reflecting the underlying physical organisation of the installation, CRUSH can model—and thereby address—potential sources of correlated device failures. Typical sources include physical proximity, a shared power source, and a shared network. By encoding this information into the cluster map, CRUSH placement policies can separate object replicas across different failure domains while still maintaining the desired distribution. For example, to address the possibility of concurrent failures, it may be desirable to ensure that data replicas are on devices using different shelves, racks, power supplies, controllers, and/or physical locations.

Figure 57 below is taken from a paper called *A Scalable, Reliable Storage Service for Petabyte-scale Storage Clusters*. A cluster of many thousands of OSDs store all objects in the system. A small, tightly coupled cluster of monitors collectively manages the cluster map that specifies cluster membership and the distribution of data. Each client exposes a simple storage interface to applications.



**Figure 41: Cluster of OSD**

### 3.15 Storage Models

The next component from the archival building blocks that we will cover from Figure 8 is storage models. Given the complexities of storage architectures, there are many resources available for use. This can lead to convoluted design and confusion over suitable storage design. This is a very important pool of knowledge for any organisation charged with implementing or maintaining a long-term archive infrastructure. To help with this, the SNIA shared storage model (described in the following section of the deliverable) intends to provide a common storage architecture outlining the advantages and disadvantages of each. According to NetApp, one of the largest suppliers of direct attached storage NAS, in 2012 companies spent up to 27% of IT budgets on data storage alone. This will clearly grow as data volume grows. The reason for this high percentage is that as data volumes grow rapidly, storage capacity is the first component to be fully consumed. The figures from NetApp relate that on average this is happening more often for storage than it

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

does for other resources such as compute power, memory or network capacity. This is worth bearing in mind when we sit down to design an archival infrastructure.

### 3.15.1 SNIA Shared Storage Model

The SNIA shared storage Model<sup>42</sup> may be used to describe common storage architectures graphically. The model exposes the services that are to be provided, showing where interoperability is required, and the advantages and disadvantages of each potential architecture model.

In essence, the SNIA model is a description of storage models but is not an actual storage architecture that you can apply to your IT business process. It is an ideal description of how a storage model can be structured.

SNIA refers to the shared storage vision as storage as a distinct resource that is best thought of separately from the server hosts that are simply consumers and beneficiaries of this back end storage. Such storage shared by multiple servers also known as hosts and managed independently from them. This is very much opposite to the older approach of direct attached storage being part of the local system such as extended disk arrays.

The growth towards shared storage provides benefits such as increased flexibility in adding and managing new storage when required i.e. organic growth. Shared storage does offer increased quality of service due to the fault tolerant and redundant nature of the architecture. Shared storage offers increased operations efficiency with a centralised support model for large end storage. The advantages of shared storage can be summarised as follows:

- **Reliability** – as required of any large, shared, critical resource
- **Scalability** – to match the size, performance, and physical and geographic placement of computing environments
- **Manageability** – to provide high levels of service, and achieve the expected reduction in operational expenses
- **Standards-based interoperation** – to avoid excessive vendor dependence in a large, critical component of data centers.

SNIA refers to the classic storage model as a mix of network storage components, with relatively small differences between them. This causes designs that are actually the same to be described in different ways; and different designs to be described sometimes using identical forms of words. SNIA shared storage model attempts to remove these problems by offering a range of different architectures enabling a clear comparison to be made around alternative architectures and designs.

Figure 42 and Figure 43, below show the SNIA shared storage model diagram which is in its second iteration. The important components of this model are from a TIMBUS perspective are:

1. The **file/record subsystem**, which includes databases and file systems.
2. The **block subsystem**, which includes both low-level storage devices, and block-based aggregation.

<sup>42</sup> [http://www.snia.org/education/storage\\_networking\\_primer/shared\\_storage\\_model](http://www.snia.org/education/storage_networking_primer/shared_storage_model)

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

3. A **management** block which provides functions such as the management of the other components.

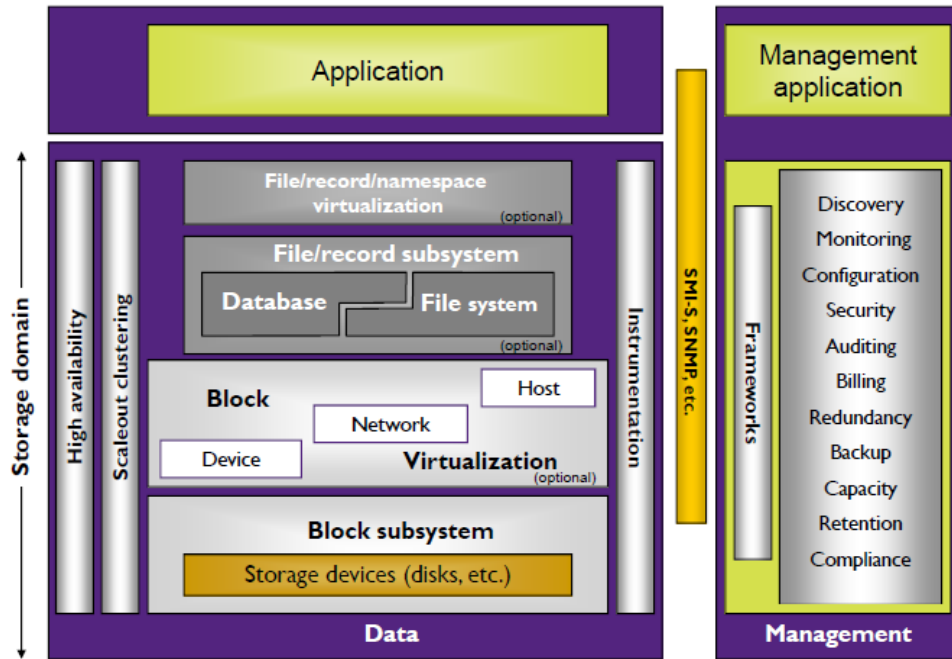


Figure 42: SNIA Shared Storage Model v2<sup>42</sup>

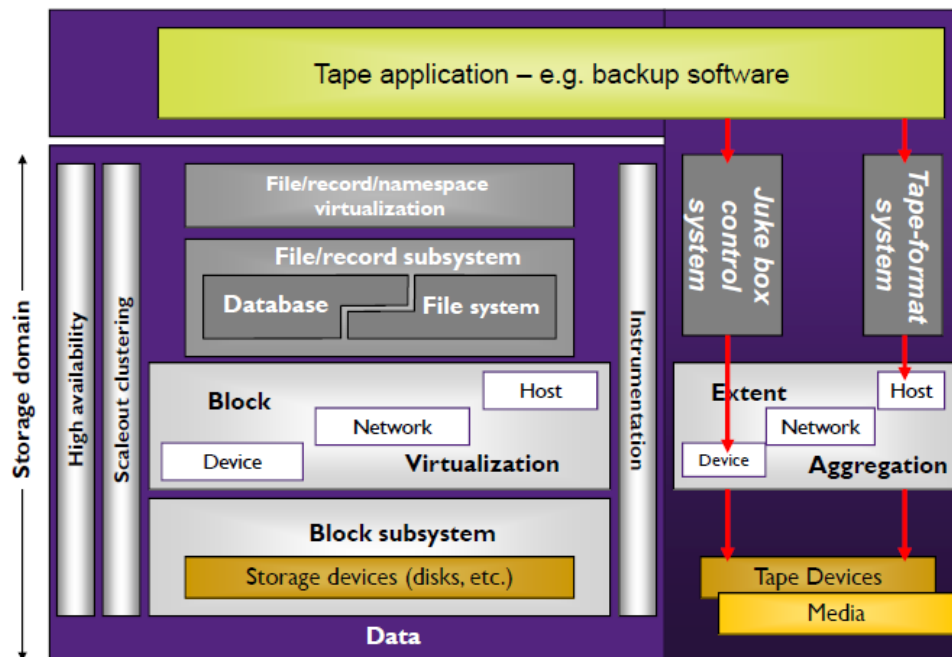


Figure 43: SNIA Shared Storage Model (for tape application)<sup>42</sup>

The SNIA share-storage model therefore is a simple and generic model for any shared storage architecture and should be consulted prior to designing a disk or tape based long-term archival solution.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

### 3.15.2 Storage Model Architectures

A host can be connected to private, non-shared storage devices e.g., disk drives attached by a local SCSI cable perhaps capable of providing a software RAID function to provide protection against disk failure. Or they can be connected over the network. This section of the deliverable explains the various options available as these are relevant if implementing a long-term storage solution if it is disk based.

#### 3.15.2.1 Direct Attached Storage (DAS)

The first of these is *direct attached* storage. This is a very common architecture used today within the industry. It is easy to conceptualise as there is no requirement for a switch between the storage and file system. Only one host can have active access (i.e. read/write) to the storage media at any particular point in time. A typical DAS configuration consists of a host such as a server directly connected to one or several hard disk drives or disk arrays. Standard interfaces are used between the disks and the servers such as SCSI, SATA or Fibre Channel.

A JBOD (Just a Bunch Of Disks) is an enclosed device with multiple disks installed off a backplane. A JBOD is a clear example of DAS direct attached storage. Direct Attached Storage is typically a disk drive local to a server or connected by a SCSI cable to the server. The server manages how the data on the disk drives is organised. Increasing the available storage space can mean either adding a server with its own storage to a network or bringing down a server and reconfiguring the disk system. JBOD's are usually connected via a SCSI interface but can also have a fibre channel interface. Internally the disks are daisy chained on a SCSI chain.

#### 3.15.2.2 Storage-network attach

This scheme introduces a storage network connecting one or more hosts to a disk array that is providing a further set of block aggregation functions. This means that the disk array resources can now be shared between multiple hosts. Figure 44 below illustrates SAN block storage.

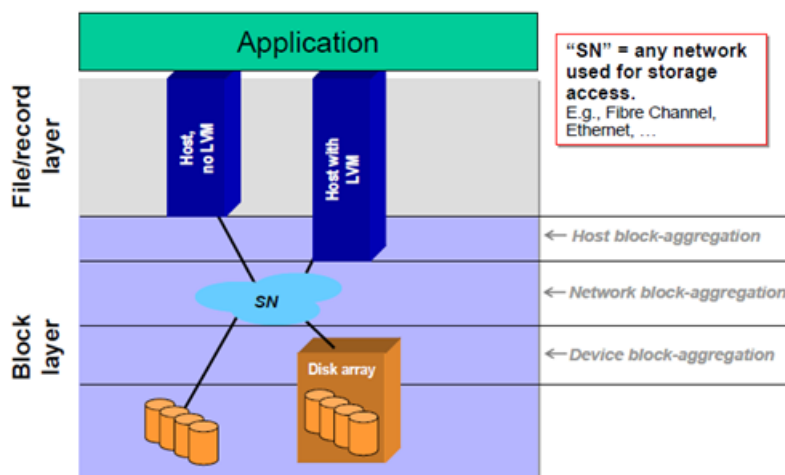


Figure 44: SAN Block Storage<sup>43</sup>

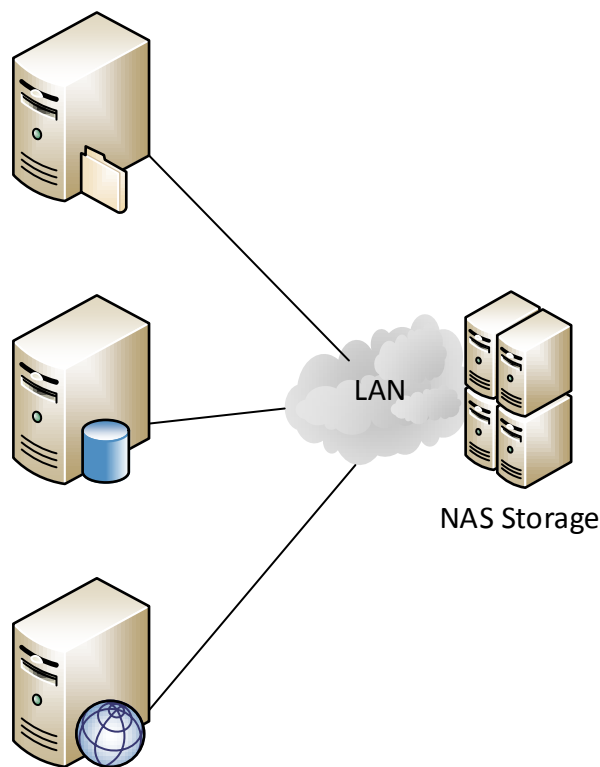
<sup>43</sup> <http://www.snia.org/sites/default/files/SNIA-SSM-text-2003-04-13.pdf>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

This section of the deliverable briefly covers Network Attached Storage (NAS) and Storage Area Networks (SAN's) which are both shared storage architectures which use some form of aggregated network device to connect the host systems (servers) to the storage media. A network switch is usually used for this. NAS is always directly attached to the local area network (LAN) managed by a network file system NFS and CIFS. Another difference between SAN and NAS is the storage model. When implementing a infrastructure to support long-term disk based storage, SAN and NAS will need to be considered.

*Network Attached Storage*

NAS is a commercial example of direct attached block storage which stands for network attached storage consisting of a standalone device called a filer. Network Attached Storage is a specialised storage that acts as a dedicated file server, set up with its own network address, rather than being attached to a network server like SAN. A NAS device consists of an operating system and a file system. The NAS can generally process only I/O requests by supporting the popular file sharing protocols CIFS and NFS. NAS enables additional storage to be quickly added by plugging it into a network hub or switch using LAN protocols such as TCP/IP over Ethernet. A NAS system will usually consist of multi-disk RAID systems. NAS is the better choice when the work consists of rapid movement of file-based, unstructured data. Figure 45 is an illustration of various front end servers connecting to a NAS over a local area network.

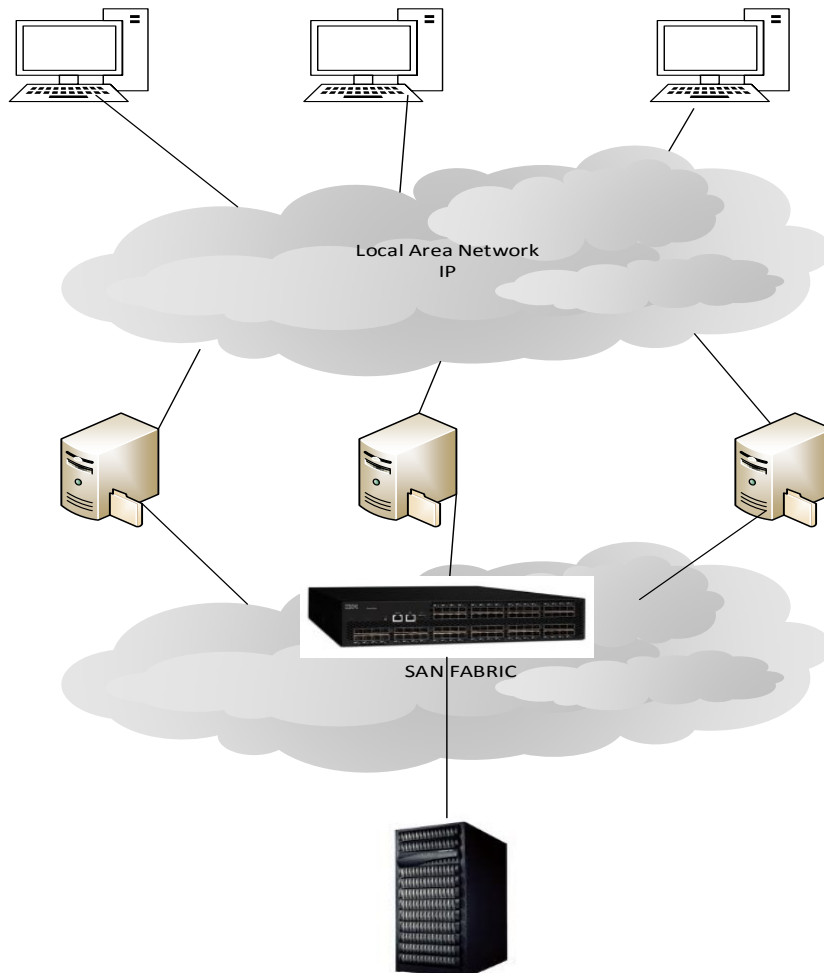


**Figure 45: NAS Storage Architecture**

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

### Storage Area Network (SAN)

A Storage Area Network is shown in Figure 46. SAN's use block oriented storage and is intended for high bandwidth block level I/O. The storage and client is usually connected via a fibre channel switch called a fabric. This fabric enables the hosts to connect to the back end storage over a high bandwidth connection. The SAN topology generally involves many hosts connecting to front end servers over the LAN. These front end servers are connected to the SAN via a fibre fabric.



**Figure 46: SAN Architecture**

#### 3.15.2.3 Tape-Based Architectures

For many organisations, tape is the primary archive solution. For tape to be used as a solution for a digital preservation archive it would need to utilise some robust tape based architectures such as automated tape library or Linear Tape File System (LTFS). LTFS has already been covered and more information is available on it appendix A.2 LTFS Editions. Instead, let's take a closer look at Automates Tape Libraries (ATL's) and discuss how large numbers of tapes are managed in an archive type solution.

Automated Tape Libraries (ATL's) are integrated hardware and software systems consisting of physical tape drives used for reading and writing data on tape. Tape libraries come in all sizes with regards to how many



<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

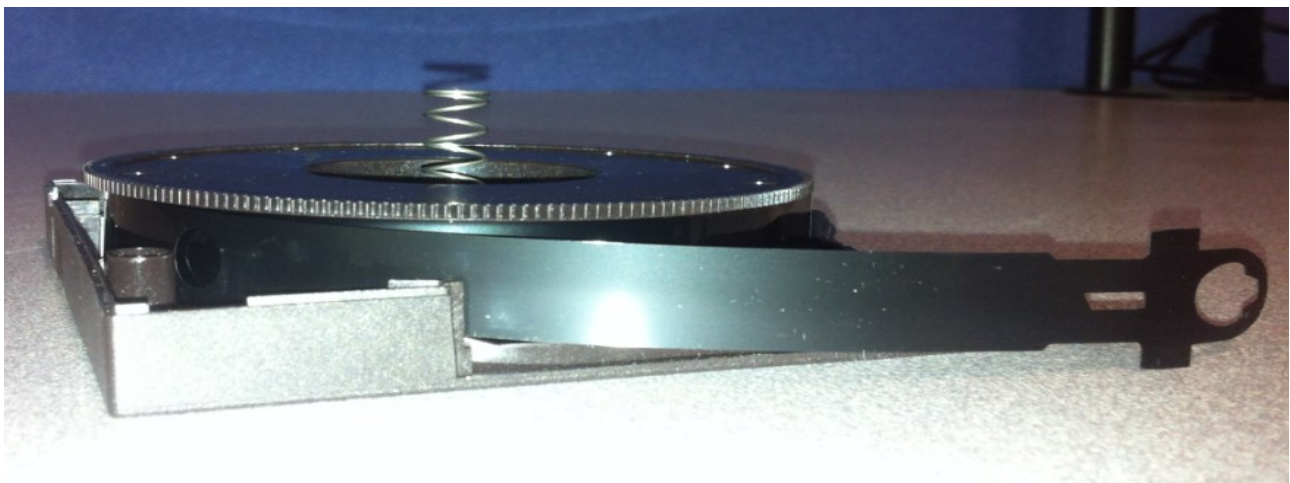
tape drives are supported – from a single drive mechanism, to hundreds of drives in the largest libraries available today. You will find the majority of tape formats available in an ATL today such as LTO, DLT (Digital Linear Tape), SDLT (Super DLT) and other vendors such as IBM and StorageTeck.

Some of the earlier ATLs used SCSI bridges to interface with external hosts but more recently ATL’s have adopted Native Fibre Channel NFC to connect directly into the storage area network removing the complexities of a SCSI bridged environment. When Fibre Channel SANs first came into being, an effort was made to enable tape drive and library owners to connect their SCSI-based backup devices to the SAN. This led to the development of SCSI-to-Fibre Channel routers that translate device mapping and addressing protocols between the two interfaces. SCSI-to-Fibre Channel routers were useful in early SAN implementations because they allowed users to experience many of the benefits of SANs without having to scrap expensive backup equipment. These bridges have proven to be problematic and complex. Native fibre channel remove the need for this router and allows the tape drive to be connected directly to the fibre switch.

A major advantage of ATL is that the tape media cartridge slots or bins, as they are often referred to. These Bins hold tapes when they’re inside the library and not in a tape drive. ATLs come in a wide variety of capacity points, from ten to tens of thousands of slots. In a busy backup environment with frequency rotations this is important to maintain capacity.

Obviously an ATL has tape media cartridges. These are the physical tape cartridges that store data and sit in the bins when they are not in use. As discussed earlier there are many tape formats used with ATL’s depending on the requirement of the customer. In most cases the same format is used for example you may have LTO-4 drives installed. In some cases the mix of drive formats is installed to support different drive formats.

The following image is taken from the inside of an SDLT tape cartridge. The leader tape in view, when inserted inside the tape drive, the tape device hooks onto this leader header and pulls the tape into the read heads on the tape drive device.



**Figure 47: SDLT (Inside View with Leader tape)**

With a large ATL with hundreds of tapes a mechanism is needed to manage and control tape locations. This is done with the use of barcodes.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

## Bar Codes

Firstly an ATL will have a barcode reader installed usually on the gripper or picker which is a robotic mechanism used to physically move tapes from the load ports to empty bins or from bins to a tape drive. Attached to this gripper or picker is a barcode reader mechanism which enables the library to read barcodes on each tape.

When the ATL is powered on it will pass through a power on self-test and then perform an inventory of each bin installed in the ATL. If a tape resides in the bin with a barcode this will be scanned and stored in the RAM of the ATL. This inventory is very important and if not done will lead to failed backups and hardware issues as the ATL gripper will not know where to locate barcodes.

Below is an example of a barcode which was manually placed onto the tape drive cassette.



Figure 48: Super DLT Tape<sup>44</sup>



HP LTO3 single bar code

Figure 49: LTO Tape Bar Code<sup>44</sup>

Barcodes are also used by backup software vendors to interact with the media management database within the backup software. Each software vendor views the barcode in different ways and it is important to

<sup>44</sup> <http://h20000.www2.hp.com/bc/docs/support/SupportManual/c00517361/c00517361.pdf>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

understand this. The full bar code up to 16 characters needs to be read and then a six-character media ID is created. The media ID defaults to the last six characters of the bar code unless a media generation rule is set to choose the desired characters for the media ID.

Tapes are added to the ATL via Cartridge access ports or load ports. This is designed to prevent disturbance to the operating library. You can add and remove tapes while backups are running. It also reduces the need to open the doors which can lead to foreign particles entering the library.

An ATL can be expanded using a pass through mechanism. This is a clever way to increase capacity and expand instead of having to scrap and buy a new environment. This is achieved using pass through ports.

The next part is the brains of the ATL. The Library control system. This is the hardware and software “glue” that helps the slots, drives, pickers, caps and everything else in the library work as a cohesive model. The control system also manages the external connectivity to the ATL using autoloader SCSI commands or through vendor specific API. The key to an ATL is it’s interaction with third party backup software which avails of the automation of the tape management.

All ATLs will have a control panel on the front of the library to perform manual actions such as power on/off, tape moves and inventory steps. Prominent ATL vendors include IBM, Sun, HP, SpectraLogic, Quantum, and ADIC. Some of the prominent backup’s software vendors are EMC/Legato NetWorker, Veritas NetBackup and Backup Exec, IBM Tivoli Storage Manager, CommVault, BakBone, and many others.

Figure 50 shows an image of inside an ATL with a gripper and barcode reader scanning some tape cassettes loaded in a bin.



**Figure 50: Inside an Automated Tape Library**<sup>45</sup>

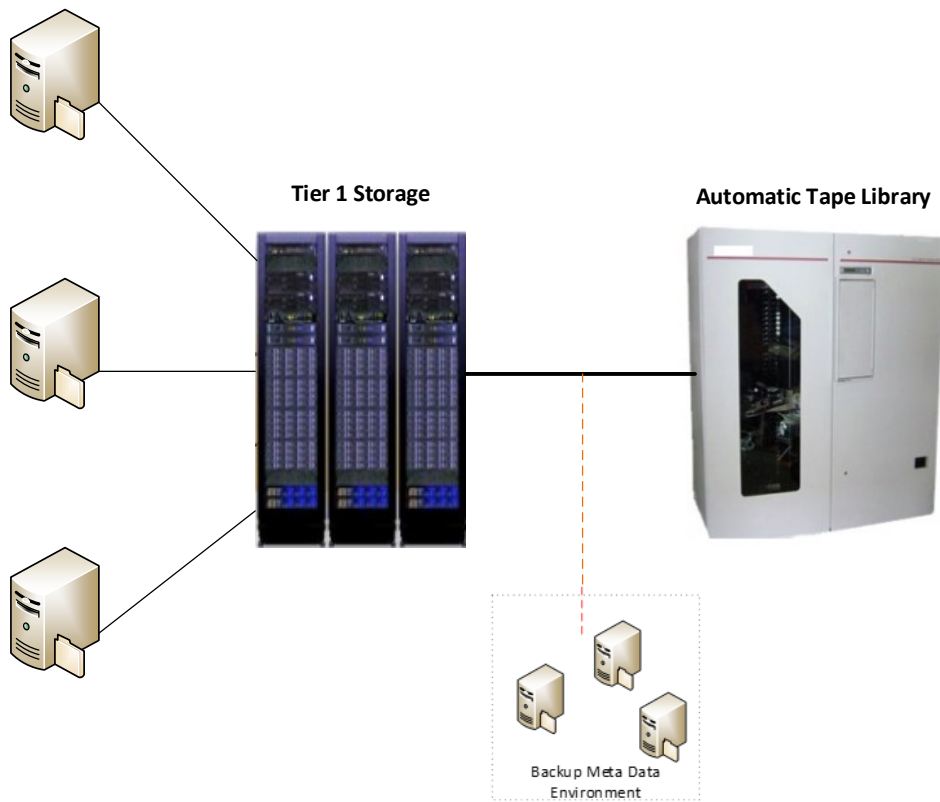
ATLs still have a strong case in today’s data protection World. For example Family Search runs the largest scale preservation project in the world that TIMBUS is aware of. Family Search has been archiving genealogy records from around the world since the late 1800’s. Today, these records are stored digitally on tape, one copy of which is kept in vaults which have been blasted out of Granite Mountain overlooking Salt Lake City in the United States. Their data typically consists of images which are digitally scanned copies of

<sup>45</sup> <http://www.ibm.com/common/ssi/GIF/ALET/3584XXXX.GIF>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

original documents. The documents can be from a large variety of sources, for example census reports, births, deaths, marriage and prison records are typical. They cannot ever afford to lose a record. In some cases, the digital objects they store are the only surviving records today as the originals may have been lost either accidentally through wars, floods, fires, mismanagement or deliberately where someone at a certain point consciously decided to destroy the records for some reason.

The sheer size of data being written each day amounts to 15TB per day. The relevance of all this to TIMBUS is the sheer scale at which Family Search operate and the infrastructure that they use to do so. Family Search writes an incredible 15TB of data to tape each day and have a requirement to increase this to a staggering 50TB by 2015 to support the integration of 6 petabyte of records which have previously been stored in two other cities up to now. The data is primarily in image formats so it must be indexed manually at some point to make it searchable. Tape systems have a very high I/O throughput so much so that Family Search is not able to write the data quickly enough to keep them fully busy. Therefore disk drives are used as a temporary staging buffer for the data from where it can then be spooled to tape 24 hours a day, 365 days per year. Family Search employs a storage abstraction layer to allow for media changes over time but they insist that tape is still the best media known today for long-term archival. Their reasons for this are cost, reliability, storage density, high throughput and the resiliency of tape to environmental conditions as compared with disks. Figure 51 is an illustration of a tape based archive solution tied into our hardware flow as outlined in section 5.1.

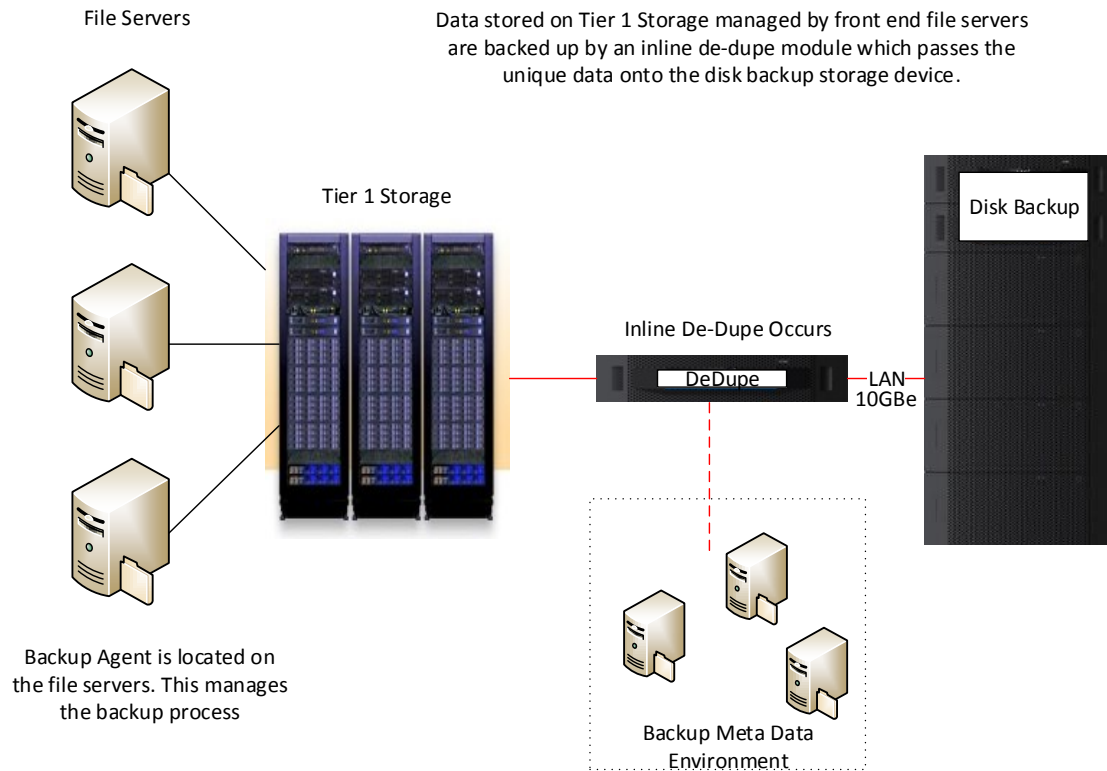


**Figure 51: Tape-Based Archive Solution**

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

### 3.15.2.4 Disk-Based Architectures

The business needs today are constantly evolving and require fast, robust archival storage models. Unstructured data now comprises a massive portion of large data growth, and hierarchical file systems are difficult to optimize and scale. Below is an illustration of a disk based Storage Architecture incorporating inline de-dupe, with a disk based backup and archive solution.



**Figure 52: Disk-Based Archive Solution with Data De-Duplication**

Disk-based backups and archiving incorporating data de-duplication allows for ten to thirty times less data than the standard full and differential rotations to be stored on disk. With this approach you can leave data online for longer which is ideal for file restores and online archive access.

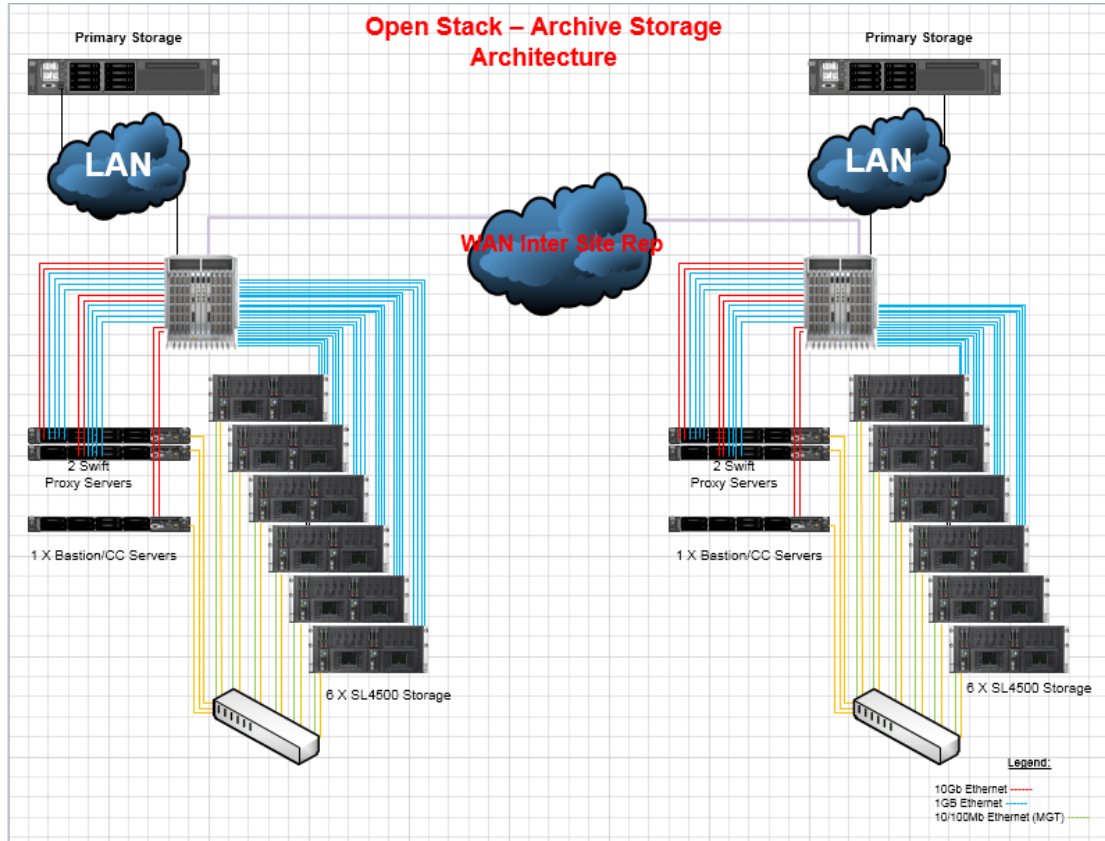
An additional step here would be to incorporate replication. With disk-based backup solutions the desire to move away from tape outs which involve moving periodic full backups to tape and sending to an offsite location is addressed by implementing a replication strategy between sites outside of the data centre. Similarly with the disk based archive we can replicate content between locations to avoid SPOFs, Single Points of Failure. As we discussed in Chapter 3, asynchronous data replication is ideal for this purpose to replicate between system instances.

### 3.15.2.5 Openstack Storage Architecture Example

The following storage architecture detailed in Figure 53 Openstack based architecture which is ideally suited to long-term preservation. Openstack is an open source project for cloud manageability. The architecture incorporates two sites connected via a WAN link to offer redundancy and business continuity.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

The primary storage is connected to secondary storage using the Swift object-oriented storage model incorporating back end disk arrays connected over a dedicated Ethernet connection. A proxy server is used to handle all user requests and communicates to underlying storage. A bastion server is used to authenticate users and administration access.



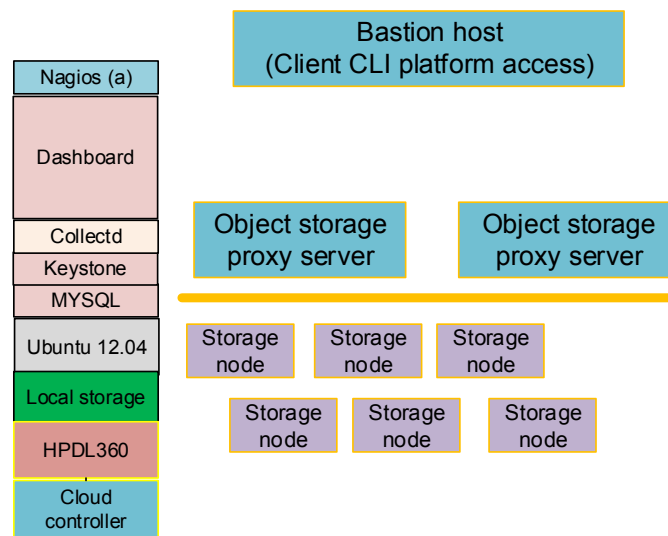
**Figure 53: Openstack – Archive Storage**

Based on many of the technologies discussed within this document we can identify some key technologies which are ideal for an open Cloud Object Based Swift storage solution.

Object-based storage as shown below in Figure 54, offers many valuable opportunities for TIMBUS and long term preservation architectures. As we stated in Chapter 4, requirements, reliability, flexibility, cost and interoperability are key requirements for long term preservation architecture. Swift and CEPH storage are capable of meeting these requirements along with the other requirements set out in Chapter four.



<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI



**Figure 54: Object Based Swift storage Solution**

Swift and CEPH with object-based storage enable an archive to use distributed storage with replication to maintain the redundancy of the data. Data is managed in chunks as opposed to blocks. These chunks can be set to a variable size. With CEPH based environments an interpreter is used to sit on top of object based storage presenting as block. This is what was previously referred to as a RADOS block device, offering even more flexibility in the storage environment.

The object-based environment is easily increased by simply adding nodes where and when needed making the object-based storage scalable. Expense is reduced as Swift is disk independent not requiring a specific vendor or disk type. Many implementations of Swift with object-based storage use low to midrange SATA disks making the hardware environment affordable. Many storage models require expensive SCSI or SSD based disks to ensure performance and reliability.

RAID is not needed with object-based storage; recovery is handled by having multiple copies of the same data. In the event of a disk failure the data is rebalanced across the new disk. The number of copies maintained is set by the storage administrator.

Flexibility is offered through the ability to manage disk space quotas for new shares when needed. As new containers are created a logical quote limit is assigned. In summary, object-based storage offers some real opportunities for long term preservation environments.

### 3.16 Summary

It is clear that TIMBUS has a number of potential archival strategies which could be used. We could, for example, follow the tape path. This is a proven technology which was designed for archival and backup data and with the development of LTFS we could treat disk just like any other file system and move data back and forth. We could also focus on the disk solutions which offer content addressed storage with online archive access making access to our archive data quicker. This is an implementation specific detail which may come down to a cost decision. The architecture developed in TIMBUS deliverable D5.3 supports both.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

## 4 System Requirements

This section of the deliverable considers the specific requirements of infrastructures required to support long-term archival. Citing external sources and leaning on the expertise that the D5.3 partners bring to task T5.3, a list of specific requirements for the TIMBUS architecture was developed.

### 4.1 Infrastructure Considerations

The TIMBUS exploitation report (deliverable D2.2) expands on the list of drivers specifically for long-term preservation rather than identifying drivers for modernisation of the archive infrastructure. This section of the deliverable relates the factors that were taken into consideration when developing the requirements for D5.3.

A 2009 NEC whitepaper<sup>46</sup> on modernising archive infrastructure listed cost avoidance, hardware obsolescence and maximising the organisations ability to respond to litigation risks as major drivers for archive infrastructure investment.

#### *Cost Avoidance:*

The SNIA 100 year survey stated that 60-80% of all data, not just archive data is static. This is a well understood phenomena in data management and is a driver behind the tiered approach to data storage as less frequently accessed data is migrated automatically over time to lower, cheaper but slower to access storage tiers and ultimately perhaps to tape if required to long term-archival. Newer technologies provide other benefits as we have seen with the comprehensive review of hardware and software components in section 3 of this deliverable so a process of periodically migrating the underlying infrastructure will bring efficiencies which should produce a positive cost return-on-investment.

#### *Hardware Obsolescence:*

A second reason for archival infrastructure refresh is to avoid issues of hardware obsolescence. Hardware will fail over time and as hardware vendors no longer support the devices once they are out of warranty, consumers need to have assurance that if they hardware fails they will be able to source replacement parts to keep their archive online.

#### *Ability to Respond to Risks:*

The NEC paper spoke specifically about litigation risks, but TIMBUS expands this to also include business re-use. The ability to rapidly access the data in your archives can be a very powerful asset to an organisation.

Figure 55, below shows some of the components of the NEC approach, highlighting some of the storage attributes which cover resiliency and redundancy.

<sup>46</sup> <http://www.necam.com/Docs/?id=e93c05ba-ce09-47cb-aa54-68d440849be0>



TIMBUS	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

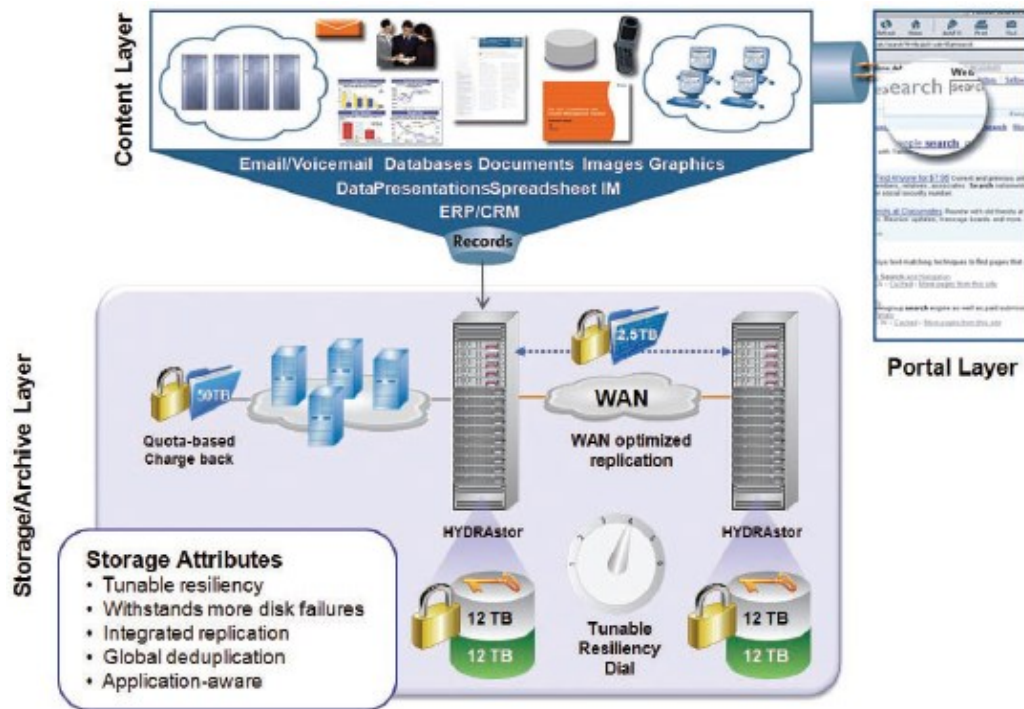
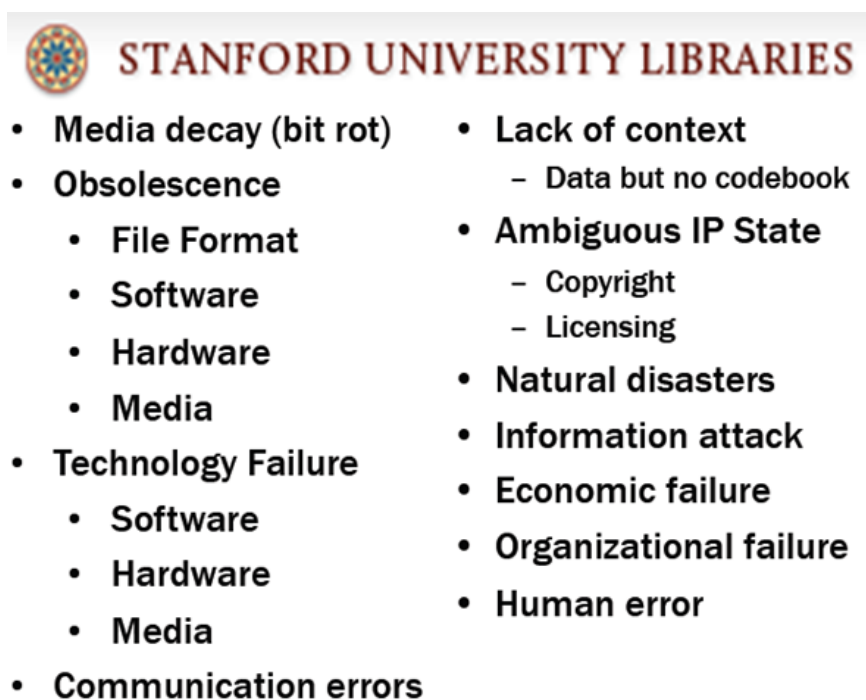


Figure 55: Scalable, Resilient archive features as defined in NEC paper<sup>46</sup>

## 4.2 Risks to Digital Preservation

The TIMBUS project is working to address many areas where there are risks to digital information preservation. Stanford University produced a cohesive listing of these which is shown below in Figure 56. Task T5.3 has the responsibility to address some of these. In the case of the architecture which is presented in this deliverable, T5.3 can help with many of these in particular media decay, hardware and media obsolesce, technology failures, communication errors, natural disasters and human error. It is the objective of the design of a long-term archival infrastructure to address these items. Many of them will turn out to be solvable with the correct architecture design but the ideal design, or at least the one that gets implemented, can sometimes be compromised by cost.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI



**Figure 56: Stanford University Risks to Digital Information**

To tackle these risks, T5.3 employs several strategies as follows in sections 4.2.1 to 4.2.6.

#### **4.2.1 Replication and Data Integrity (Fixity)**

The architecture promotes the concept of replication to avoid single points of failure. Data in the archive should be replicated to multiple locations. The architecture can help ensure the integrity of the data contained within it through the implementation of checksums when reading and writing data and through multiple copies of data. Checksums can be compared to ensure that the data has not changed or become corrupted. If corruption is detected, the ideal preservation system should be intelligent enough to detect it and recover from a second copy of the data so that at least 2 known good copies are always available.

#### **4.2.2 Migration of Media**

Processes need to be in place to cycle media and keep them up to date over time. In the case of tape, the data should be migrated to new tapes every 5 years to ensure that the media does not degrade or become obsolete. This is a curation function which must be implemented as part of the business process which is responsible for preservation.

#### **4.2.3 Migration of Technology**

The architecture assumes that technological advances will make the underlying infrastructure obsolete. It is not possible to design an architecture that will itself not change as new infrastructure capabilities become available but it is possible to design one which allows for reasonable change. Bearing in mind that all the

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

technology components will change over time, the curator of a preservation process should have a plan for how each component of their preservation infrastructure could be swapped out for a more advanced and capable piece of hardware that might become available. This is catered for through the addition of the storage abstraction layer. Keeping the architecture as simple as possible is therefore a good objective.

#### **4.2.4 Emulation**

The architecture needs to support emulation in the re-deployment environment as this is the preferred method in TIMBUS for re-running a preserved business process. This can be done by provisioning a small compute environment when required to host the re-deployment activity.

#### **4.2.5 Sustainability**

The architecture should be technically feasible. It should not contain concepts which require the development of new technologies not commercially available today. The novelty aspect of D5.3 is in how it uses existing technologies and applies them to solve a problem in the preservation of business processes. The architecture needs to be implementable, that means that the scope of D5.3 should not require the invention of new file systems, storage media, models or interfaces. The work carried out in task T5.3 has not discovered a need for such development as is not already happening in the industry.

#### **4.2.6 Redundancy and Heterogeneity**

Redundancy is built in through the adoption of all the technologies detailed in chapter 3 of this deliverable. The architecture is vendor and hardware agnostic. Those implementation details are for the consideration of an organisation using the architecture to implement a long-term archival infrastructure. In TIMBUS, that responsibility falls to task T6.7, the preservation test-bed.

### **4.3 Functional Requirements**

A digital preservation storage architecture aims to provide a storage based sub system of hardware and software components to ensure that business processes and associated data is captured and stored, accessible at any time in the future, is secure, efficient, reliable and protected.

#### **4.3.1 Physical Storage Management**

The archive environment contains the following functional requirements to enable the archive to exist, perform and grow as required by the business. The archive environment should address vendor specific constraints and technology dependencies.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Table 4: Flexibility**

<b>ID: FR1.1</b>	<b>Name: Flexibility</b>
<b>Description:</b>	The architecture must be flexible enough so that it can meet the needs of a small office or SME to a large enterprise.
<b>Additional Information:</b>	Not all components of the architecture will be required in every use case.

**Table 5: Reliability**

<b>ID: FR1.2</b>	<b>Name: Reliability</b>
<b>Description:</b>	The architecture should not contain single points of failure.
<b>Additional Information:</b>	The preservation environment cannot tolerate single points of failure. All disks, interfaces and channels must offer redundancy where needed to avoid data loss due to single points of failure.  For more information on reliability refer to sections 3.6 and 3.11.3 of this deliverable.

**Table 6: Storage Interoperability**

<b>ID: FR1.3</b>	<b>Name: Storage Interoperability</b>
<b>Description:</b>	The architecture must comprehend the challenges of underlying hardware advances over time
<b>Additional Information:</b>	The business process archive must be able to withstand future changes in the hardware subsystem. External access to the storage subsystem must prevent vendor lock in and hardware specific constraints. The implementation of a hardware storage abstraction layer to provide transparent data access.  More information on Storage Interoperability is available in section 3.9.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Table 7: Data Migration**

<b>ID: FR1.4</b>	<b>Name: Data Migration</b>
<b>Description:</b>	The business process archive must be able to support the ability to migrate data between environments internally and externally through the use of supported interfaces.
<b>Additional Information:</b>	For more information on Storage Interfaces see section 3.9.

**Table 8: Zoning**

<b>ID:FR1.5</b>	<b>Name: Zoning</b>
<b>Description:</b>	In the case of SAN implementations, physical access ports, such as HBA's must be zoned to access only its assigned fibre port. A HBA must be rezoned each time it is replaced or changes LUN assignments.
<b>Additional Information:</b>	For more information on Zoning see section 3.11.2

#### **4.3.2 Archive Capacity & Performance**

The time it takes to access archived data will depend on the type of medium used, be it tape or disk, the interface used to connect to it from the outside world and the host system. Traditionally archives were always seen as slow access repositories located on remote storage such as tape. This traditional approach has changed considerably given the growth and usage of business archives. A business process archive will need to redeploy data from varying timeframes within a predefined time period based on business requirements.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Table 9: Performance**

<b>ID:FR2.1</b>	<b>Name: IO Performance</b>
<b>Description:</b>	The storage infrastructure should implement optimisations which meet acceptable performance for access latency while balancing that with capital investment.
<b>Additional Information:</b>	<p>The preservation archive should be housed on a storage medium that meets the business requirements. Archives can be online with rapid access or infrequently accessed data. A tiered based storage model can be implemented to avoid housing less accessed data on expensive disk. See section 3.13.2.</p> <p>The business process archive must meet require access times as outlined by business. Disk Access times to data will vary on the physical disk type and disk interface. Online archives and offline archives will also determine access performance. See section 3.4.1.1.</p>

**Table 10: Network Capacity**

<b>ID:FR2.2</b>	<b>Name: Network Capacity</b>
<b>Description:</b>	The network bandwidth internally in the archive (node-to-node) should be sufficient to not cause failure of the basic fundamentals of the TIMBUS tools. The external uplink to the internet (inbound and outbound traffic) should be sufficient to allow remote access for the purpose of tool testing.
<b>Additional Information:</b>	<p>This is enabled by the following:</p> <ul style="list-style-type: none"> <li>• Internal capacity. The archive should be expected to support 100Mbps or higher if required.</li> <li>• Externally, the test-bed should have sufficient capacity not to cause undue performance degradation. This consideration is constrained by its nature and is generally difficult and costly to upgrade.</li> </ul> <p>No requirement exists for high inbound/outbound network capacity. High data loads (that is: preservation requests and re-deployment requests) will be carried on the internal, higher bandwidth network.</p> <p>For more information on network capacity see section 3.11.</p>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Table 11: Primary Storage**

<b>ID:FR2.3</b>	<b>Name: Primary Storage</b>
<b>Description:</b>	The archive architecture must be capable of meeting the high compute demands of a digital archive preservation environment
<b>Additional Information:</b>	<p>Front-end compute devices and back-end controllers must have sufficient Random Access Memory, cache and processor architecture to meet the high compute demands of a long term preservation environment. Services such as de-duplication and data compression add a compute load to archival workflows which must be comprehended.</p> <p>Intel Processor Comparison<sup>47</sup></p>

**Table 12: Scalability**

<b>ID:FR2.4</b>	<b>Name: Scalable Capacity</b>
<b>Description:</b>	The architecture should permit headroom to enable organic growth.
<b>Additional Information:</b>	<p>The first element of a scalable capacity strategy is easy expandability, so that as the organisation and/or amount of data increases, you can expand storage space to meet your needs without interfering with the ability of users to access previously stored data.</p> <p>Additional information on this is available in section 3.15 of this deliverable which covers storage models.</p> <p>Industry recommendation range from 20-40% headroom at any given time. The headroom of a set of resources is equal to the total capacity of the resources minus a margin. Margin is a portion of the total capacity not to be used, so as to allow for acceptable response times. An initial margin may be selected using a graph plotting total resource utilisation level percentage against response time. For a desired or acceptable response time factor, the initial margin is selected as 100% minus the utilisation percentage that yields the selected response time factor.<sup>48</sup></p>

<sup>47</sup> <http://www.intel.com/content/www/us/en/processor-comparison/compare-intel-processors.html>

<sup>48</sup> <http://www.freepatentsonline.com/7467291.html>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

### 4.3.3 Archive Security

The function requirements relate to how we secure the archive environment from the actual data storage subsystem, file system, storage interfaces, physical storage frames, data centre access and authorisation of end users.

**Table 13: Location Suitability**

<b>ID:FR3.1</b>	<b>Name: Location Suitability</b>
<b>Description:</b>	The geographical location of the physical storage environment(s) must consider access constraints and risks of natural disasters
<b>Additional Information:</b>	The business process archive should be located in a suitable location with access to high speed local area networks, wide area internet access, raised flooring, appropriate cooling and lighting. Awareness of natural risks such as flooding, earthquakes, high temperatures and cold temperatures should be considered

**Table 14: Physical Storage Port Security**

<b>ID:FR3.2</b>	<b>Name: Physical Interface Access</b>
<b>Description:</b>	Ability to access the physical data frame/storage subsystem via local ports such as Ethernet, SCSI.
<b>Additional Information:</b>	Implement lockdown policies on local ports directly accessible from the physical data storage.  Host Bus Adapters, for example, must be hard coded with the World Wide Name which is unique. HBA can only be connected to a node to which it has been predefined.



<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Table 15: Remote Interface Access**

<b>ID:FR3.3</b>	<b>Name: Remote Interface Access</b>
<b>Description:</b>	Storage interfaces provide access into the data disks/tapes via an interface from the remote system.
<b>Additional Information:</b>	For more information on Securing the Archive see section 3.11.4.

**Table 16: Remote Access**

<b>ID:FR3.4</b>	<b>Name: Storage Frame Remote Access</b>
<b>Description:</b>	Administrative access to the SAN frames and fibre Switches may only be allowed via Terminal Server over a secure network.
<b>Additional Information:</b>	See section 3.15.2.2.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Table 17: Authorised Access**

<b>ID:FR3.5</b>	<b>Name: Authorised Physical Access</b>
<b>Description:</b>	Physical access to the storage frames must be controlled to allow on permitted access on a required basis. Prevention of accidental hardware configuration changes, human error, malicious activity can be prevented by controlling physical access to the data frame.
<b>Additional Information:</b>	<p><b>Electronic Access Control Systems (ACS):</b> Access to all entry points into and within the data centre should be protected by electronic access control mechanisms which allow only authorized individuals to enter the facility. Included within the framework of electronic access control should also be biometric safeguards, such as palm readers, iris recognition, and fingerprint readers.</p> <p><b>Off-boarding Process:</b> Personnel working for the data centre or clients utilising the facility services must be immediately removed from systems that have allowed access to the facility itself. This includes all electronic access control mechanism along with removal of all systems, databases, Web portals, or any other type of sign-in mechanism that requires authentication and authorization activities.</p> <p><b>Visitors:</b> All visitors must be properly identified with a current, valid form of identification and must be given a temporary facility badge allowing access to certain areas within the data centre. This process must be documented in a ticketing system also.</p> <p><b>Alarms:</b> All exterior doors and sensitive areas within the facility must be hard wired with alarms.</p> <p><b>Cameras:</b> The facility should have a mixture of security cameras in place throughout all critical areas, both inside and out, of the data centre. This should include the following cameras: Fixed and pan, tilt, and zoom (PTZ) cameras.</p>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Table 18: Secure Data Transportation**

<b>ID:FR3.6</b>	<b>Name: Secure Data Transportation</b>
<b>Description:</b>	In the event that archival data needs to be transported outside the confines of a secure network and secure data storage frame then it must be maintained in a secure and safe manner.
<b>Additional Information:</b>	<ul style="list-style-type: none"> <li>• Stored in fire proof boxing.</li> <li>• Electrostatic protection to be used when handling disk drives.</li> <li>• Avoid extreme temperatures during transportation.</li> <li>• Avoid any impacts during transportation.</li> <li>• Adequate labelling of devices to ensure identification of devices such as bar code labelling or disk sequence labelling.</li> <li>• Use of an approved certified vendor for transportation.</li> <li>• Identification and signature of all parties with access to data devices during transit.</li> </ul>

#### 4.3.4 Archive Protection

A major part of any archive management is ensuring the data stored within the archive is protected with a comprehensive backup and recovery environment. A key requirement for data archives is to routinely backup, store, replicate and recover archived data to ensure the end to end recovery and protection environment is working.

**Table 19: Archive Protection**

<b>ID:FR4.1</b>	<b>Name: Archive Protection</b>
<b>Description:</b>	Providing a backup and restoration service for all archived data.
<b>Additional Information:</b>	For more information on Backup & Recovery see section 3.10.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Table 20: Offsite Protection**

<b>ID:FR4.2</b>	<b>Name: Archive Offsite/Redundant Protection</b>
<b>Description:</b>	Alongside providing a backup and recovery service, the archive must be stored in a redundant state at an alternative location to provide business continuity in the event of failure.
<b>Additional Information:</b>	For more information on Backup & Recovery see section 3.10.

#### 4.3.5 Archive Supportability

The Archive architecture must allow interoperability with all standard hardware and software systems in use today.

**Table 21: Operating Systems**

<b>ID:FR5.1</b>	<b>Name: Support for Operating Systems</b>
<b>Description:</b>	Interoperable with all major operating systems and file systems.
<b>Additional Information:</b>	The preservation environment should be capable of running any required operating system as part of the source and re-deployment environments. The source and re-deployment environments host the business processes which are at the heart of the preservation and re-deployment requests.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Table 22: File Systems**

<b>ID:FR5.2</b>	<b>Name: Support for File Systems</b>
<b>Description:</b>	Interoperable with all standard file systems
<b>Additional Information:</b>	<p>The preservation environment should be capable of identifying and interacting with all major file systems in use today.</p> <ul style="list-style-type: none"> <li>• FAT,</li> <li>• FAT12, 16, 32</li> <li>• HFS,</li> <li>• MFS (Macintosh File System),</li> <li>• UFS (Unix and Linux file system),</li> <li>• ext2, ext3, ext4,</li> <li>• JFS,</li> <li>• XFS,</li> <li>• OS/2,</li> <li>• NTFS,</li> <li>• Mac OS X,</li> <li>• HFS</li> <li>• ZFS</li> </ul> <p>For more information on File systems see section 3.8.1.</p>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Table 23: Hardware Systems**

<b>ID:FR5.3</b>	<b>Name: Support for Hardware Industry Standards</b>
<b>Description:</b>	Interoperable with all major hardware formats.
<b>Additional Information:</b>	<ul style="list-style-type: none"> <li>• SATA, eSATA, eSATAp</li> <li>• Solid State Drive (Flash)</li> <li>• ATA.</li> <li>• Fibre Channel.</li> <li>• SCSI.</li> <li>• iSCSI.</li> <li>• Infiniband.</li> <li>• Ethernet.</li> </ul> <p>For more information on Hardware Interfaces see section 3.9.</p>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

## 5 Infrastructure Architecture for Business Process Preservation

This chapter presents the TIMBUS reference architecture for long-term archival infrastructures. As stated in the introduction chapter, the architecture is closely related to other parts of the TIMBUS project, particularly task T6.7 which is an implementation of the architecture but it also supports the other TIMBUS architecture components.

There are a number of considerations that drive the design of long-term preservation storage architectures. In chapter 3, we have discussed the various hardware and software components that make up an archival architecture such as disk or tape, storage logical structures, file systems, hardware redundancy and storage interfaces and network connectivity. In this chapter we use these building block components to present the TIMBUS architecture for the preservation and re-deployment of business processes.

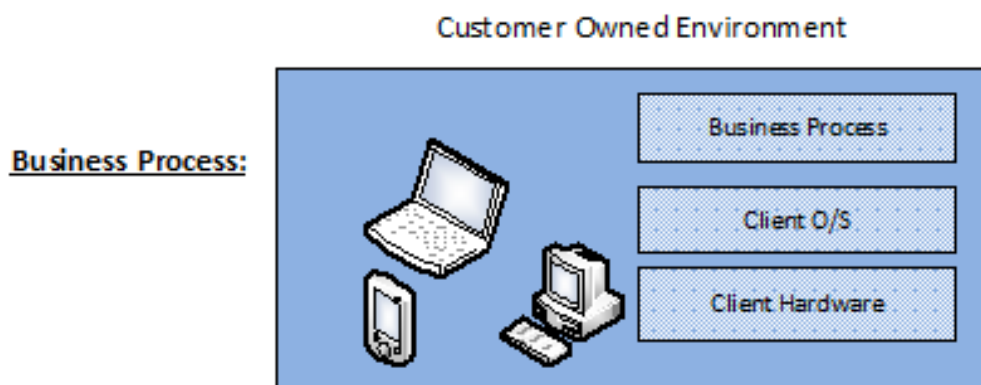
We begin with a clear understanding of the data flow through the various hardware modules that exist in a layered data storage environment. Taking our business process digital object and following its path from the client end system to the archive will allow us to understand how our long-term preservation environment is structured. This document therefore introduces each part of the architecture first, explaining its role and function before adding it to the next step in the flow. The complete architecture is presented in Figure 64 at the end of this chapter.

### 5.1 The Business Process Hardware Architecture Flow

#### 5.1.1 Business Process

This section refers to the input of a business process. A collection of host compute devices such as desktops, laptops, smartphones and virtualised environments with associated operating systems, GUI (Graphical User Interface) or command line based systems. This process enables end users to interact with the compute environment by inputting data which creates new objects or modifies existing objects.

This process may also be the final stage of the compute cycle with output data presented to the host console in the form of complete data or information.



**Figure 57: Technical Components constituting business process**

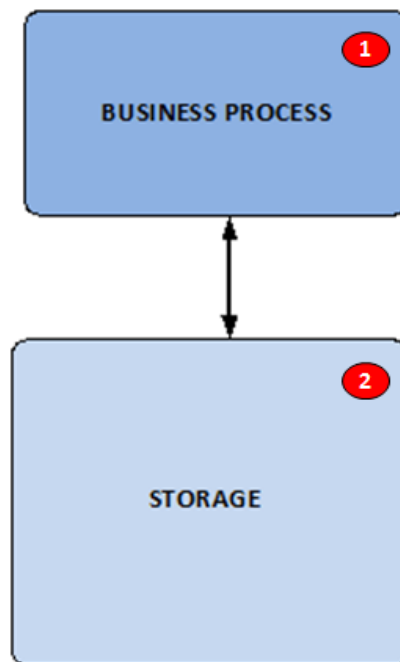
<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

### 5.1.2 Primary Storage

Traditionally primary storage refers to the processing environment of a compute device alongside the memory and fast memory called cache. Given the sheer speed of most applications it is not possible to commit each write directly to disk as even the latest hard disks on offer today cannot write at the speeds issued by the multi CPU's. The primary storage component in the TIMBUS infrastructure architecture offers fast access memory to store write information until the disk is ready to commit. It is important to state clearly that the term primary storage refers to processing and memory although it is often referred to as the storage environment that data resides on which is connected to a PC or a server. That environment is known as the secondary storage.

### 5.1.3 Secondary Storage

Sharing a close relationship with primary storage, the secondary storage component of the TIMBUS infrastructure architecture completes the input output I/O cycle between primary fast storage and secondary storage. There are many types of secondary storage available to host computers such as hard disk drives, optical drives and tape devices. Although we have differentiated primary and secondary storage, it is common to refer to both as the same environment, i.e. "storage".



**Figure 58: Client/Server Storage**

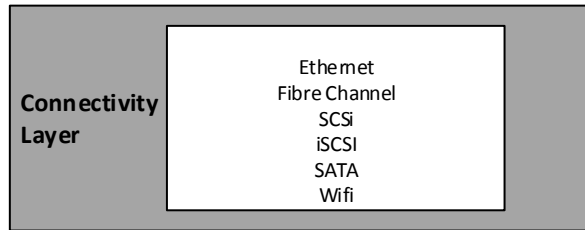
### 5.1.4 Network and Storage Interface

To interconnect each module in the TIMBUS storage architecture, a form of connectivity is required. Connectivity can range from Ethernet over a local area network or wide area network sharing multi path communications. Interconnectivity may also be local via a SCSI cable, USB cable or over fibre channel to enable high bandwidth connectivity. To enable these interconnections a suitable interface is needed. For Ethernet a network interface card or host bus adapter is located in the compute host. Other suitable



<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

interfaces are SATA, iSCSI, Infinband and 10G Ethernet. These were covered in detail in chapter 3 of this deliverable. The connectivity layer is given in Figure 59 below and is indicated in the overall architecture by the connecting arrows.



**Figure 59: Connectivity Methods**

### 5.1.5 Backup Meta-Database & Backup Hardware Infrastructure

The Backup Meta-Database module is concerned with protection of data and recoverability of data. This is not to be confused with the metadata extractors listed in the D5.5 architecture and under development in WP6 tasks T6.2 and T6.5 which are concerned with meta-data which describes the constituent components of a business process and their interdependencies. There are two main sections to the Meta-Database component outlined in Figure 60, the meta-data collection contained in the catalogue database which tracks all media usage and locations along with data, retention levels and scheduling parameters. Our backup meta-data environment will manage the transition of data between the host and the backup storage device.

The second section of this component is the actual Backup Hardware Architecture. This module focuses on tape based backups and disk based backups. This model enables primary data to be moved off storage and protected on either tape, automated tape library or disk based backup devices.

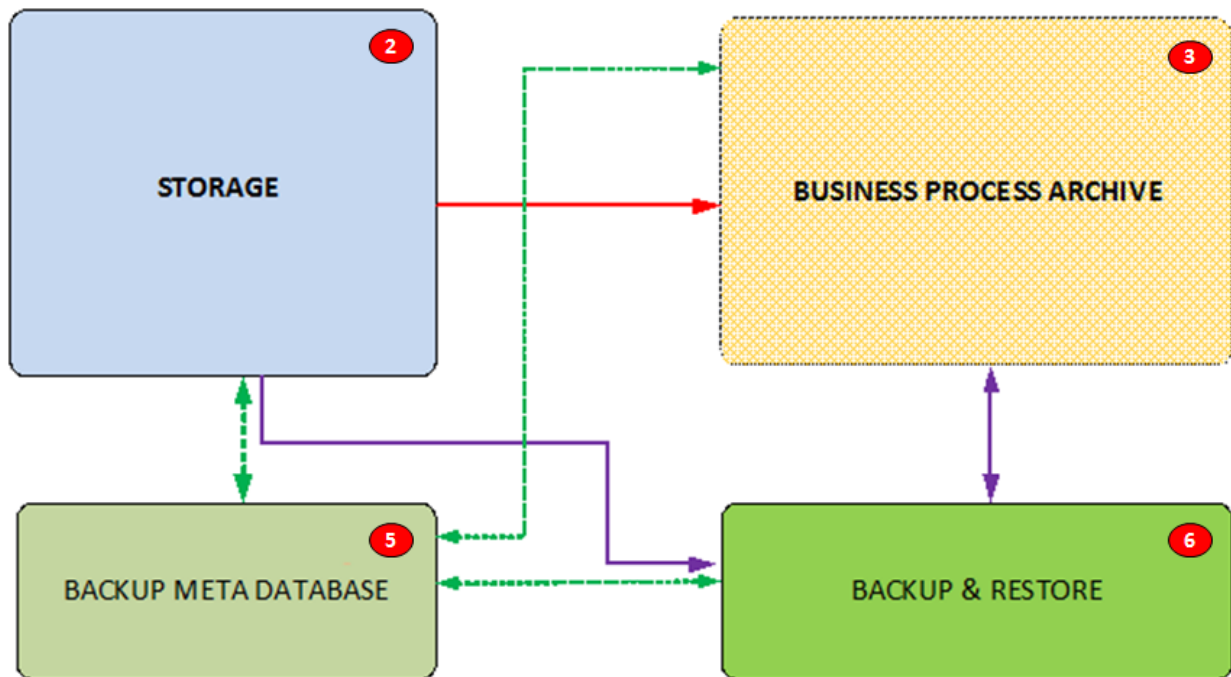


**Figure 60: Archive Protection Environment**

### 5.1.6 Business Process Archive Architecture

The Business Process Archive component enables suitable objects to be stored on an appropriate archive-based storage solution which offers direct, secure and protected capabilities. Once an object has been identified as required for archive, the object is moved off tier 1 storage to a lesser tiered architecture which primary objective is to store this object without any amendments to it and provide an indexed search capability to enable access to the object at a future date. Archives can be disk-based or tape-based.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

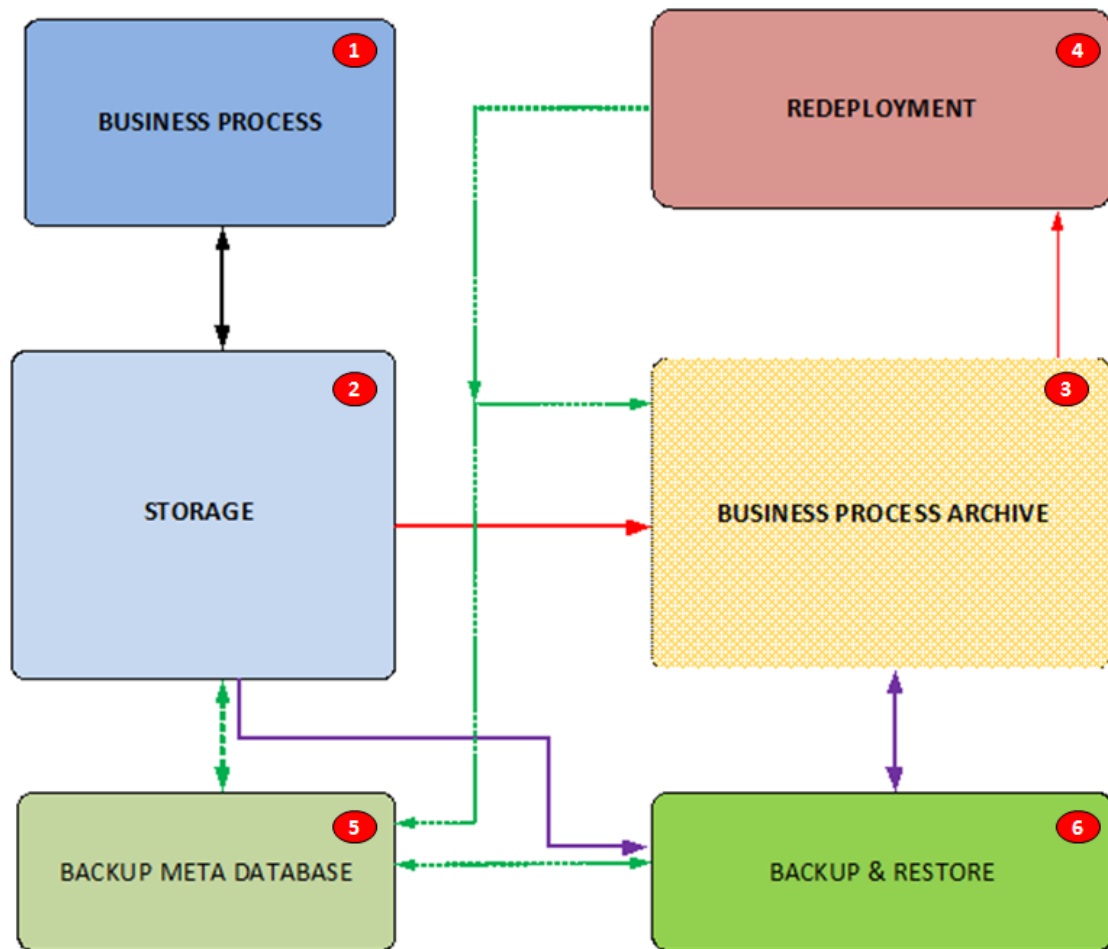


**Figure 61: The Business Process Archive**

### 5.1.7 Redeployment Architecture

The redeployment architecture enables the business process environment to be redeployed at any stage in the future. This environment has a direct relationship with the Backup Meta Data environment to understand the archive locations via the catalogue. The Redeployment module must also have a direct communication link and interface with the Business Process Archive. To ensure the storage interface between the Redeployment module and Business Process module are maintained over time a Storage Abstraction Layer is implemented.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

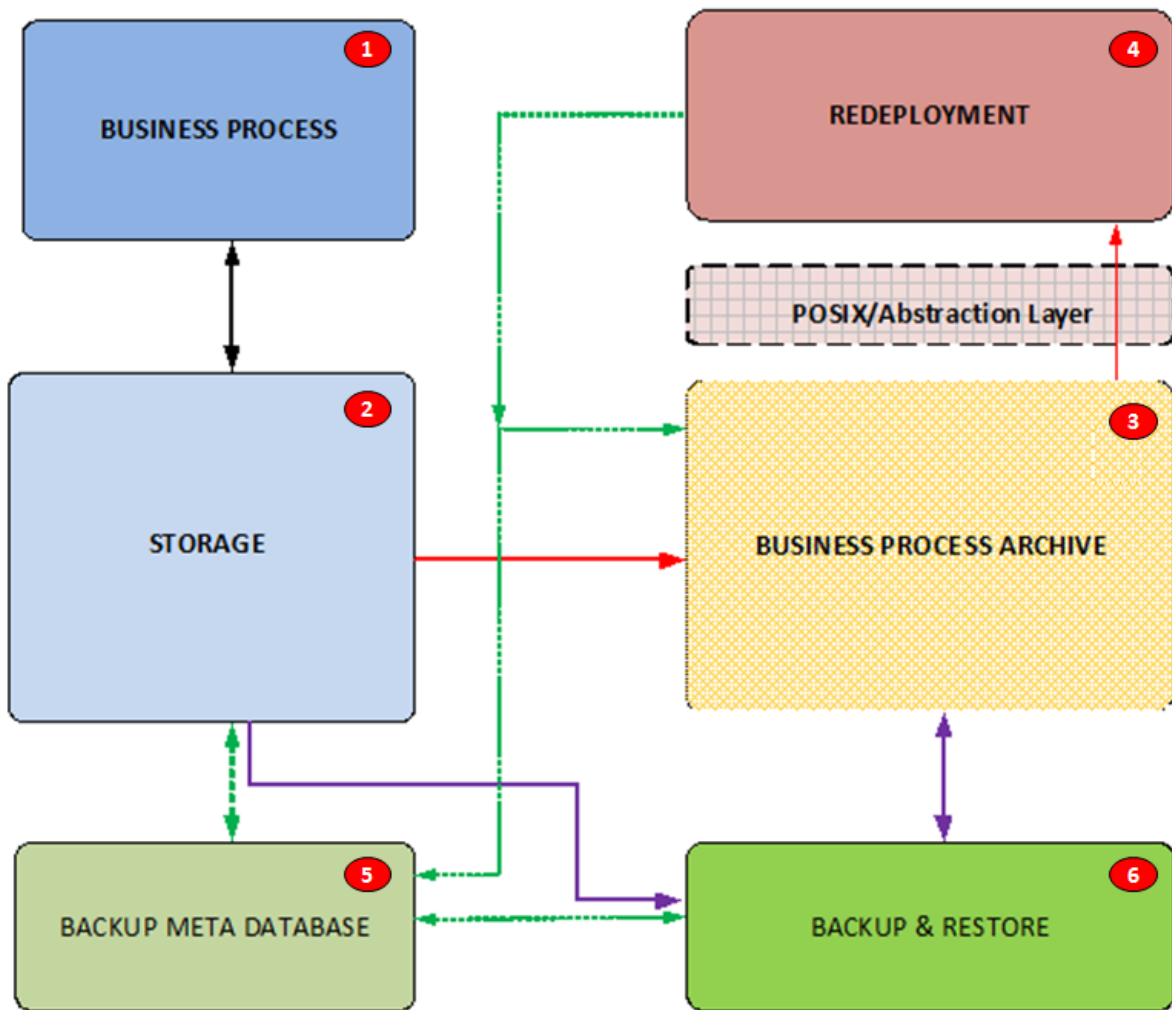


**Figure 62: The Redeployment Environment**

### 5.1.8 Storage Abstraction Layer

The Storage Abstraction Layer acts as a common interface to prevent storage architectures and products becoming a point of failure or vendor lock in constraints. The goal of the abstraction layer is to ensure over time that both the abstraction layer and redeployment layer can operate independently of hardware and vendor changes by creating a common independent layer for open storage stacks to interact with API's without needing to fully know everything about the storage subsystem.

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI



**Figure 63: The Storage Abstraction Environment**

### 5.1.9 Offsite Protection

To complete the protection cycle, archival data that has been moved to backup and restore environment must be replicated or tape-out to offsite to ensure redundancy in the protection environment. This also addresses concerns over the infinite loop of backing up the backups. When there is a tertiary copy of the data we remove the single point of failure within a specific location.

TIMBUS	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

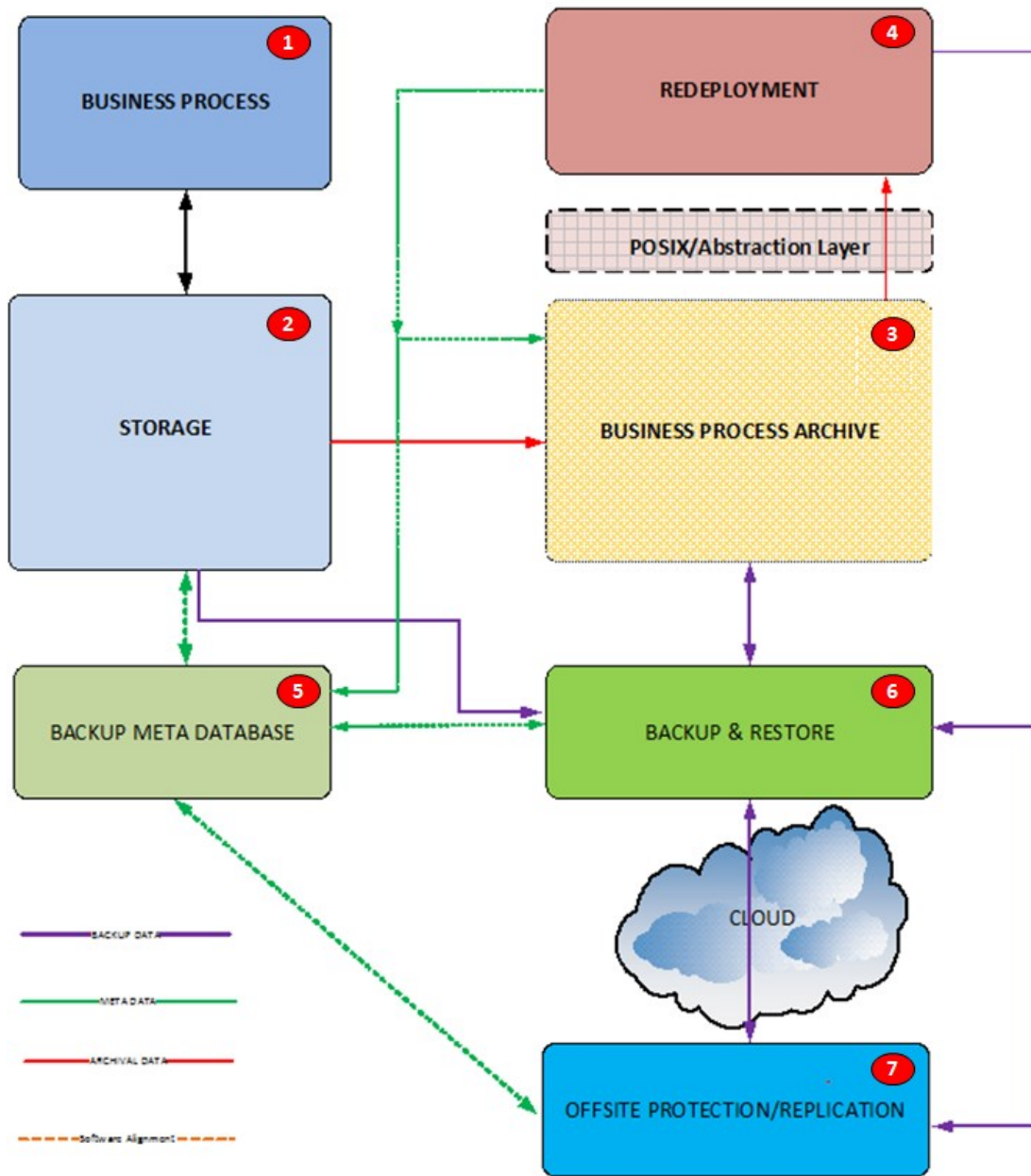


Figure 64: Offsite Protection

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

### 5.1.10 Relationship of D5.3 Architecture to D5.5 Architecture

The architecture developed in D5.3 has a relationship with the 5.5 work package “Refined Preservation Architecture” because it provides the infrastructure configuration on which the WP6 tools will run. To understand these relationships the following mapping of infrastructure architecture components from D5.3 to the software architecture components in D5.5 has been undertaken. Tool integration is underway at the time of writing and as stated in the introduction, as the TIMBUS development executes, the infrastructure architecture, along with the relationship to the D5.5 components will be under a continual improvement process. It is not possible in all cases to map the infrastructure components to a software component, for example the offsite protection offered in the infrastructure architecture does not have a corresponding representation in the software architecture.

**Table 24: D5.3 to D5.5 Mapping**

<b>Infrastructure Architecture Components (D5.3)</b>	<b>Software Architecture Components (D5.5)</b>
1. Business Process	Context Acquisition Module & Context Agent Module
2. Storage	DP Engine & Reasoning Module & Context Module
3. Business Process Archive	Digital Preservation Expert Suite
4. Redeployment	Digital Preservation Expert Suite
5. Backup Meta Database	No direct mapping is possible
6. Backup and Restore	No direct mapping is possible
7. Offsite Protection	No direct mapping is possible

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

## 6 Outlook and Conclusions

This document related the detailed look that the partners contributing to D5.3 took into the hardware and software components that are used today to provide archive architectures. These were carefully assessed and their relative advantages and disadvantages were considered. Based on this analysis, the reference infrastructure architecture for the TIMBUS project was proposed. The architecture is not intended to be prescriptive. It will not dictate whether disk is better than tape, whether SANs are better than NAS, or if block, file or object storage is the best solution for your organisation. It is not possible to categorically say that one of these is always going to be better than the other in every scenario. Often, it will come down to the requirements of the specific use case, aversion to undue risks relating to data loss and the budget available to the project. Bearing this in mind, the architecture presented in this deliverable is based on industry best practice and is flexible enough to support all of these approaches.

Disk is one of the fundamental elements of any archival solution and with its proven track record and long term roadmap in place there is no doubt that hard disks will be a predominant feature in any long term preservation architecture. Disk is not alone, tape was widely used in the backup and archive business well before disk was introduced. Many debates are on-going surrounding tape-based architectures; this document related that tape has many advantages such as flexibility, reduced error rate, constant capacity development and in recent years a major breakthrough with LFTS enabling tape to extend into a file system format allowing for easy access and distribution of data. Writing to two or more disks at the same time or having the same data stored on separate disks enables the data to be recovered in the event of a disk failure without resorting to expensive data recovery techniques; this approach is called RAID, Redundant Array of Inexpensive Disk. RAID offers various combinations of redundancy and data allocation across many disks.

Clearly primary storage in the shape of CPU and cache based memory has increased rapidly outpacing disk and tape. Solutions such as Disk Caching Disks (DCD's) have enabled disk and tape to keep up with the high performing applications available today.

Moving up the technology stack, logical storage structures come in to focus. There are three structures discussed, block level, file level and object level. Each has specific characteristics that make them appealing to the storage architect and end users.

The next level incorporates the actual file system. A key element in file systems is possessing understanding of the underlying nature of the data. This is called meta-data. We must know what the last modified date is and where exactly on the storage the file is stored. There are numerous file systems offering file management.

Storage models incorporate the disk enclosures and compute that make it an intelligent system. There are two types of model, direct attached DAS and Network Attached/Storage attached NAS/SAN. How these storage models interconnect with the front end systems is via the storage interface and connectivity. A SAN may use fibre channel over Ethernet or just fibre cable to connect to a centralised fabric to enable front end devices to communicate with the back end storage. There are many interfaces available such as SCSI, SATA, USB, Ethernet and Fibre Channel.

D5.3_M24_Architecture_For_VSRI	Dissemination Level: Public	Page 103
--------------------------------	-----------------------------	----------

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

There are generally two ways we can archive data from a hardware perspective, on disk or on tape. As we stated earlier, tape has been around a long time and is a long established archival medium. Disk has grown in popularity mainly due to online archives. The business requirement will establish what is appropriate to use as a medium for the archive. It is likely that the cost of the medium will range from expensive to less expensive based on accessibility and performance. This is known as tiered storage. It is recommended to base less accessed data on lower tier hardware thus reducing the operational costs.

There is no specific follow-up deliverable for D5.3 in the remaining time in the TIMBUS project as is the case with the D5.5 architecture, work package 6, specifically T6.7, will become the custodian of D5.3 after M24. It is anticipated that any experience gained in the execution of the TIMBUS research, and any feedback obtained on the architecture may lead to architectural changes. In this event, those changes will be published when T6.7 reports out its M36 status in deliverable D6.6. It is also intended that the information in this deliverable will form the basis for the development of training modules which will be delivered as part of the dissemination work in work package 3.



<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

## 7 References

No	Referenced Material	URL Link to Referenced Material
1	IBM's big data website	<a href="http://www-01.ibm.com/software/data/bigdata/">http://www-01.ibm.com/software/data/bigdata/</a>
2	DHSR's Blog	<a href="http://blog.dshr.org/2012/10/storage-will-be-lot-less-free-than-it.html">http://blog.dshr.org/2012/10/storage-will-be-lot-less-free-than-it.html</a>
3	Virgin Atlantic's IT director David Bullman recently stated that their new 787 fleet will create a half terabyte of data for each flight	<a href="http://www.networkworld.com/news/2013/030713-boeing-787s-to-create-half-267470.html?source=nww_rss">http://www.networkworld.com/news/2013/030713-boeing-787s-to-create-half-267470.html?source=nww_rss</a>
4	IDC Report on digital archive growth	<a href="http://www.insic.org/news/A&amp;S%20Roadmap.pdf">http://www.insic.org/news/A&amp;S%20Roadmap.pdf</a>
5	ESG Report, 2010	"Total Worldwide Archive Capacity by Media Type"
6	HAMR could enable up to 60TB drives on the 3.5" standard desktop HDD	<a href="http://www.insic.org/news/A&amp;S%20Roadmap.pdf">http://www.insic.org/news/A&amp;S%20Roadmap.pdf</a>
7	Maximum HDD Areal Density	HIS iSuppli Research, May 2012
8	Longitudinal versus Perpendicular Recording	<a href="http://www.wdc.com/wdproducts/library/other/2579-701185.pdf">http://www.wdc.com/wdproducts/library/other/2579-701185.pdf</a>
9	HDD Track	<a href="http://static.ddmcdn.com/gif/hard-disk-track.gif">http://static.ddmcdn.com/gif/hard-disk-track.gif</a>
10	Magnetic drives and are more responsive making them more efficient	<a href="http://www.imation.com/Global/enUS/Scalable%20Storage/Products/InfiniVault%20Storage%20Appliances/ESG_Marke_Report_Logic%20and%20Value%20of%20a%20Tiered%20Archive.pdf">http://www.imation.com/Global/enUS/Scalable%20Storage/Products/InfiniVault%20Storage%20Appliances/ESG_Marke_Report_Logic%20and%20Value%20of%20a%20Tiered%20Archive.pdf</a>
11	TIMBUS training day October 2012	<a href="http://www.youtube.com/watch?v=ulbmURmkv6w">http://www.youtube.com/watch?v=ulbmURmkv6w</a>
12	Magnetic tape was first used to record computer data in 1951	<a href="http://www.cratsindia.com/b/magnetic_tape_storage.pdf">http://www.cratsindia.com/b/magnetic_tape_storage.pdf</a>
13	Random vs Sequential Access	<a href="http://en.wikipedia.org/wiki/File:Random_vs_sequential_access.svg">http://en.wikipedia.org/wiki/File:Random_vs_sequential_access.svg</a>
14	Cloud based long term archival solution	<a href="http://www.horison.com/21stCenturyTape.pdf">http://www.horison.com/21stCenturyTape.pdf</a>
15	Worldwide Archive Data in Petabytes	<a href="http://www.storagenewsletter.com/images/public/sites/StorageNewsletter.com/articles/icono8/insic_tape_roadmap_540.jpg">http://www.storagenewsletter.com/images/public/sites/StorageNewsletter.com/articles/icono8/insic_tape_roadmap_540.jpg</a>
18	Microsoft Technet	Technet.microsoft.com
19	NFS Architecture	<a href="http://www.ibm.com/developerworks/linux/library/l-network-file-systems/?ca=drs-">http://www.ibm.com/developerworks/linux/library/l-network-file-systems/?ca=drs-</a>
21	Example SAN Architectures	<a href="http://www.ts.avnet.com/uk/solution_guides/assets/10gb_ethernet_solution_guide">http://www.ts.avnet.com/uk/solution_guides/assets/10gb_ethernet_solution_guide</a>
22	USB Connectors	<a href="http://www.l-com.com/content/USB-Tutorial.html">http://www.l-com.com/content/USB-Tutorial.html</a>
24	NASA Degaussed tapes	<a href="http://www.reuters.com/article/2009/07/16/us-nasa-tapes-idUSTRE56F5MK20090716">http://www.reuters.com/article/2009/07/16/us-nasa-tapes-idUSTRE56F5MK20090716</a>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

25	NASA McDonalds tapes	<a href="http://www.thelivingmoon.com/47john_lear/02files/Lunar_Orbiter_Tapes_Found.html">http://www.thelivingmoon.com/47john_lear/02files/Lunar_Orbiter_Tapes_Found.html</a>
26	Symantec NetBackup Architecture	<a href="http://www.symantec.com/business/support/index?page=content&amp;id=DOC5159">http://www.symantec.com/business/support/index?page=content&amp;id=DOC5159</a>
30	High Availability Storage	<a href="http://www.sqlsoldier.com/wp/wp-content/uploads/2011/01/GeoCluster.jpg">http://www.sqlsoldier.com/wp/wp-content/uploads/2011/01/GeoCluster.jpg</a>
31	DPN	<a href="https://lib.stanford.edu/files/pasig-oct2012/11-Cramer_DPN_for_PASIG.pdf">https://lib.stanford.edu/files/pasig-oct2012/11-Cramer_DPN_for_PASIG.pdf</a>
34	Data De-Duplication Methodologies Example	<a href="http://www.it-ops.com/assets/files/documents/BakBone_Evaluating_Deduplication_Solutions_wp.pdf">http://www.it-ops.com/assets/files/documents/BakBone_Evaluating_Deduplication_Solutions_wp.pdf</a>
35	Active Archive Environment	<a href="http://www.activearchive.com/common/pdf/ActiveArchiveOverviewBrochure.pdf">http://www.activearchive.com/common/pdf/ActiveArchiveOverviewBrochure.pdf</a>
36	Mark Peters report	<a href="http://www.esg-global.com/author/mark-peters/#cs_Mark%20Peters%20esg">http://www.esg-global.com/author/mark-peters/#cs_Mark%20Peters%20esg</a>
37	Intel IT's Data Storage Infrastructure	<a href="http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/data-storage-solutions-paper.pdf">http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/data-storage-solutions-paper.pdf</a>
38	Integrated In-System Storage Architecture for High Performance Computing	<a href="http://www.mcs.anl.gov/uploads/cels/papers/P2092-0512.pdf">http://www.mcs.anl.gov/uploads/cels/papers/P2092-0512.pdf</a>
39	CSAL generic storage abstractions	<a href="http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&amp;arnumber=5708493">http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&amp;arnumber=5708493</a>
40	Data Multiplexing	<a href="http://documentation.commvault.com/dell/release_7_0_0/books_online_1/english_us/images/data_multiplexing/data_multiplexing.gif">http://documentation.commvault.com/dell/release_7_0_0/books_online_1/english_us/images/data_multiplexing/data_multiplexing.gif</a>
41	Backup Selections Window	<a href="http://www.symantec.com/business/support/library/BUSINESS/ATLAS/images_v1/267291/multiplexing-4.jpg">http://www.symantec.com/business/support/library/BUSINESS/ATLAS/images_v1/267291/multiplexing-4.jpg</a>
42	SNIA shared storage Model	<a href="http://www.snia.org/education/storage_networking_primer/shared_storage_model">http://www.snia.org/education/storage_networking_primer/shared_storage_model</a>
43	SAN Block Storage	<a href="http://www.snia.org/sites/default/files/SNIA-SSM-text-2003-04-13.pdf">http://www.snia.org/sites/default/files/SNIA-SSM-text-2003-04-13.pdf</a>
44	Super DLT Tape	<a href="http://h20000.www2.hp.com/bc/docs/support/SupportManual/c00517361/c00517361.pdf">http://h20000.www2.hp.com/bc/docs/support/SupportManual/c00517361/c00517361.pdf</a>
45	Inside an Automated Tape Library	<a href="http://www.ibm.com/common/ssi/GIF/ALET/3584XXC.GIF">http://www.ibm.com/common/ssi/GIF/ALET/3584XXC.GIF</a>
46	2009 NEC whitepaper	<a href="http://www.necam.com/Docs/?id=e93c05ba-ce09-47cb-aa54-68d440849be0">http://www.necam.com/Docs/?id=e93c05ba-ce09-47cb-aa54-68d440849be0</a>
47	Intel Processor Comparisons	<a href="http://www.intel.com/content/www/us/en/processor-comparison/compare-intel-processors.html">http://www.intel.com/content/www/us/en/processor-comparison/compare-intel-processors.html</a>
48	Response time factor	<a href="http://www.freepatentsonline.com/7467291.html">http://www.freepatentsonline.com/7467291.html</a>
49	Access arm travels shorter distances across the diameter of the disk	<a href="http://home.olemiss.edu/~misbook/hgmm.htm">http://home.olemiss.edu/~misbook/hgmm.htm</a>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

50	LTFS Partition	<a href="http://www.spectrallogic.com/index.cfm?fuseaction=products.displayContent&amp;CatID=2173">http://www.spectrallogic.com/index.cfm?fuseaction=products.displayContent&amp;CatID=2173</a>
51	Relationship between LTFS, the O/S and the tape hardware	<a href="http://storageconference.org/2010/Papers/MSST/Pease.pdf">http://storageconference.org/2010/Papers/MSST/Pease.pdf</a>
52	Tape capabilities and data transfer rates	<a href="http://www.spectrallogic.com/index.cfm?fuseaction=home.displayFile&amp;DocID=2513&amp;fromRegForm=true">http://www.spectrallogic.com/index.cfm?fuseaction=home.displayFile&amp;DocID=2513&amp;fromRegForm=true</a>
53	RAID 0	<a href="http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo/artwork/raid0-d.gif">http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo/artwork/raid0-d.gif</a>
54	RAID 1	<a href="http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo/artwork/raid1-d.gif">http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo/artwork/raid1-d.gif</a>
55	RAID 1-E	<a href="http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo/artwork/raid1e-d.gif">http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo/artwork/raid1e-d.gif</a>
56	RAID 5-E	<a href="http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo/artwork/raid5-e.gif">http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo/artwork/raid5-e.gif</a>
57	Openstack Folsom Architecture	<a href="http://ken.pepple.info/openstack/2012/09/25/openstack-folsom-architecture/">http://ken.pepple.info/openstack/2012/09/25/openstack-folsom-architecture/</a>
58	Openstack Cloud Components	<a href="http://ken.pepple.info/openstack/2012/09/25/openstack-folsom-architecture/">http://ken.pepple.info/openstack/2012/09/25/openstack-folsom-architecture/</a>
59	Openstack website	<a href="http://docs.openstack.org/developer/swift/overview_architecture.html">http://docs.openstack.org/developer/swift/overview_architecture.html</a>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

## Appendix A: Supplementary Archival Technology Material

### A.1 Forms of Hard Disks

There are two forms in which you can use a hard disk, internally within your system such as a server, PC or a storage array. Alternatively you could use this hard drive disk as an externally connected disk offering portability.

*Internal Hard Disk:* Internal hard disk is made up of several metallic platters, a motor, an access arm and read-write heads sealed inside a container. There are two sizes of drives 5.25” and 3.5” in a diameter. 3.5” hard disks are faster because the access arm travels shorter distances across the diameter of the disk<sup>49</sup>.

*External Hard-Disk Drives:* These are hard disk drives that are not built into the chassis of a computer. These types of drives are usually treated as peripheral devices transportable and mainly used as portable disks used for backup and recovery or smaller less critical systems. You can connect many of these disks using USB or SCSI.

### A.2 LTFS Editions

The Linear Tape File System offers two different editions. The Single Drive Edition allows access to all of the data in a cartridge loaded on a single drive as if it were on disk. The Library Edition adds support for multiple cartridges in a tape library, and automatically loads the cartridge per file access.

Both editions use the file system’s format and resources of the operating system (OS) on which it is running to graphically display the contents of a tape library in the OS graphical user interface (GUI) format, typically a folder/tree structure. Both editions support LTO Ultrium generation 6 and 5.

#### Linear tape File System (LTFS)

LTFS is an open source format which allows an end user to access files on a tape as if it were a disk device. LTFS will enable drag and drop actions from tape to disk which is just not possible today. Users can run any application designed for disk files against tape data without concern for the fact that the data is physically stored on tape.

An LTFS Volume can be mounted and volume content accessed with full use of the data without the need to access other information sources. Data can be passed between sites and applications using only the information written to an LTFS Volume. Files can be written to, and read from, an LTFS Volume using standard POSIX file operations.

The Linear Tape File System offers two different editions. The Single Drive Edition allows access to all of the data in a cartridge loaded on a single drive as if it were on disk. The Library Edition adds support for multiple cartridges in a tape library, and automatically loads the cartridge per file access.

---

<sup>49</sup> <http://home.olemiss.edu/~misbook/hgmm.htm>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

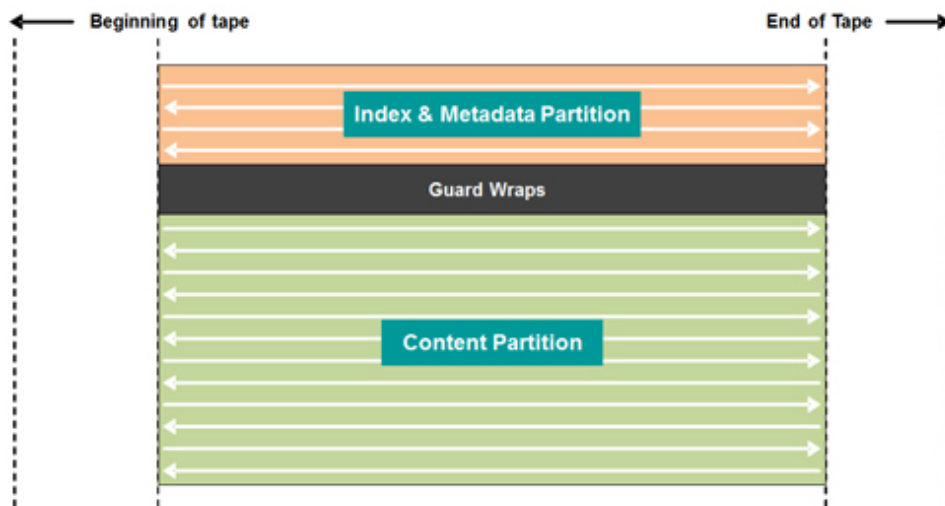
Both editions use the file system’s format and resources of the operating system (OS) on which it is running to graphically display the contents of a tape library in the OS graphical user interface (GUI) format, typically a folder/tree structure. Both editions support LTO Ultrium generation 6 and 5.

The meta-data of each cartridge, once mounted, is cached in server memory. Meta-data operations, such as browse directory and filename search, do not require tape movement.

### LTFS Architecture

The LTFS technology consists of two components, the on-tape index format and the file system implementation. LTFS requires the multi-partition capability found in tape systems like LTO-5. In a basic LTFS implementation, one partition, typically a comparatively small one referred to as the index partition, is used to record the tape index; another partition, made up of the remainder of the tape and referred to as the data partition, is used to store file data blocks.

An LTFS volume is comprised of a pair of LTFS partitions. LTFS defines two partition types, data partition and index partition. An LTFS volume must contain exactly one Data Partition and exactly one Index Partition. The figure below illustrates the partitions with LTFS format on a tape.



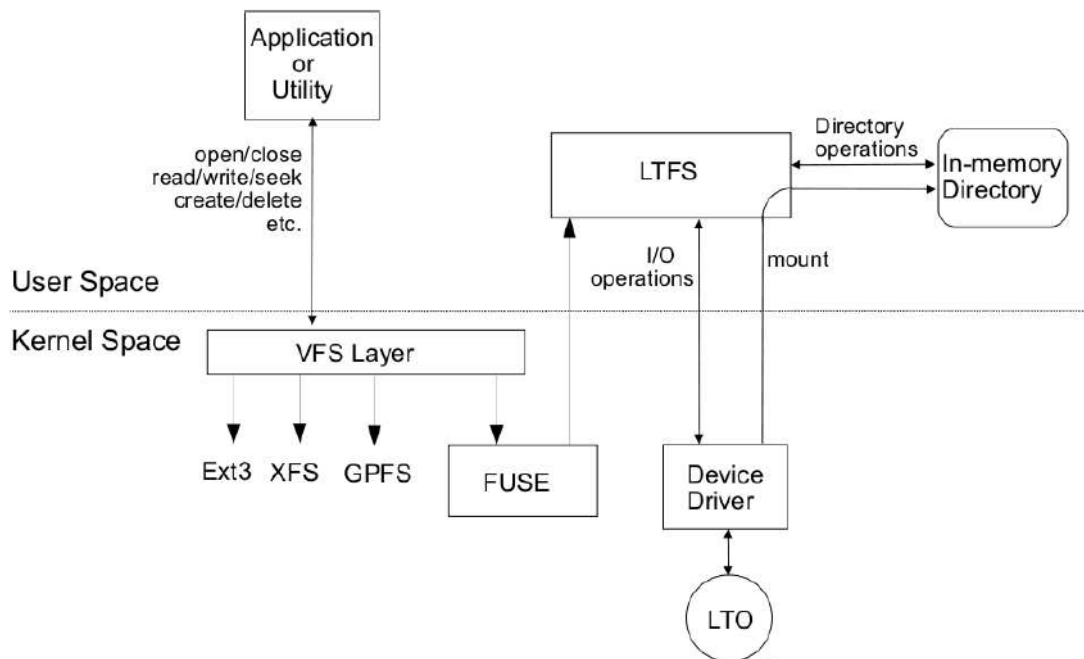
**Figure 65: LTFS Partition<sup>50</sup>**

After the tape cartridge has been loaded the drive then reads the index and metadata partition. The entire tape is not scanned leading to a faster and more efficient experience. Traditional tape methods will access data sequentially meaning all data is scanned before your file is accessed.

LTFS can be implemented on a standalone tape drive or on a tape library, single drive mode or library mode. The figure below illustrates the relationship between LTFS, the O/S and the tape hardware.

<sup>50</sup> <http://www.spectrallogic.com/index.cfm?fuseaction=products.displayContent&CatID=2173>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI



**Figure 66: Relationship between LTFS, the O/S and the tape hardware<sup>51</sup>**

### A.3 Tape Performance and Capacity

Data transfer rate is defined as the speed at which data is written to tape from the drive's internal buffer. This is usually measured in megabytes per second (MB/sec). If the data transfer from the host system to the drive is significantly slower than the drive's transfer rate, a great deal of start-stop tape motion occurs while the drive waits for more data. Through speed matching and data buffers, stop-start action is minimised.

Tape capacity is measured by the amount of data that can be recorded on a single tape cartridge. Tape manufacturers maximise capacity by increasing the bit density on a given area of tape. Hardware data compression is also used to increase capacity. As can be seen in Figure 67, per tape, an LTO-5 drive can offer 1.5TB of native storage or 3TB if the data is compressed. This format can also offer 140 native and 280 MBps transfer rates. The chart below also from Spectra Logic outlines the native and compressed capacities and also the transfer rates for each tape technology.

<sup>51</sup> <http://storageconference.org/2010/Papers/MSST/Pease.pdf>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

Drive Technology	Data Transfer Rate, Native	Data Transfer Rate, Compressed	Native Capacity	Compression Ratio
LTO-6	160 MB/s	400 MB/s	2.5 TB	2.5:1
LTO-5	140 MB/s	280 MB/s	1.5 TB	2:1
TS1140	250 MB/s	Not available	4 TB	3:1
T10000C	240 MB/s	360 MB/s	5 TB	2:1
T9940B	30 MB/s	70 MB/s	200 GB	2:1
T9840D	30 MB/s	70 MB/s	75 GB	2:1

**Figure 67: Tape capabilities and data transfer rates<sup>52</sup>**

In relation to interfaces the standard tape drive interfaces available today are Serial attached SCSI (SAS) and fibre channel port.

#### **A.4 Disk Caching Disks (DCD)**

Disk storage namely hard disk drives have not kept with the pace of other forms of primary storage such as RAM in terms of speed mainly due to the mechanical nature of hard disks. On the other hand, hard disks have dramatically increased in capacity as we have discussed previously in this chapter.

Disk caching disks (DCD) are a new approach which is designed to improve write performance for mainly office based environments which would ideally suit a digital archive solution. The main idea behind DCD is to take advantage of the differences between large and small disk transfer sizes by implementing an architecture consisting of three layers. The first layer is a RAM buffer, the second is a cache disk which stores data in a log format and the third level being a data disk which stores data the same as any of the data disks outlined above. The architecture for DCD works by having a log disk called the cache-disk and is used as an extension to the RAM buffer combined both cache write data. When the system becomes idle the cache disk then moves data across to the data disk.

The RAM will handle smaller write requests but when a large burst of requests occur the RAM will fill quickly. The RAM buffer will take this large request and transfer to the cache-disk in one large transfer. Now the RAM buffer is free again to handle requests. As far as the system is concerned it has one very large virtual RAM cache. A phase called 'de-staging' occurs when the system returns to normal or low levels of activity.

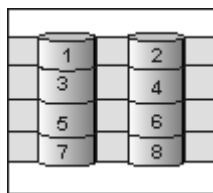
<sup>52</sup> <http://www.spectralogic.com/index.cfm?fuseaction=home.displayFile&DocID=2513&fromRegForm=true>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

The operating system does not need to understand anything different about the cache disk or RAM. In essence the DCD architecture after extensive testing yielded improved write performance across various types of workloads presented during the testing.

### A.5 RAID Levels

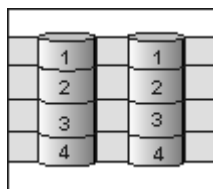
**RAID 0:** RAID level 0 simple stripes the data across all the disk drives in the array. This offers great speed but there is no redundancy in the data. You can implement RAID 0 on a single disk up to 16 drives. In the below diagram we have two disks which form part of a logical array with data stripped across the drives. There is no redundant data here.



**Figure 68: RAID 0**<sup>53</sup>

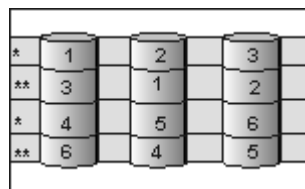
**RAID 1:** This level of RAID is known as mirroring. There are two physical hard disks form an array and the data is then striped across this array. The first half of the array is the original data that was written to disk and the second half of the array is the mirrored version of that original data.

RAID 1 offers data redundancy while also offering high levels of performance. The main problem with RAID 1 is the storage capacity is severely impacted due to the split in original data and mirrored data, i.e. 50%.



**Figure 69: RAID 1**<sup>54</sup>

There is an add-on to RAID1 called RAID1-E which combines the advantages of RAID1 by adding data striping thus extending the number of drives used from a minimum of 3 to a maximum of 16.



<sup>53</sup> <http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo/artwork/raid0-d.gif>

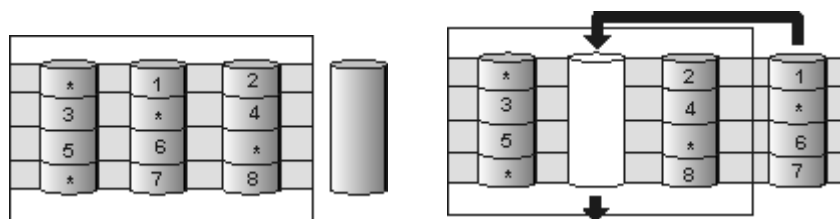
<sup>54</sup> <http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo/artwork/raid1-d.gif>



<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Figure 70: RAID 1-E** <sup>55</sup>

**RAID 5:** This is a very common implementation of RAID in the server environment. With RAID 5 stripes of data are written across all disks in the array along with parity information. RAID 5 offers data protection and increased throughput. You must have a minimum of three hard disk drives for RAID 5 and supports up to 16 hard disk drives. A key element with RAID 5 is capacity is reduced due to the need for capacity to store the parity information. This is known as the parity block and is noted as \* on the diagram below. Each drive will contain parity blocks which will reduce the overall capacity. This parity information contains information about the data from other blocks in the same stripe and is used to rebuild the spare drive if one disk fails. RAID 5 is only effective when a hot spare hard disk is used which further decreases the overall capacity as additional disks are needed to maintain the data protection. RAID 5 offers greater read times than write times and is ideal for environments that require faster read response times.



**Figure 71: RAID 5 and RAID 5-E** <sup>56</sup>

**RAID 6:** Is very similar to RAID 5 with an additional set of parity information. This form of RAID offers even higher levels of fault tolerance and can sustain two drive failures as opposed to just one with RAID5.

**RAID10:** Often referred to as RAID 1 + 0 is a combination of RAID level 0 which is striping and RAID 1 which is mirroring. Disks are mirrored in pairs for redundancy and performance and this data is striped across multiple disks for greater performance.

You could have six hard disks. By creating three physical arrays with two hard disks each we then create a spanned array between the three sets. A logical drive is created on each array and data is striped across the drives in the array creating blocks. With RAID level 10 we have mirroring on the sub logical drive (separate array) and spanning RAID 0 across the array of logical drives. This approach supports up to 128 hard disks on SATA controllers and offers 100% redundancy.

We now have a stack created, from the raw disk or tape level we can decide on whether SCSI or SSD is the physical type of disk we need. We can decide on the logical storage structure that best suits our requirements, whether it is block based or object based. We implement a file level structure above this to manage the files in an organised system. We then protect the data on disk and offer better performance while writing data and reading data from this disk by implementing RAID if we are using block based storage. It must be noted that RAID does not offer benefits on object based storage. We will discuss redundancy for object based storage in chapter five.

<sup>55</sup> <http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo/artwork/raid1e-d.gif>

<sup>56</sup> <http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo/artwork/raid5-e.gif>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

For many organisations piecing each component together can be a challenging task and require a lot of time and investment. There are many vendor based solutions and open cloud offerings in the storage solutions market. We call these storage models.

## A.6 Object-based storage

Object based storage opens up a whole new range of dynamic functionality. Object-based storage assigns unique object IDs to access data across all locations. This approach moves away from a lot of the traditional storage management tasks such as creating disks and RAID groups. The flat, universal namespace allows global access to stored content from anywhere the distributed application runs. Applications can also efficiently associate metadata with stored objects without using a dedicated database. Sharing vast storage resources means application administrators do not need to modify application files. Object-based storage usually has elements of file systems in order to handle processes like file archiving, but it is not founded on that architecture and its drawbacks. We are familiar already with CAS, Content Addressable Storage. This is a close relation of object-oriented storage, CAS ensures there is only one ID for any object. When the CAS object is retrieved, it can be hashed again and checked against its ID to verify identity. CAS de-dupes at the object level for copy control.

There are currently seven components in Openstack storage. These are Object Store, Image, Compute, Dashboard, Identity, Network and Block Storage.

To achieve a highly scalable cloud operating system each component is designed to work in unity to provide a complete Infrastructure as a Service (IaaS). This integration is facilitated through public application programming interfaces (APIs) that each service offers and uses.

**Object Store** or codename *Swift* allows you to store or retrieve files but not mount directories like a fileserver. Several companies provide commercial storage services based on Swift. These include KT, Rackspace from which Swift originated and Internap. Swift is also used internally at many large companies to store their data.

**Image** codenamed *Glance* provides a catalogue and repository for virtual disk images. These disk images are mostly commonly used in OpenStack Compute. While this service is technically optional, any cloud of size will require it.

**Compute** codenamed *Nova* provides virtual servers upon demand. Rackspace and HP provide commercial compute services built on Nova and it is used internally at companies.

**Dashboard** codenamed *Horizon* provides a modular web-based user interface for all the OpenStack services. With this web GUI, you can perform most operations on your cloud like launching an instance, assigning IP addresses and setting access controls.

**Identity** codenamed *Keystone* provides authentication and authorization for all the OpenStack services. It also provides a service catalogue of services within a particular OpenStack cloud.

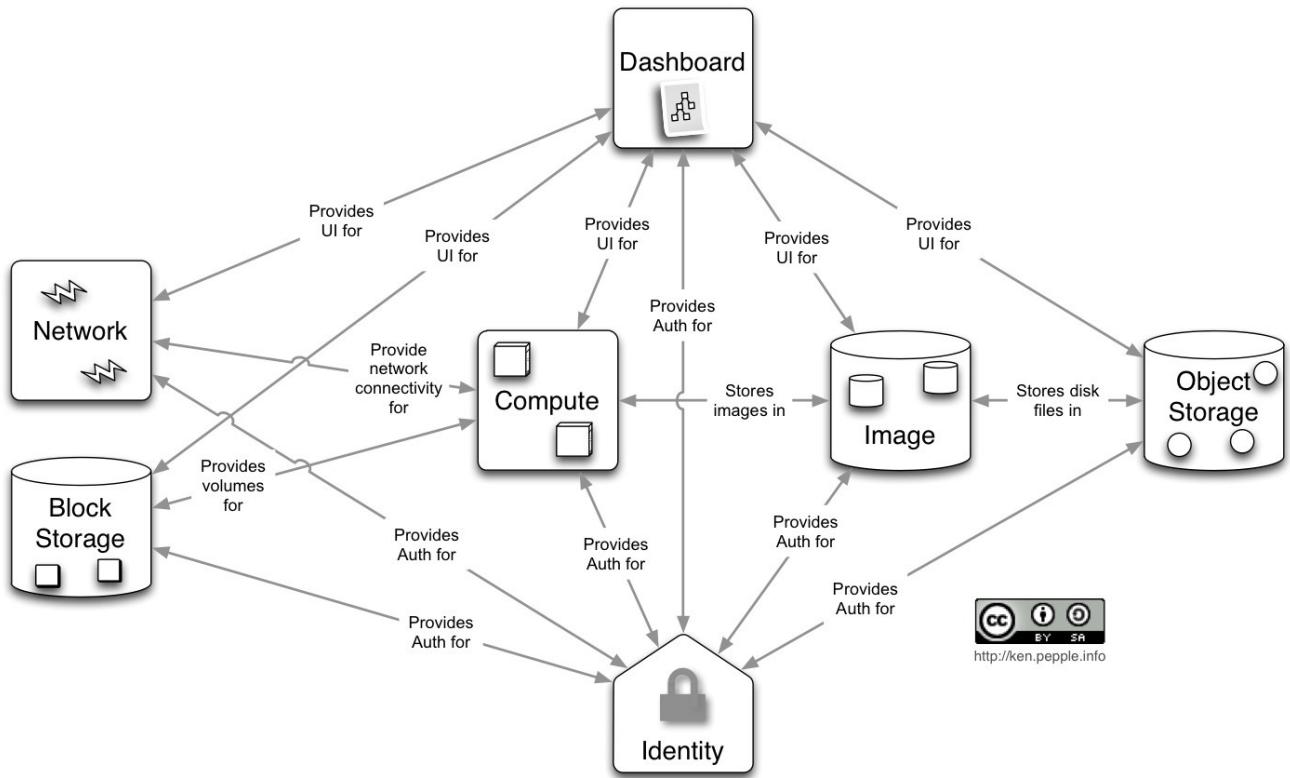
**Network** codenamed *Quantum* provides "network connectivity as a service" between interface devices managed by other OpenStack services. The service works by allowing users to create their own networks and then attach interfaces to them. Quantum has a pluggable architecture to support many popular networking vendors and technologies.

D5.3_M24_Architecture_For_VSRI	Dissemination Level: Public	Page 114
--------------------------------	-----------------------------	----------

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Block Storage** codenamed *Cinder* provides persistent block storage to guest VMs. This project was born from code originally in Nova, the nova-volume service. Please note that this is block storage (or volumes) not file systems like NFS or CIFS share.

The following image is taken from the Openstack Folsom Architecture:



**Figure 72: Openstack Folsom Architecture<sup>57</sup>**

The dashboard provides web front end for the other services. Compute stores and retrieves Images and Meta-data. Network provides virtual networking for compute. Block Storage provides storage volumes for Compute and Image can store the virtual disks in Object store. All the components authenticate with Identity. To summarise the open cloud components and associated projects the following table:

<sup>57</sup> <http://ken.pepple.info/openstack/2012/09/25/openstack-folsom-architecture/>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

**Table 25: Openstack Cloud Components<sup>58</sup>**

Project	Component	Description
Horizon	Dashboard	User & Admin interface
Nova	Compute/block device	Virtual servers and volumes
Glance	Image Service	VM disk images
Swift	Storage as a Service	Object Storage
Quantum/Melange	Networks	Secure Virtual Networks

In Chapter 3 we discussed logical storage structures on how object-based storage was becoming more prevalent in the Open Cloud World today. The main components of the Swift architecture are proxy servers, storage/object servers, The Ring, Container Server, Replication and Updaters.

#### ***Proxy server***

Handle all user requests and communicates to underlying storage on isolated network to send and receive data from users. User authentication performed by keystone. The proxy server owns the storage ring definition for all storage nodes.

#### ***Storage servers***

Also called the object server, Swift designates a drive or partition as the storage volume, usually /sdb/sdb1. The File system is independent for Swift storage. Whatever you can mount and format as XFS will work using either local or remote storage. Swift is independent of the underlying file system. The storage or object server executes the partition map and ring configuration.

Each object is stored using a path derived from the object name's hash and the operation's timestamp. Last write always wins, and ensures that the latest object version will be served. A deletion is also treated as a version of the file, a 0 byte file ending with ".ts", which stands for tombstone. This ensures that deleted files are replicated correctly and older versions don't magically reappear due to failure scenario.

#### ***The Ring***

A ring represents a mapping between the names of entities stored on disk and their physical location. There are separate rings for accounts, containers, and objects. When other components need to perform any operation on an object, container, or account, they need to interact with the appropriate ring to determine its location in the cluster.

<sup>58</sup> <http://ken.pepple.info/openstack/2012/09/25/openstack-folsom-architecture/>

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

The Ring maintains this mapping using zones, devices, partitions, and replicas. Each partition in the ring is replicated, by default, 3 times 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 in failure scenarios.

Data can be isolated with the concept of zones in the ring. Each replica of a partition is guaranteed to reside in a different zone. A zone could represent a drive, a server, a cabinet, a switch, or even a datacentre.

The partitions of the ring are equally divided among all the devices in the Swift installation. When partitions need to be moved around (for example if a device is added to the cluster), the ring ensures that a minimum number of partitions are moved at a time, and only one replica of a partition is moved at a time.

Weights can be used to balance the distribution of partitions on drives across the cluster. This can be useful, for example, when different sized drives are used in a cluster. The ring is used by the Proxy server and several background processes.

### **Container Server**

The Container Server's primary job is to handle listings of objects. It does not know where those objects are, just what objects are in a specific container. The listings are stored as sqlite database files, and replicated across the cluster similar to how objects are. Statistics are also tracked that include the total number of objects, and total storage usage for that container.

### **Replication**

The replication process keeps the system consistent. It works by comparing local data with each remote copy to ensure they all contain the latest version. Object replication uses a hash list to quickly compare subsections of each partition, and container and account replication use a combination of hashes and shared high water marks.

The replication updates are push based. An update is done by syncing the files to another node. The replicator also ensures that data is removed from the system. When an item object, container, or account is deleted, a tombstone is set as the latest version of the item. The replicator will see the tombstone and ensure that the item is removed from the entire system.

### **Updaters**

When data or an account cannot be updated immediately due to network failure or excessive loads an update is queued on the file system and then the updater will deal with the failed updates. The information contained above is derived from the official documentation from Openstack.org.<sup>59</sup>

CEPH is another Open Cloud logical storage solution which uniquely delivers object, block, and file storage in one unified system. The CEPH Object Store, called RADOS, is the object storage component for CEPHFS file systems, CEPH RADOS Gateways, and CEPH Block Devices.

<sup>59</sup> [http://docs.openstack.org/developer/swift/overview\\_architecture.html](http://docs.openstack.org/developer/swift/overview_architecture.html)

<b>TIMBUS</b>	WP 5 – Software Architecture for Digital Preservation
Deliverable	D 5.3: Architecture for VSRI

## CEPH Block Storage

A block is a sequence of bytes e.g. a 512-byte block of data. Block-based storage interfaces are the most common way to store data with rotating media such as hard disks, CDs, floppy disks, and even traditional 9-track tape. The ubiquity of block device interfaces makes a virtual block device an ideal candidate to interact with a mass data storage system like CEPH.

CEPH block devices are thin-provisioned, resizable and store data striped over multiple OSDs in a CEPH cluster. CEPH block devices leverage RADOS capabilities such as snapshotting, replication and consistency. CEPH's RADOS Block Devices (RBD) interact with OSDs using kernel modules or the librbd library.