D6.6: Business Process Preservation Test Bed

WP 6 – Intelligent Tools and Technologies to Support Digital Preservation of Business Processes

Delivery Date: 31/03/2014

Dissemination Level: PU

TIMBUS is supported by the European Union
under the 7th Framework Programme
for research and technological development and demonstration activities (FP7/2007-2013)
under grant agreement no. 269940

Deliverable Lead		
Name	**Organisation**	**e-mail**
Perumal Kuppuudaiyar	Intel	perumal.kuppuudaiyar@intel.com

Contributors		
Name	**Organisation**	**e-mail**
Mike Nolan	Intel	michael.nolan@intel.com
Rudolf Mayer	SBA	rmayer@sba-research.at
Johannes Binder	SBA	jbinder@sba-research.at
Stephan Strodl	SBA	sstrodl@sba-research.at
Paulo Ferreira	INESC-ID	paulo.ferreira@inesc-id.pt
Sridhar Voorakkara	Intel	sridhar.voorakkara@intel.com
Carlos Coutinho	CMS	carlos.coutinho@caixamagica.pt

Internal Reviewer		
Name	**Organisation**	**e-mail**
Rudolf Mayer	SBA	rmayer@sba-research.at
Stephan Strodl	SBA	strodl@ifs.tuwien.ac.at

Document History			
Version	**Date**	**Author**	**Changes**
V1.0	05/02/2012	Isabel Mundet	Draft Version
V1.1	19/11/2013	Mike Nolan	Initial Structuring
V1.2	14/01/2014	Perumal Kuppuudaiyar	SoTA update
V1.3	21/01/2014	Sridhar Voorakkara	API & Implementation
V1.4	25/02/2014	Perumal Kuppuudaiyar	Requirements & Design
V1.5	5/03/2014	Perumal Kuppuudaiyar	PI & Implementations

V1.6	8/03/2014	Sridhar Voorakkara	Use cases
V1.7	11/03/2014	Perumal Kuppuudaiyar	Finishing the doc
V1.9	11/03/2014	Mike Nolan	Peer Review
V2.0	20/03/2014	Rudolf Mayer	Internal Review
V2.1	25/03/2014	Perumal Kuppuudaiyar	Addressed Internal Review
V2.2	26/03/2014	Stephan Strodl	PCC and Quality Review Completed
V2.3	28/03/2014	Perumal Kuppuudaiyar	Review comments are addressed

Disclaimer

Table of Contents

TABLE OF CONTENTS 5

1 EXECUTIVE SUMMARY 14

2 INTRODUCTION 15

2.1 OBJECTIVE 15

2.1.1 Virtualisation and Storage Manager (Task 6.4) 16

2.1.2 Rerun/Integration Environment (Task 6.6) 16

2.1.3 Simulated Test Bed (Task 6.7) 16

2.2 SCOPE 17

2.3 TECHNICAL CONTEXT 17

3 STATE OF THE ART 19

3.1 STORAGE 19

3.1.1 Block Level Storage 19

3.1.2 Object Storage 20

3.1.3 File Level Storage 23

3.1.4 Object, Block and File Storage Comparison 23

3.1.5 Storage Service 24

3.1.6 Summary 26

3.2 VIRTUALISATION 26

3.3 TYPES OF VIRTUALISATION 28

3.3.1 Full or Native Virtualisation 28

3.3.2 Hardware-assisted Virtualisation 28

3.3.3 Para-Virtualisation 29

3.3.4 OS-Level Virtualisation 29

3.3.5 Application Level Virtualisation 30

3.4 EMULATION 30

3.4.1 Emulators in Digital Preservations 30

3.4.2 Preserving the Emulated Hardware 31

3.4.3 Digital Preservation Approaches 31

3.5 SOFTWARE CONTAINERS 33

3.6 SUMMARY 34

4 DIGITAL PRESERVATION EXPERTS SUITE (T6.4/T6.6) REQUIREMENTS 36

4.1 FUNCTIONAL REQUIREMENTS 37

4.2 NON-FUNCTIONAL REQUIREMENTS 44

5 DIGITAL PRESERVATION EXPERT SUIT - ARCHITECTURE ... 50

5.1 APPLICATION ARCHITECTURE OF DPES ... 50

5.1.1 Web UI and Service Interfaces ... 51

5.1.2 Business Object management ... 51

5.1.3 Data Layer ... 58

5.2 TECHNICAL ARCHITECTURE OF DPES SYSTEM ... 59

5.2.1 DPES Web Application Server ... 60

5.2.2 TIMBUS Storage Repository ... 61

5.2.3 Redeployment Server ... 62

6 USE CASES – PRESERVATION PERSPECTIVE ... 64

6.1 IOT – INTERNET OF THINGS (T6.7) ... 64

6.1.1 Digital Preservation Requirements ... 64

6.1.2 Preservation approaches for IoT use case. ... 65

6.1.3 Redeployment Plan ... 66

6.2 WP7 – USE CASES ... 67

6.2.1 Introduction ... 67

6.2.2 Phaidra Technical Architecture ... 67

6.2.3 Phaidra Business Processes ... 68

6.2.4 RCAAP Technical Architecture ... 69

6.2.5 Preservation approaches for Phaidra ... 70

6.2.6 Preservation approaches for RCAAP ... 70

6.2.7 Re-deployment plan for Phaidra ... 71

6.2.8 Re-deployment plan for RCAAP ... 71

6.3 WP8 – USE CASE ... 72

6.3.1 Introduction ... 72

6.3.2 Technical Architecture ... 73

6.3.3 Business Process ... 75

6.3.4 Preservation Requirements ... 76

6.3.5 Preservation Approaches for CAD/CAM Environment ... 76

6.3.6 Redeployment Plan ... 77

6.4 WP9 – USE CASE ... 77

6.4.1 Introduction ... 77

6.4.2 Technical Architecture ... 78

6.4.3 Business Process ... 80

6.4.4 Preservation approaches for eHealth ... 80

6.4.5 Redeployment Plan ... 81

7 IMPLEMENTATION AND INTEGRATION ... 82

7.1 iERM and DPES API ... 82
7.2 Storage Client Implementation ... 87
7.3 DPES User Interfaces ... 87
7.4 Re-Run Environment (T6.6) ... 88
7.4.1 Linux Container Based Redeployment ... 88
7.4.2 Redeploying in a Virtual Machine ... 89
7.5 Status of Tools and Services Development ... 90

8 BUILDING, DEPLOYING AND TESTING DPES APPLICATION ... 92

8.1 Development Environment ... 92
8.2 Deployment Requirement ... 92
8.3 Installation and Testing ... 92

9 CONCLUSION AND FUTURE WORK ... 95

10 APPENDIX ... 96

10.1 DPES Screen Captures ... 96
10.2 IoT Use case ... 107
10.2.1 Introduction to the TIMBUS IoT Scenario ... 107
10.2.2 KaZSM business services ... 107

11 REFERENCES ... 109

List of Figures

Figure 2-1: Mapping of WP6 Tasks with TIMBUS Architecture 18
Figure 3-1: Block Level Storage Illustration 19
Figure 3-2: High level Object Storage Architecture [10] 21
Figure 3-3: Comparison of Object Storage Implementations [11] 22
Figure 3-4: Frequency of data usage Object vs. Traditional file storage [13] 22
Figure 3-5: File Level Storage Illustration 23
Figure 3-6: PDS Cloud High Level Architecture [22]. 24
Figure 3-7: Business Process Virtualisation [33] 27
Figure 3-8: Full Virtualisation 28
Figure 3-9: Hardware Assisted Virtualisation [35] 29
Figure 3-10: Para-Virtualisation [38] 29
Figure 3-11: OS-Level Virtualisation 30
Figure 3-12: Software Emulation of hardware 31
Figure 3-13: Stacked emulation over time [43] 32
Figure 3-14: Migrated emulation over time [43] 33
Figure 5-1: Application Architecture of DPES 50
Figure 5-2: Sample File Metadata population [60] 53
Figure 5-3: Sources of Business Artefacts 55
Figure 5-4: Packaging Structure of Business Artefacts 57
Figure 5-5: DPES High level Technical Architecture 60
Figure 5-6: DPES Web Service Deployment 61
Figure 5-7: DPES Repository Communication 62
Figure 5-8: Proxmox Web GUI 63
Figure 6-1: Phaidra High Level Architecture 68
Figure 6-2: Phaidra Architecture 68
Figure 6-3: RCAAP services and the infrastructure 69
Figure 6-4: Photos from a TIMBUS consortium trip to a dam in Portugal hosted by LNEC in year 2 73
Figure 6-5: Screen Capture of GestBarragens UI 74
Figure 6-6: LNEC DAM Monitoring Business Process 75
Figure 6-7: eHealth Technical Environment 79
Figure 7-1: Sequence Diagram of DPES and iERM Tool 83
Figure 7-2: DPES Storage Client Side Implementation 87
Figure 7-3 Linux Container Based Redeployment Model 88
Figure 7-4: Redeployment using Virtual machines 89
Figure 7-5: Virtual machine using Emulated CPU 90
Figure 8-1: Export DPES WAR file 93
Figure 8-2: Deploy DPES WAR file into Tomcat Server 94

Figure 10-1: DPES Login 96
Figure 10-2: Home Page 97
Figure 10-3: Create Project Screen 97
Figure 10-4: Network Scan 97
Figure 10-5: Software Environment Selection 98
Figure 10-6: Extract Software Environment 98
Figure 10-7: Cost Modelling Input 99
Figure 10-8: Cost Modelling Result 99
Figure 10-9: Edit Project Level Metadata 100
Figure 10-10: Edit Artefacts level Metadata 100
Figure 10-11: Storage Repository Selection 101
Figure 10-12: Storage Repository Containers creation 101
Figure 10-13: Preservation Process Initiated 102
Figure 10-14: DPES Management Screen Concept 102
Figure 10-15: Redeployment – General Options 103
Figure 10-16: Redeployment OS Selection 103
Figure 10-17: Redeployment Boot Options 104
Figure 10-18: Redeployment – Disk Options 104
Figure 10-19: Redeployment CPU Options 105
Figure 10-20: Redeployment – Memory Options 105
Figure 10-21: Redeployment – Network Options 106
Figure 10-22: Redeployment Summary 106
Figure 10-23: An Example of Storm Topology 108

List of Tables

Table 1 Comparison Table of Block, File and Object Storage [15] 23
Table 2: List of Software Containers Implementation (source: wiki [49]) 34
Table 3: Preserving Original User Interfaces 37
Table 4: Maintain Business Object Identity 37
Table 5: Authentic Object Rendering 38
Table 6: Business Process Event History 38
Table 7: Logging Preservation Events 39
Table 8: Preserving the Business Context 39
Table 9: Query and Browse Repository 40
Table 10: Support Business Process Change Update 40
Table 11: Redeploy for localised Analysis 41
Table 12: Detect Information Redundancy 41
Table 13: Digital Artefacts Format Database 42
Table 14: Digital Artefacts Format Recommendation 42
Table 15: Preserve Business Process 43
Table 16: Redeploy Business Process 43
Table 17: Preserve External Functionalities 44
Table 18: Maintain the Integrity of the Digital Objects 44
Table 19: Preservation and Redeploy Access Control 45
Table 20: Logical Preservation of Digital Artefacts 46
Table 21: Secure and Reliable Storage 46
Table 22: Self-Contained and functional Completeness 47
Table 23: Scalability of Preservation System 47
Table 24: Preserve the security functionality 48
Table 25: Ease of Use 48
Table 26: Non-intrusive and Transparent Capture 48
Table 27: RCAAP Business Process 70
Table 28: LNEC Business Process 75
Table 29: eHealth Software Resources 79
Table 30: DPES – iERM API calls 83
Table 31: DPES API to create project 84
Table 32: DPES API to update Project 84
Table 33: DPES API to get Status 84
Table 34: DPES API for Risk Alternative 86
Table 35: DPES API for Recommendation 86
Table 36: DPES API for user authorisation 86
Table 37: DPES Software Implementation Status 90

List of Acronyms

ACL	Access Control List
ADE	Adverse Drug Event
ADR	Adverse Drug Reaction
AFP	Apple Filing Protocol
AMD-v	Virtualisation technology provided by Advanced Micronics Devices
API	Application Programming Interface
AWS	Amazon Web Services
BP	Business Process
BPRM	Business Process and Resource Model
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Planning Process
CDMI	Cloud Data Management Interface
CIFS	Common Internet File System
CND	Computer Network Defence
COOP	Continuity Of OPerations
CPU	Central Processing Unit
CRUD	Create, Retrieve, Update, Delete
CSS	Cascading Style Sheets
CUDF	Common Upgradability Description Format
DIO	Domain Independent Ontology
DNS	Domain Naming System
DP	Digital Preservation
DPES	Digital Preservation Expert Suite
DPRM	Digital Preservation Resource Manager
DR	Disaster Recovery
DR-DOS	Digital Research - Disk Operating System
DSO	Domain Specific Ontology
ENSURE	Enabling kNowledge Sustainability Usability and Recovery for Economic value, an EU Project
ESX	Elastic Sky X , A VMWare virtualisation product for enterprises.
EVM	Emulation Virtual Machine
FCoE	Fiber Channel over Ethernet
FEA	Finite Element Analysis
FORTRAN	FORmula TRANsformation, A programming language
FTP	File Transfer Protocol
GSX	Graphics System eXtension
GUI	Graphical User Interface

HA	High Availability
HTML	Hyper Text Mark-up Language
HTTP	Hyper Text Transport Protocol
HW	Hardware
IA	Information Assurance
ID	Identity
IDH	Intel Distribution of Hadoop
iERM	Intelligent Enterprise Risk Management
IoT	Internet of Things
iSCSI	Internet Small Computer System Interface
IT	Information Technology
JAR	Java Archive
JHOVE	JSTOR/Harvard Object Validation Environment
JSTOR	Journal STORage
JVM	Java Virtual Machine
KaZSM	Kafka, Zookeeper, Storm, MQTT
KVM	Kernel based Virtual Machine
LAN	Local Area Network
LBA	Logical Block Addressing
LLM	Legal Life-cycle Management
LNEC	Laboratório Nacional de Engenharia Civil
METS	Metadata Encoding and Transmission Standard
MIME	Multipurpose Internet Mail Extensions
MQ	Message Queue
MQTT	MQ Telemetry Transport
MRP	Materials Replenishment Planning
NAS	Network Attached Storage
NFS	Network File Storage
NTFS	New Technology File System
OAIS	Open Archival Information System
OJS	Open Journal Systems
OS	Operating System
OSD	Object-based Storage Devices
OWL	Web Ontology Language
PC	Personal Computer
PDS	Preservation DataStores
PHP	Hypertext Preprocessor, a programming language for Web
PI	Preservation Identifier
PII	Personally Identifiable Information
PREMIS	International standard for Preservation Metadata

PRONOM	Online registry of Technical information developed by National library, UK
QEMU	Quick EMUlator
RAID	Redundant Array of Inexpensive Disks
RCAAP	Repositório Científico de Acesso Aberto de Portugal
REST	REpresentational State Transfer
S3	Simple Storage Service, part of Amazon Web Services
SAN	Storage Area Network
SARBOX	Sarbanes–Oxley Act
SARC	Serviço de Alojamento de Revistas Científicas
SARDC	Serviço de Alojamento de Repositórios de Dados Científicos
SCAPE	SCAlable Preservation Environments
SCSI	Small Computer System interface
SDK	Software Development Kit
SMB	Server Message Block
SOAP	Simple Object Access Protocol
SOTA	State Of The Art
SW	SoftWare
UI	User Interface
UTF	Unicode Transformation Format
UVC	Universal Virtual Computer
VA	Virtual Applicance
VB	Visual Basic
VE	Virtual Environment, Virtualisation Engine
VMFS	VMWare File System
VMM	Virtual Machine Manager
VPN	Virtual Private Network
VPS	Virtual private Server
VT	Vanderpool Threading, virtualisation technology provided by Intel Corporation
WAN	Wide Area Network
WAR	Web Archive
WINE	Windows EMUlator
WP	Work Package
XEN	A native hypervisor (Virtual Machine Manager)
XML	eXtensible Markup Language

1 Executive Summary

Digital Preservation (DP) has been extremely popular in the area of electronic libraries, eJournals, preserving arts and national heritages since the evolution of computing technologies. This is changing and there is a growing need to digitally preserve Business Processes (BP) of almost all domains ranging from individual citizens to eCommerce, eGovernment, manufacturing, health, and finance. The reasons for these range from the sentimental value of data in the case of citizens to legal obligation, recovery from operational impacts and long term access to the data contained within the BP in the case of businesses. The TIMBUS project focuses on the requirements of businesses which profoundly depend on vast IT infrastructures to successfully manage and run complex business environments.

The TIMBUS project has been harvesting the knowledge and best practice which exists in memory institutions and in the digital preservation community. The objectives of the TIMBUS exploitation plans includes an investigation of what is needed in order to bring these practices to wider audiences which may not traditionally have possessed a technical long term preservation capability. This was also a request of the reviewers in year 2. An important consideration in this is that TIMBUS is built on feasibility and cost-benefit analysis which recommends which aspect of a business process should be preserved and how to preserve them in conjunction with Enterprise Risk Management (ERM) and Business Continuity Management (BCM) best practices. Within the TIMBUS project, work package 6 (WP6) focuses on developing a set of tools and methodologies to digitally capture and formalise business process both technical and organisational level and redeploy in the later time.

This deliverable therefore presents the novel work and the current status of Tasks T6.4 (Business Process Virtualisation Storage Manager), T6.6 (Business Process Rerun/Integration Environment) and T6.7 (Simulated Test Bed). Tasks 6.4 and 6.6 are highly correlated as the preservation of a business process cannot be carried out without a detailed understanding of how re-deployment will work and vice versa.

This deliverable builds on the last iteration, D6.4 which had introduced the Digital Preservation Expert Suite (DPES). This document contains the results of our state-of-the-art assessment and how virtualisation and emulation can support the requirements of a Business Process preservation environment. The latest enhancements to the DPES are detailed in this document. In particular, integration between the DPES and other TIMBUS components has taken large strides since M30. The technical architecture has not been radically altered, but the interface specifications and implementation status of the WP6 tools has advanced and the project is on a good trajectory to meet its goals.

Additionally, this deliverable relates how the DPES is supporting the TIMBUS use cases and it details the IoT (Internet of Things) scenario which Intel presented at the year 2 review. This work was carried out under the auspices of Task T6.7 in year 3 of TIMBUS.

2 Introduction

Digital Preservation is the persistent archival of digital assets for future access and reuse, irrespective of the underlying platform and software solution. Today's archival solutions are costly and research is needed into easing the ability of future users to easily consume that data. The problem arises because the hardware, the media, the operating systems, the software and the file formats in which our data is stored becomes obsolete very quickly. The problem gets magnified when it's an organisation running a business process which relies on various technologies, systems and tools. In order to minimise these problem a solution is needed to model the business processes comprehensively at various function layers and collect or characterise those resources that are identified as risk for long term preservation and access. At present there is no ready solution available to prevent the disruption of business processes due to the concept of the Digital Dark Age [1] or any other unforeseeable events. Furthermore, the data stored in archives today needs to be readily accessible, and intelligible, in 20, 50 or 100+ years time. Business process preservation addresses these by preserving the tools and the software environments required to access and interact with the data in its native environment. By addressing these problems today, the EU co-funded TIMBUS project is addressing these problems and helping to ensure that we avoid a digital dark age so that our know-how and our legacy will be available to future generations.

2.1 Objective

The purpose of this deliverable is to update on the current status of Tasks T6.4, T6.6 and T6.7 in M36. The core outcome from these tasks are to design and develop an integrated prototype software application called DPES (Digital Preservation Expert Suite) to collectively demonstrate the capabilities of TIMBUS framework by integrating tools and services that are being developed in other tasks. This document is a reflection of work done towards the development of prototype software application (DPES).

The structure of the document is such that it starts with State of the Arts (SoTA) study in Chapter-3 where it discusses the three major IT technologies namely Storage, Virtualisation and Emulation. In Chapter-4, it is presented with summary of technical and non-technical requirements gathered from TIMBUS use cases for designing the prototype software. Software design, architecture and functional components are detailed in Chapter-5 to satisfy the requirements described in the previous chapter. Chapter-6 details the preservation strategies for each use cases using prototype software applications that are being developed in the WP6. Chapter-7 & Chapter-8 are focusing on actual software implementation (since last deliverable D6.4 in M30), user guide for downloading, building and deploying the DPES prototype software. Conclusion and future works are discussed in Chapter-9 to provide the readers with overall status and expected future work from the above-mentioned tasks. Finally Chapter-10 & Chapter-11 contains appendix and references respectively relating to this deliverable document.

The following sections introduce each task in brief.

2.1.1 Virtualisation and Storage Manager (Task 6.4)

The goal is to provide the user with a set of tools to execute a digital preservation action on a business process and to be able to redeploy it in the future. This task has to date focused on feasibility analysis for digitally preserving the selected BP, sourcing various digital objects of BP process such as compute environments, documents describing business, service dependencies (context model), etc. using preservation identifier recommendations. The work to date uses the information about the BP components supplied by the TIMBUS context model (cf Deliverables D4.3 and D4.9 in M24 and M36, resp.) to generate and populate metadata in a standard XML schema to describe the digital objects which are required to be preserved and as a historical record of the actions carried out on these objects during their entire digital curation lifecycle.

2.1.2 Rerun/Integration Environment (Task 6.6)

Developing a system and integrating with tools developed in T6.4 to support BP redeployment in the future time in a virtualised or non-virtualised environment. This system will guide the users through the entire redeployment process. This will enable creating virtualised redeployment environment fulfilling hardware specification requirement by integrating with third party virtualisation server platform, redeploying business processes and services, resolving dependencies technically and functionally between services and business application. Further, this task integrates with Windows and Linux dependency extractors to resolve system technical dependencies, such as adding missing libraries or packages (.dll, .deb, .rpm, and etc.), restoring registry entries, configuration and etc. The business processes defined within context model will be used to interpret and resolve the dependencies (functional) between services when the redeployment is performed.

2.1.3 Simulated Test Bed (Task 6.7)

This task is providing an infrastructure platform for the deployment of the tool suite developed in work-package 6. Its goal is to allow the entire tool suite to be operated in an integrated manner. Task T6.7 will run through to the end of the TIMBUS project. It is an implementation of the reference architecture detailed in deliverable D5.3 which will be adapted over the course of the project based on the output of the TIMBUS work packages. The purpose of the T6.7 test bed is therefore to be able to host the tools and systems to allow the demonstration of the full preservation life-cycle. This requires that a source BP be available in the test bed and that there is sufficient storage and computing capacity available that can support the re-deployment.

Intel used some of the effort in Task T6.7 to analyse the long-term preservation requirements of an end-to-end IoT system which is one of the main exploitation initiatives Intel is undertaking because it provides a potential avenue to include TIMBUS concepts in a planned product. More details on this are available in the deliverable D2.3: Exploitation Plan Iteration 3. It also allows us to test the TIMBUS tools on an end-to-end use case whose backend is hosted in Amazon AWS [2] as a learning exercise.

2.2 Scope

TIMBUS aims to preserve business processes using digital preservation. It analyse the risk and identifies resources which are vulnerable to risks in the near future, and then uses digital preservation as a solution to mitigate the risk. For doing so, it starts with analysing business processes to identify the risk areas and estimate potential impact to business process in terms of monetary value.

Digital preservation solutions are deployed upon comparing the cost and feasibility of preservation and risk value. Business process digital preservation utilises the context model and various extractor tools to gather business contextual information, technical resources and the dependencies of the business environment. All the information including digital artefacts are analysed further and alternates are identified if applicable and migration is performed for long term sustainability. Then the digital resources are organised and stored in a secure repository for future access and redeployment.

This deliverable – *D6.6 Business Process Preservation Test Bed* - will deliberately focus on T6.4, T6.6 and T6.7, and forms iteration to the deliverable document D6.4 - Virtualisation Manager which was submitted in month M30.

2.3 Technical Context

WP6 (Intelligent Tools and technologies to Support Preservation of Business Processes) within TIMBUS implements a set of tools to demonstrate the preservation of business processes and redeploying back into a virtualised for proving the concept of business process digital preservation. Figure 2-1 shows the various tasks responsible for implementing the tools via the architecture diagram.

The context modelling techniques developed as part of WP4: Process and Methods for Digitally Preserving Business Process are used to build a context model using the Domain Specific Ontology (DSO) and Domain Independent Ontology (DIO) to describe the business process and the contextual information surrounded by business scenarios. The Intelligent Enterprise Risk Management System – iERM, developed as part of T6.1, imports the Context Model and performs risk analysis on each resource supporting the business processes. Upon identifying and quantifying the risk, iERM seek to find a cost economic solution to mitigate the risk. DPES (T6.4, T6.6, and T6.7) receives a request from iERM with the list of resources of each business process which are under risks, seeking for possible Digital Preservation as a solution to eliminate the risk.

DPES first analyses the iERM request with the help of the Preservation Identifier (PI) to understand the available alternative options for the selected resources. Using PI's recommendation, the DPES estimates the cost of digitally preserving the business process to eliminate the risk in future. If the iERM business decision is to pursue preservation as the desired solution, the DPES is invoked to digitally preserve the Business Process. As part of the Digital Preservation process, DPES utilises the output from context extractor (T6.5) and reasoning tool (T6.2) to precisely capture the list of business software and hardware environment components and their dependencies in metadata form. Later, the DPES will source the actual digital artefacts listed in the metadata using the Digital Preservation Resource Manager (DPRM), a component within DPES. An artefact in the TIMBUS context is simply a file which must be preserved in the long-term storage repository. The prototypes under development in the TIMBUS project, of which DPES is one, use the simulated Test Bed T6.7 and Re-deployment Environment (T6.6) to store and redeploy the digitally preserved business process at a later stage.

The following are the list of tasks within WP6:

- T6.1: Intelligent Enterprise Risk Management System
- T6.2: Dependencies, Reasoning and Constraints
- T6.3: Regulatory/Contractual Life Cycle Management
- T6.4: Business Process Preservation Manager
- T6.5: Context Information and Metadata Capture Tool
- T6.6: Business Process Rerun/Integration Environment
- T6.7: Simulated Test Bed

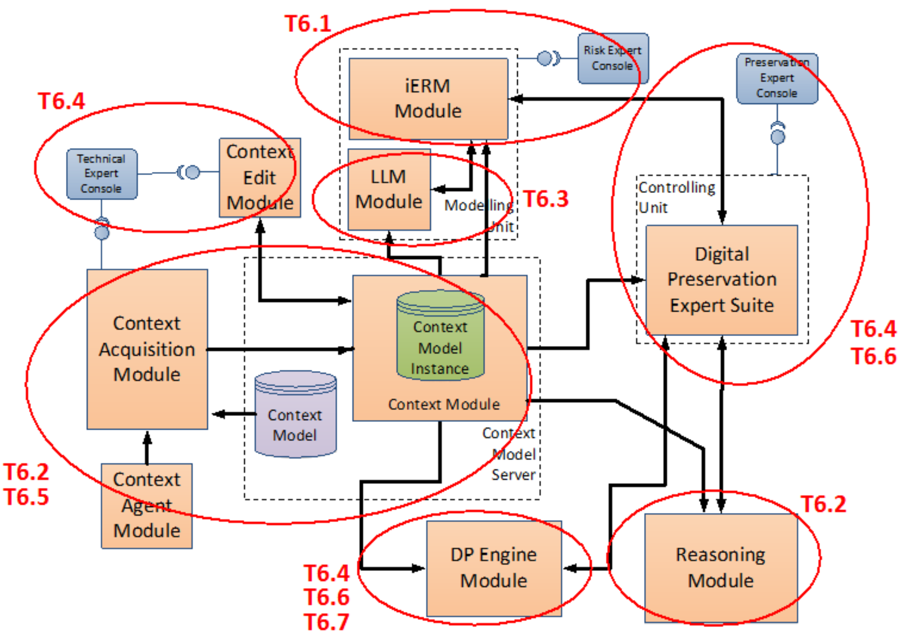

Figure 2-1: Mapping of WP6 Tasks with TIMBUS Architecture

3 State Of The Art

This chapter reviews, and updates the standards and technologies in addition to the SOTA review submitted in M30 D6.4 deliverable [3]. This chapter will also compare and contrast the technologies of similar nature to gain deep understanding before making a decision to select which particular technology to integrate within TIMBUS Preservation and Redeployment environment and also for TIMBUS test bed infrastructure.

3.1 Storage

In the technology sector, storage or data storage refers to technologies consisting of computer components and recording media used to retain digital data. It is a fundamental component of modern computer systems. Storage is considered as a critical system while designing any long term preservation solution. Timbus Deliverable D5.3 [4] submitted in M24 presented detailed reference architecture with various storage mediums (disk, taps and etc.) and their pros and cons with regard to designing effective storage solutions for long term digital preservation. This SoTA review will examine various storage related specifications and protocols that are currently in use. It will also discuss some of the storage service models currently being offered by companies such as Amazon S3 [5] and Dropbox [6].

3.1.1 Block Level Storage

In block level storage, raw volumes of storage are created and each block can be controlled as an individual hard drive. These Blocks are controlled by server based operating systems and each block can be individually formatted with the required file system [7].

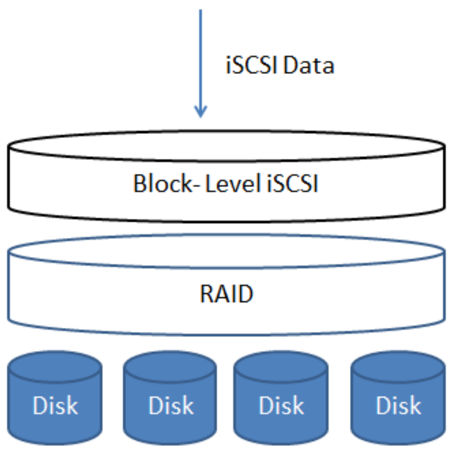

Figure 3-1: Block Level Storage Illustration

In its most basic form, think of block level storage as a hard drive in a server except the hard drive happens to be installed in a remote chassis and is accessible using Fibre Channel or iSCSI, Figure 3-1 illustrate the high level structure of block storage system. Block level storage is usually deployed in SAN or storage area network environment.

This storage offers boot-up of systems which are connected to them as each storage volume can be treated as an independent disk drive and it can be controlled external operating system. Block level storage can also be used to store files and can work as storage for special applications such as databases, Virtual machine file systems and so on.

3.1.2 Object Storage

Object storage [8] Object storage (also known as object-based storage [9]) is a storage architecture that manages data as an object. Other storage architectures use file systems which manage data as a file hierarchy while block storage manages data as blocks within sectors and tracks on the storage medium. In Object storage, each object typically includes the data itself, a variable amount of metadata, and a globally unique identifier. Object storage can be implemented at multiple levels, including the device level (object storage device), the system level, and the interface level. In each case, object storage seeks to support capabilities not addressed by other storage architectures, such as interfaces that can be directly programmable by an application, namespaces that can span multiple instances of physical hardware, and also enables data management functions such as data replication and data distribution at object-level granularity.

Figure 3-2 shows the high level architecture of an object storage system. One of the design principles of object storage is to abstract some of the lower layers of storage away from the administrators and applications. Thus, data is exposed and managed as objects instead of files or blocks. Abstraction is always interesting from a long-term preservation perspective because it offers potential protection from underlying changes in the storage medium. Objects contain additional descriptive properties which can be used for better indexing or management. Administrators do not have to perform lower level storage functions such as constructing and managing logical volumes to utilise disk capacity or setting RAID levels to deal with disk failure.

Object storage also allows the addressing and identification of individual objects by more than just file name and file path. Object storage adds a unique identifier within a bucket, or across the entire system, to support much larger namespaces and eliminate name collisions.

3.1.2.1 Separation of metadata and data:

Object storage explicitly separates file metadata from data to support additional capabilities. Additional metadata is particularly important for long-term preservation; as the archive grows, it will become harder to locate the specific objects the future designated user is interested in. This is an important preservation consideration because we must take in to account how future users are going to retrieve the artefacts when we perform the preservation. This section of the deliverable explains some of the ways in which object storage solutions can help with this:

- Additional metadata to capture application-specific or user-specific information for better indexing purposes. This is also a required step in performing preservation because it provides an ability to more easily find the artefacts stored in the repository.
- Additional metadata to support data management policies (e.g. a policy to drive object movement from one storage tier to another).
- Scalability of metadata and business artefact storage nodes is important over time as the volumes of such information will grow. The system needs to be scalable to support such requirements.
- Unified access to data across many distributed nodes and clusters.
- Centralised management of storage across many individual nodes and clusters.
- Optimisation of metadata storage (e.g. database or key-value storage) versus data storage (e.g. unstructured binary storage).

3.1.2.2 Programmatic data management:

Object storage provides programmatic interfaces to allow applications to manipulate data. At the base level, this includes CRUD functions for basic Read, Write, Update and Delete operations. Some object storage implementations go further, supporting additional functionality such as object versioning, object replication, and movement of objects between different tiers and types of storage. Figure 3-2 shows the object storage architecture detailing interfaces to access metadata and object nodes and link between them. Most API implementations are ReST-based, allowing the use of many standard HTTP calls.

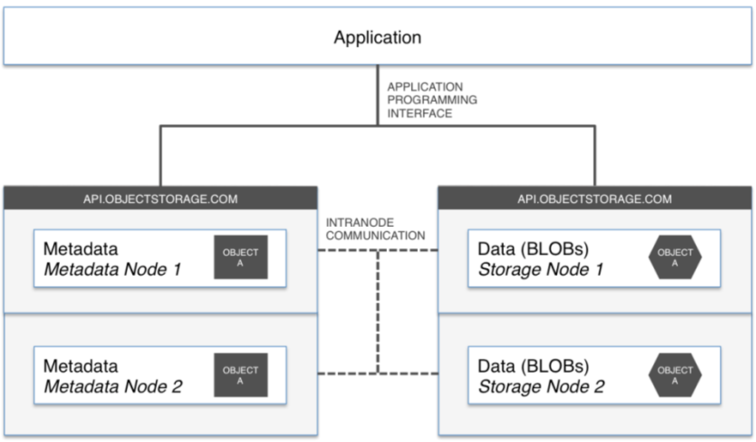

Figure 3-2: High level Object Storage Architecture [10]

There are many API alternatives for accessing object storage; Figure 3-3 shows the cross-protocol compatibility for object storage. Having cross-protocol support enables larger numbers of customers to start adopting object storage implementations with the help of compatible third-party applications and tools which also contain support for legacy applications.

Examples of this include the well-known providers such as AWS S3, Google Cloud storage, OpenStack Swift, and Windows Azure Storage to name a few.

Interface	S3	OpenStack Swift	CDMI	HDFS	WebDAV
Object Read	Supported	Supported	Supported	Supported	Supported
Object Write	Supported	Supported	Supported	Supported	Supported
Object Delete	Supported	Supported	Supported	Supported	Supported
Metadata Operations	Object Metadata Bucket Metadata	Object Metadata Container Metadata	Object Metadata Container Metadata	File/Directory Attributes	PROPPATCH PROPFIND
Authentication	Keyed HMAC-based AWS Auth	Token-based Auth	Traditional HTTP Auth Mechanisms	Host OS Process Owner	Traditional HTTP Auth Mechanisms
ACLs	Bucket & Object ACLs Bucket Policy IAM Policy	Container ACLs (Read/Write only)	Metadata Property "cdmi_acl"	POSIX File/Dir Permissions	ACL: RFC3744
Versioning	Enabled per Bucket Each unique write assigned a revisionId	Enabled per Container X-Versions-Location header specifies target for previous versions	*CDMI Extension	N/A	DeltaV: RFC3253
Large Object Support	Multipart Upload	Dynamic Large Object Static Large Object	HTTP Range, X-CDMI-Partial Headers	N/A	N/A
Provisioning	Create Bucket Delete Bucket	Create Container Delete Container	Create Container Delete Container	Create Directory Delete Directory	MKCOL Ext. MKCOL: RFC5689

Figure 3-3: Comparison of Object Storage Implementations [11]

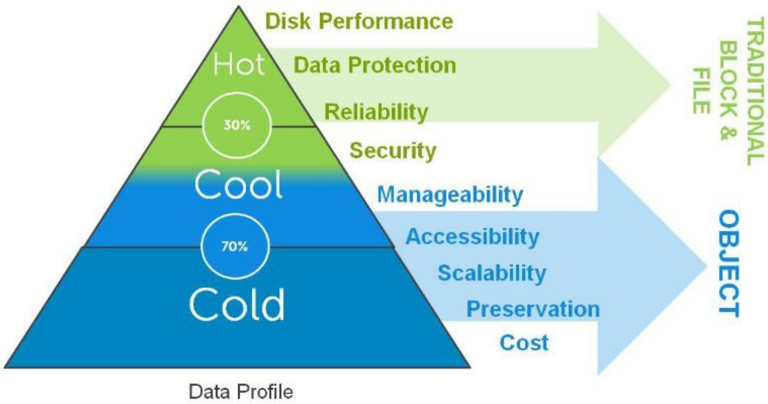

Figure 3-4: Frequency of data usage Object vs. Traditional file storage [13]

In general object storage works best when large number of unstructured data files need to be stored, and also suitable archiving when data is relatively static and not frequently accessed as shown in Figure 3-4. This isn't a restrictive criterion for object storage use cases. Researchers found that 70% of data that is generated is never accessed after its initial creation and remains static, while 20% is semi-active [12]. However, 10% of all data is actively used, and it is for this data that traditional file systems, such as NAS, are best suited.

3.1.3 File Level Storage

This storage technology is most commonly found in hard drives, NAS systems and so on. In file level storage, the storage disk is configured with a protocol such as NFS or SMB/CIFS and the files are stored and accessed from it in bulk [14]. Figure 3-5 shows the file level storage model where storage disk are formatted into specific file system and then consumed by file sharing protocols such SMB, NFS, FTP and AFP.

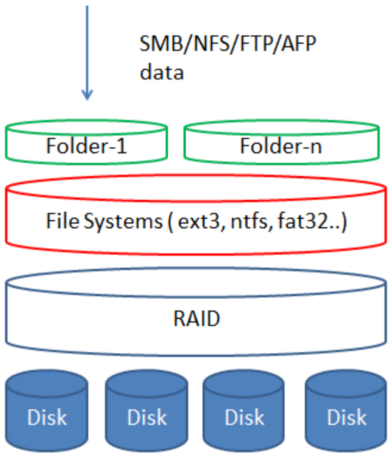

Figure 3-5: File Level Storage Illustration

3.1.4 Object, Block and File Storage Comparison

The Table 1 shown below compares some key features among the block, file and object storage systems.

Table 1 Comparison Table of Block, File and Object Storage [15]

	Block Storage	File Storage	Object Storage
Reference mechanism	Logical block address (LBA) on a device or logical unit	Named file in a directory in a tree of named directories	Object name or key on an object server
Atomic unit of access	4 Block, 512 or 4096 bytes	Byte	Object value
Access semantics	write, read, trim	open, write, read, position, insert, append, close, etc.	put, get, delete

	Block Storage	File Storage	Object Storage
Relationship semantics	LBA x+1 is the fastest block to access after LBA x	Files in the same directory are related	Key x and Key y are related for "sequential" keys x and y where there is no other key z such that x < z < y.
Scaling limits	Limited to blocks on a single device	Limited to files in a single file system	Limited to object on a single object server

3.1.5 Storage Service

Cloud storage is one arm of the larger area of networked enterprise storage [16] where data is stored in virtualised pools which are generally hosted by third parties. Storage services are managed by hosting companies such as Google, Amazon and EMC, to mention some of the larger ones, by deploying enterprise scale datacentres across the multiple geographical regions. Cloud storage has the same characteristics of cloud computing in terms of agility, scalability, elasticity and multi-tenancy. Figure 3-6 shows the PDS Cloud Architecture which an OAIS-based preservation-aware storage service employing multiple heterogeneous cloud providers.

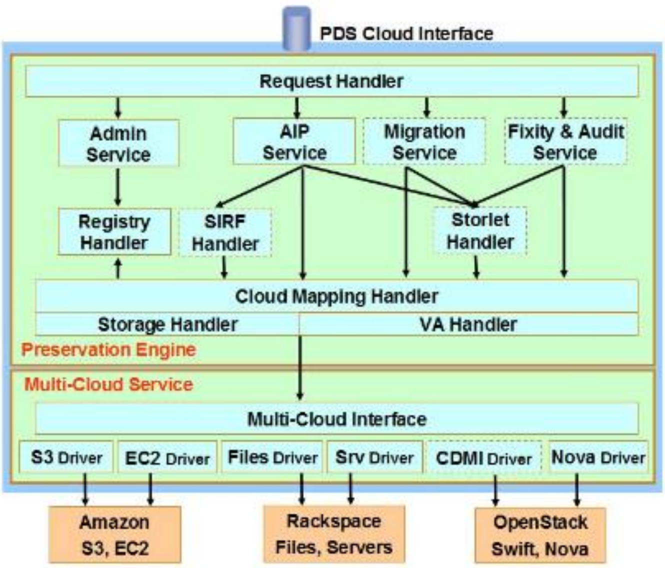

Figure 3-6: PDS Cloud High Level Architecture [22].

PDS cloud is being developed as an infrastructure component of the European Union ENSURE [17] project. The architecture shows a solution for the integration of multi-cloud storage as well as compute needs (rendering the preserved object). Some of the characteristics of cloud based storage models are listed below.

Characteristics of Cloud based Storage Services:

- Made up of many distributed resources, but still act as one pooled storage – often referred storage clouds [18]
- Highly fault tolerant through redundancy and distribution of data
- Highly durable through the creation of versioned copies
- Typically eventually co insistent with regard to data replicas [19].

Advantages of Cloud based Storages:

- Companies need only pay 'on-demand' for the storage they actually use. No initial capital investment is needed for building storage infrastructure.
- Storage availability and data protection is intrinsic to the object storage architecture, so depending on the application, the additional technology, effort and cost to add availability and protection can be eliminated [20].
- Storage maintenance tasks, such as purchasing additional storage capacity, are offloaded to the responsibility of a service provider.
- Cloud storage provides users with immediate access to a broad range of resources and applications hosted in the infrastructure of another organisation via a web service interface [21].

Dis-advantages of Cloud Storages:

- Security and privacy of the data could be compromised when data is distributed at more locations which increase the risk of unauthorised physical access and damage to the information.
- Performance, reliability and availability depend on wide area network and the bandwidth of the internet service. Increase the overall network traffics in the WAN (wider area network).
- Some of the risk mitigating action taken in the cloud storage system (such as encrypting the data to avoid being read by others during transmission or unauthorised access) adds additional complexities regarding intelligibility of the data. Loosing of key may lead to permanent inaccessibility to our own data
- Cloud storage provides can go bankrupt, expand and change their products/services. Companies can be purchased by other larger companies with different business policy and interest. New companies can be headquartered in or move to a country that negates compliances with export restriction and thus necessitates a move.

Cloud based Object Storage is recommended for long term preservation need [22] for following reasons [23].

- The operating costs are higher when using in-house storage,

- Cooling consumes about 45% of the power delivered to data center,
- Storage consumes 13% of total data center power, with 15% for servers.
- Cloud-based storage is 74% less expensive than-in house [24].

3.1.6 Summary

There are no doubts that storage is an essential part of long term business process preservation systems, but selecting the right storage model or services is extremely important for long term reliability, performance and the overall success of the preserved environment. Business process preservation involves the gathering of enormous amounts of data related to specific business environments, most of this data is *unstructured* and diverse in formats such as documents, spreadsheets, graphics, images, videos, executable software application and libraries, etc. Conventional file based storage methodologies are not enough to manage heterogeneous digital data for long term preservation and access need. Organisations require the data be readily accessible for business, regulatory and compliance needs after long period as well as regular basis. Unstructured data growth is expected to continue at a compound annual growth rate estimated to exceed 60 percent [25]. Other research suggests that the amount of digital information will double every 18 months, with 95% of this coming from unstructured data, and only the remaining 5% being driven by traditional structured data [25]. This growing data volume serves to reinforce the conclusion that object based storage has several key advantages when used in a long term archival solution and sstorage managers are looking at new ways to cope with this.

The statistics presented in the previous section provides us a clear indicator for cost economics scale of cloud based storage model, and transfer of IT infrastructure responsibility to external providers, adoption of an industry standards for cloud storage such as CDMI (Cloud Data Management Interface), RESTful principle (self-descriptive URL) in interface design are among the other factors that makes cloud storage more attractive for long-term preservation need [27]. In summary cloud based object storage model is the best suitable option for long-term digital preservation based on above facts and figures.

3.2 Virtualisation

IT Virtualisation [28] is the abstraction of IT resources that masks the physical nature and boundaries of those resources from resource consumers or users. An IT resource can be a server, a client, storage, networks, applications or an operating system. Essentially, any IT building block can potentially be abstracted from resource consumers or users. This section of the deliverable will discuss the usefulness of virtualisation to TIMBUS in some detail.

Within TIMBUS, virtualisation is considered as a potential enabling approach for preserving business processes for long term access and realisation needs. The main enabler of most contemporary Virtual processes is IT, which, following O’Brien (2002, p.7) [29], is defined as “computing hardware, software, communications networks, and data resources that collect, transform, and disseminate information”. Figure 3-7 (Martha et.al 2004) shows the various IT Technologies that are currently available to virtualise many parts of the business process environment. Successful long-term preservation of business process depends on the ability to capture the context of the existing business process, safely store the digital artefacts supporting the business process and maintaining integrity of those digital artefacts over extended period of time, and allow changes of underlying technology while retaining accessibility of these business environments. Context capturing and modelling are outside the scope of this document as Context modelling is addressed in separate tasks than this deliverable deals with, namely those can be found in task T6.2; in this document we will focus on the later points. In order to achieve those tasks for successful long term preservation Sangeeta Dhamdhere has outlined some preservation approaches in her book [30] as below.

The Digital Preservation Tactics:

- Technology preservation
- Technology emulation
- Digital information migration

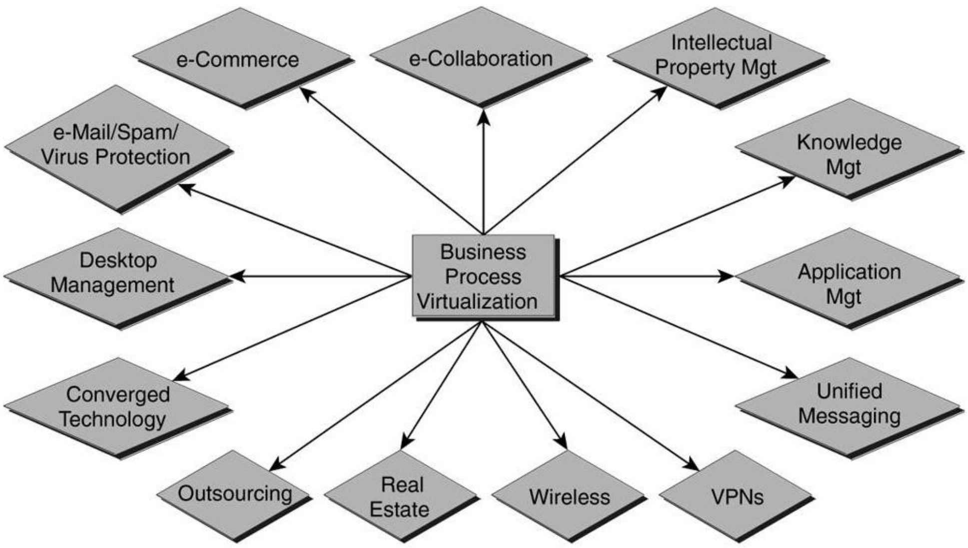

Figure 3-7: Business Process Virtualisation [33]

Virtualisation technologies can significantly help to simplify the process of preserving business application and legacy software tools by building the virtual machine with the help various hypervisor [31] platform available today. Although virtualisation is primarily targeted at allowing the hardware consolidation, creation of affordable high-availability systems as well as easier-to-administer datacenter system [32], virtualisation has brought with it the ability to logically detach complete “self-sustained” software environments, which include operating systems, applications and the digital data of any format or type, from the hardware and/or software on which it depends.

3.3 Types of Virtualisation

3.3.1 Full or Native Virtualisation

Full virtualisation is a technique used to obtain a virtual machine environment, namely, one that is a complete simulation of the underlying hardware. Full virtualisation requires that every salient feature of the hardware be reflected into one of several virtual machines – including the full instruction set, input/output operations, interrupts, memory access, and whatever other elements are used by the software that runs on the bare machine [34], and that is intended to run in a virtual machine. Figure 3-8 illustrates the building blocks within the full virtualisation technique.

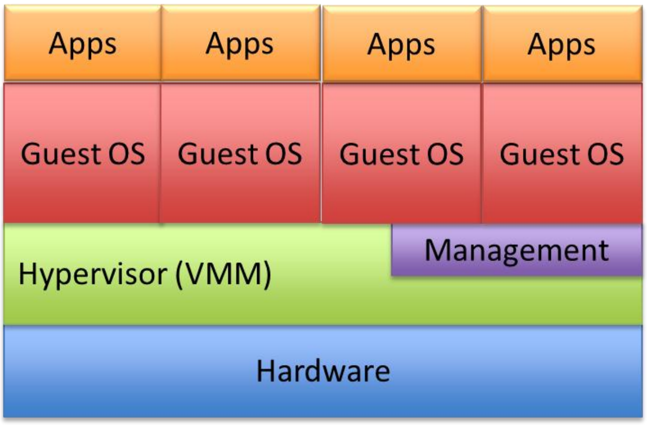

Figure 3-8: Full Virtualisation

Examples: Mac-on-Linux, Parallels Desktop for Mac, Parallels Workstation, QEMU, and XEN/Virtual iron.

3.3.2 Hardware-assisted Virtualisation

An example hardware-assisted virtualisation technique is shown in Figure 3-9 using VMWare virtualisation software, the VM has its own hardware and allows a guest OS to be run in isolation. It is also know a platform virtualisation approach that enables efficient full virtualisation using help from hardware capabilities, primarily from the host processors. Hardware-assisted virtualisation was added to x86 processors (Intel VT-x or AMD-V) in 2006.

Example: VMware Fusion, Parallels Desktop for Mac and Parallels workstation.

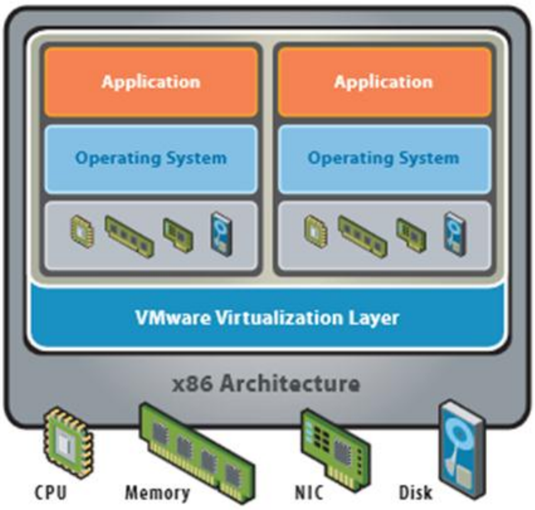

Figure 3-9: Hardware Assisted Virtualisation [35]

3.3.3 Para-Virtualisation

In Para-Virtualisation technique virtual machine does not simulate hardware, but special API (Hypercalls [36]) calls are made between modified guest OS and the underlying hypervisor platform. In this virtualisation technique hypercalls are trapped by the hypervisor and serviced to the request. The main limitation of paravirtualisation is the fact that the guest OS must be tailored specifically to run on top of the virtual machine monitor (VMM), the host program that allows a single computer to support multiple, identical execution environments [37]. Figure 3-10 shows the paravirtualisation concept where modified OS are running directly on top of hypervisor to support the original application.

Examples: Xen, VMWare ESX Server

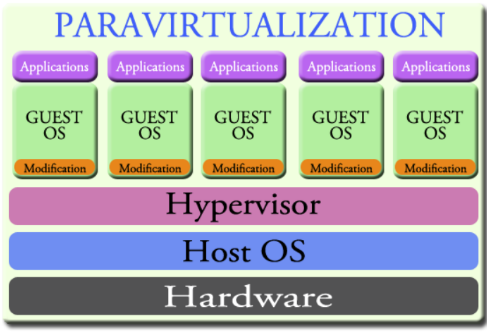

Figure 3-10: Para-Virtualisation [38]

3.3.4 OS-Level Virtualisation

Operating system-level virtualisation is a server virtualisation method where kernel of an operating system allows for multiple user-space instances. Such instances (often called containers, virtualisation engines (VE), virtual private servers (VPS) or jails) may look and feel such as a real server, from the point of view of its owner. Using OS-Level virtualisation technique, virtualisation are performed on the physical server at the operating system level, enabling multiple isolated and secure virtualised servers to run on a single physical server as shown in Figure 3-11.

On UNIX-such as operating systems, this technology can be thought of as an advanced implementation of the standard *chroot [39]* mechanism. In addition to isolation mechanisms, the kernel often provides resource management features to limit the impact of one container's activities on the other containers.

Example: OpenVZ, Solaris Containers, FreeBSD Jails, Linux-VServer, Chroot and Docker.

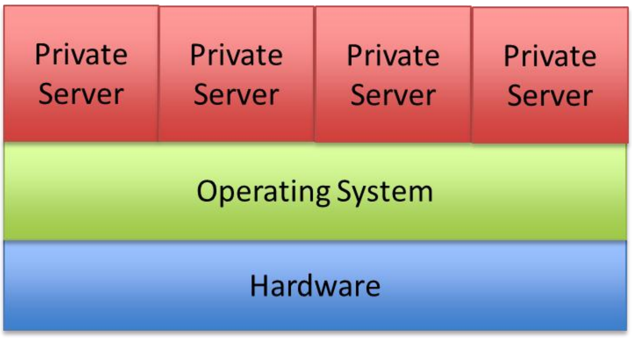

Figure 3-11: OS-Level Virtualisation

3.3.5 Application Level Virtualisation

Application are given its own copy of components that are not shared such as own registry files, global objects. Virtual environment (VE) preventing shared component from being conflicted during operation of these applications. Example: JVM, Rosetta on Mac (also emulation), WINE.

3.4 Emulation

Emulation is an imitation of behaviour of a computer or other electronic system with the help of another type of computer. Virtual machines emulates/simulates complete hardware such that an unmodified guest OS can be run on different hardware setup. Emulation technique helps to achieve various level of virtualisation i.e. Full or Native virtualisation, Application-level virtualisation. Examples: WINE, VirtualPC for Mac and QEMU.

3.4.1 Emulators in Digital Preservations

Emulation is an extremely versatile and durable solution [40] for retaining access to many kind of digital content. Emulators can help to preserve old or obsolete original digital environments which are supporting the current business process. It is interesting point to note from von Suchodoletz [41] that in general emulating hardware is a much more effective and efficient approach when compared to emulating the software applications itself which increases the number of emulation requirement.

3.4.2 Preserving the Emulated Hardware

Digital preservation depends on well tested and stable virtual machine hardware; because of this the emulator plays a crucial role. Emulator bridges the widening gap between the digital past and todays current environments. As an operating system likes DRDOS 6.0 or Windows 3.1 is not maintained any more, there will be no support for newer hardware such as actual gigabit net-work adaptors or high resolution 3D graphic cards. Thus, the reproduction of a certain 20 year old software environment depends on emulators such as QEMU [42] providing exactly the hardware configuration in use in those times.

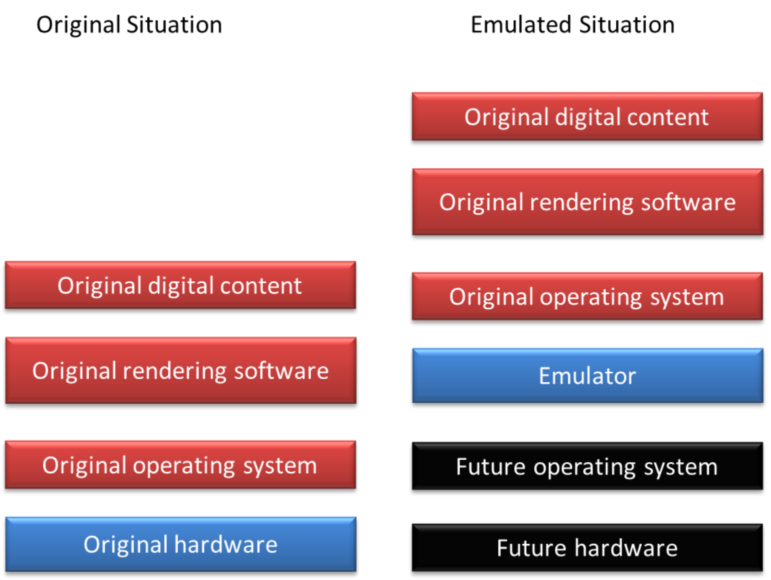

Figure 3-12: Software Emulation of hardware

In order to be able to deploy such an environment in 20 years from now the emulated hard-ware must remain compatible to run the targeted application across the time and kept updated with recent hardware where emulators can be run. Figure 3-12 illustrates that the original and emulated situation where existing hardware is emulated which can be preserved to run unmodified original OS and applications in the future HW and SW environment.

3.4.3 Digital Preservation Approaches

An emulation strategy for digital preservation needs to meet certain key requirement for successful implementation. The main criteria is how to make sure that an emulator developed today, can still execute in the future technology. The emulators that have been developed for the purposes as described in the earlier section, are dependent on the host platform and cannot simply be used for preservation purpose because we have less visibility about the future host platform, additionally the accuracy of an emulated software in terms of timing and performance are also important to render digital objects in an authentic way, as it has to be verified while the original platform still works [43].

As per van der Hoeven [43] emulation in digital preservation context can be described using three different approaches.

3.4.3.1 Stacked emulation

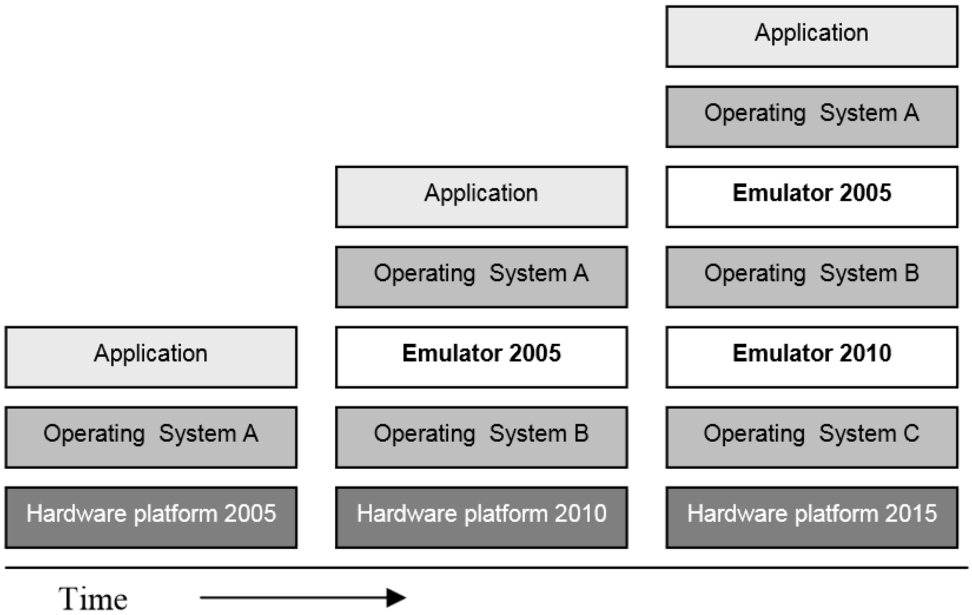

Figure 3-13: Stacked emulation over time [43]

In Emulation technique, a specific hardware platform is emulated directly on top of a current platform and OS. The advantage of applying this kind of emulation is the efficiency that can be gained by specifically developing the emulator that can run directly on top one particular host platform. This generally enhances performance and functional behaviour of the emulated platform. However, it lacks compatibility with other platforms in the long due to major technology changes. To keep these emulators running on subsequent platforms over time, stacked (or layered) emulation should be applied: the platform an emulator is built for in its turn needs to be emulated when it becomes obsolete as shown in Figure 3-13.

Stacked emulation generally results in a loss of performance, but other drawbacks exist as well. Each emulator strongly depends on its underlying environment, which means that functionality that is not available on the host platform and possible intermediate emulators cannot be offered on the target platform. Moreover, if one of the in-between emulators is corrupted or missing, the chain relying on this emulator is inevitably lost, resulting in inaccessible applications and documents [43].

3.4.3.2 Migrated emulation

In this approach, the emulator once created is adopted to a different environment [44]. A source code of an emulator developed once for one particular system can be periodically recompiled and run on newer host environment, when for instance the old OS becomes obsolete. The emulator is translated on a new host environment, but the translation process requires compiler that is able to compile the source code emulator written in "*language X*" binary executable to run on new platform. Figure 3-14 shows how the original emulator is periodically migrated to subsequent environment. Migrated emulation approaches are generally more suitable for preserving the software application when we have access to source code and compiler platform for new system [45].

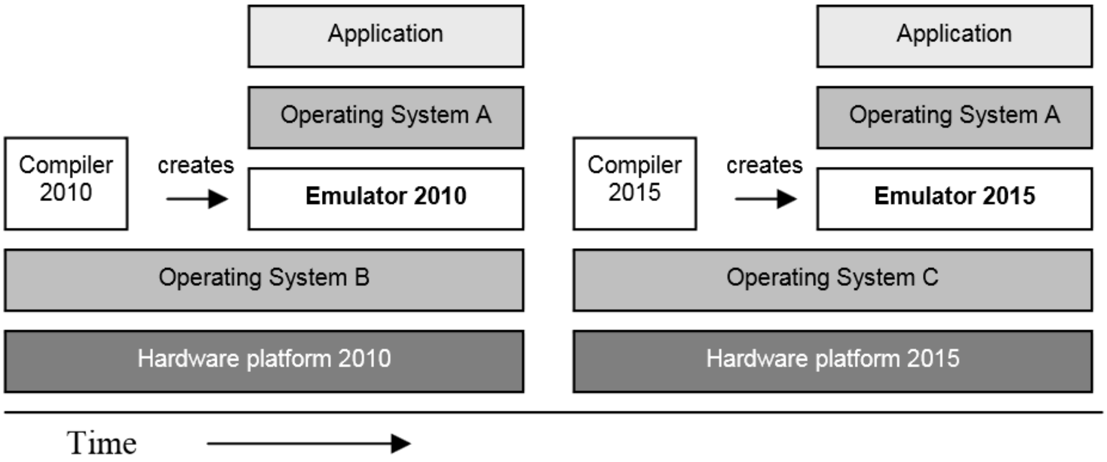

Figure 3-14: Migrated emulation over time [43]

However, there are some concerns while considering this approach such as the compiler used for source language X may not be available in the distant future. And even if it is the compiler's libraries may not be compatible with the original one, which can cause unwanted behaviour in the system.

3.4.3.3 Emulation Virtual Machine

In ideal circumstance, an emulator should run independent of time and platform. As per Jeff Rothenberg's approach to reach these independent states, we need an additional layer between the host platform and the emulator. This concept is called an Emulation Virtual Machine (EVM) [46], further researches have to be done in the emulation world to make it practical for preservation application. Combining an emulator specification and an emulation interpreter, emulators can be created to run on the EVM [43]. According to Rothenberg, the EVM is stable over time and able to run on various host platforms. The emulator is no longer bound to any particular platform. Furthermore, it is possible to run multiple emulators on the same EVM. However, the side effect is that the EVM must be maintained over time and will be quite complex. On the similar approach UVC-based preservation [47] are developing as a viable strategy for digital preservation on a technical level. A Universal Virtual Computer (UVC) is VM specially designed for preservation of digital object using emulation and migration approaches.

3.5 Software Containers

A software container is a hosting environment for software components. It facilitates the runtime support to the components it hosts in a way that is similar to an operating system hosting process [48]. It also serves as a protective barrier, monitoring the interactions between hosted components and their clients, restricting the interactions to those that are deemed unsafe.

Table 2: List of Software Containers Implementation (source: wiki [49])

Mechanism	Operating system	License	Available since	Features								
				File system isolation	Copy on Write	Disk quotas	I/O rate limiting	Memory limits	CPU quotas	Network isolation	Partition checkpointing	Root privilege isolation
chroot	most UNIX-li	varies by operating system	1982	Partial	No	No	No	No	No	No	No	No
Docker	Linux (using LXC)	Apache License 2.0	2013	Yes	Yes	Not directly	Not directly	Yes	Yes	Yes	No	No
FreeBSD Jail	FreeBSD	BSD	1998	Yes	Yes (ZFS)	Yes	No	Yes	Yes	Yes	No	Yes
HP-UX Containers (SRP)	HPUX	Proprietary	2007	Yes	No	Partial. Yes with logical volumes	Yes	Yes	Yes	Yes	Yes	?
iCore Virtual Accounts	Windows XP	Proprietary/Free ware	2008	Yes	No	Yes	No	No	No	No	No	?
Linux-VServer (security context)	Linux	GNU GPL v.2	2001	Yes	Yes	Yes	Yes	Yes	Yes	Partial	No	Partial
LXC	Linux	GNU GPL v.2	2008	Partial	Partial.	Partial. Yes with LVM or Disk quota.	Yes	Yes	Yes	Yes	No	No
OpenVZ	Linux	GNU GPL v.2	2005	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Parallels Virtuozzo Cont	Linux, Windows	Proprietary	2001	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Sandboxie	Windows	Proprietary/Shar eware	2004	Yes	Yes	No	?	?	?	?	?	?
Solaris Containers	Solaris and Open Solaris	CDDL	2005	Yes	Partial. Yes with ZFS	Yes	Partial	Yes	Yes	Yes	No	Yes
sysjail	OpenBSD, NetE	BSD	no longer supported as of 03-03-2009	Yes	No	No	No	No	No	Yes	No	?
WPARs	AIX	Proprietary	2007	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	?

Container based models provide a clear separation of concerns between application logic and enterprise services, such as transaction management, persistence, security, etc.

Table 2 shows comparison of well-known container technologies, operating systems, and their supported features.

3.6 Summary

Virtualisation and emulation are two important technique of today's IT world, and the further technology development will be focused more toward the benefit of emerging cloud computing [50]. On the other hand Virtualisation and emulation are looked on as potential enablers for preserving complex business environments for continued access regardless of technology changes over time. However, there are some challenges exists in emulation research for successfully adopting emulator solution into long term digital preservation. The main challenges are keeping the emulators portable throughout the digital curation period, KEEP [72] (Keeping Emulator Environment Portable) is an EU funded FP7 project working on developing emulation framework to reproduce the original environment in which the digital objects were created or to enable those objects to be migrated accurately to another environment. It will be useful to do some research on KEEP project to learn any useful tools or methodologies available for integrating into TIMBUS. Beside all of that, activities such as maintaining the emulator source code and migrating the compiler platform to the further technologies are also needed if we were relying on emulation for any part of our TIMBUS project. But, these technologies are outside the scope of TIMBUS project hence we would actively monitor the advancement for future integration.

4 Digital Preservation Experts Suite (T6.4/T6.6) Requirements

This chapter will recap and update the set of functional and non-functional requirements that were presented in D6.4 in month M30. These requirements are primarily driven by use cases presented in work packages WP7, WP8, and WP9 as well as the recommendation made in TIMBUS architectural design deliverables D5.5 in M18. In addition, the requirements were also revisited to align the DPES development with the latest technologies and tools. Some requirements may have already been discussed in D6.4 which are now mapped against use cases in the following tables with appropriate references. The new requirements are identified with format FRi and NFRi (FR- Functional Requirement, NFR-Non-Functional Requirement and i - sequential number). The responsibility of fulfilling the use case requirements are spread across multiple tasks within WP6, this chapter will extract all the requirements that are relevant to digital preservation expert suite (DPES) and will be incorporated in the design and implementation of T6.4 and T6.6.

The following list of use case deliverable documents are analysed and DPES specific requirements are extracted for implementations. Unfortunately not all the deliverable documents are publically available for reference due to restriction in the dissemination level. Those deliverable documents available on the public TIMBUS website are only referenced below using their URL link.

WP7 – Industrial project 1: Engineering Services and system for Digital preservation

- **Phaidra (D7.5) & RCAAP (D7.5)** – Digitally Preserving an Open Source System – case Analysis
- **Open Source Workflow – (D7.7)** – Preservation of an Open Source Workflow – Case description and Analysis

WP8 – Industrial project 2: CIVIL Engineering Infrastructure

- **LNEC (8.1)** – Use case Definition and digital Preservation Requirements [51]
- **CAD and CAM (D8.4)** – Digitally Preserving CAD/CAM Business Process

WP9 – Industrial project 3: eHealth Business Process, eScience and Mathematical Simulations

- **eScience & Math Simulation (D9.1)** – Use case Definition and Digital Preservation Requirements [52]
- **eHealth (D9.3)** - Definition of an eHealth Business Process and requirements Specification for Digital Preservation

WP6 – Implementation Requirements – Intelligent Tools and technologies to Support Digital Preservation of Business Processes

- **DPES (D6.4)** – Business Process Virtualisation and Storage Manager [3]

Intel Use Case: The IoT (Internet of things [53]) use case has been developed internally as part of a learning exercise towards the preservation and redeployment of a real business process under Task T6.7 – Integrated Test Bed. More detailed description and requirements are discussed in Section 6.1.

4.1 Functional Requirements

Table 3: Preserving Original User Interfaces

ID: FR1	**Name**: Preserving Original User Interfaces
Description	The DPES system should be able to capture the current user interface tools (GUI) and their dependent platform from original source environment and be able to restore back in the distant future without modification to provide the user with same interaction experience with redeployed system
Additional Information	User faces are the destined to provide user friendly interaction with backend system to perform certain business tasks. Over a period of time users are familiar with GUI that are in use for long time, any changes on the UI may be undesirable when redeployed in the future. Hence preservation system should maintain the format, data input and the layout of the original navigation elements.
Use cases	Phaidra (D7.5) – PFR1
Cross References:	None.

Table 4: Maintain Business Object Identity

ID: FR2	**Name**: Maintain Business Object Identity
Description	During the lifecycle of digital preservation any objects/entities used within the business process should be maintained with original identification.
Additional Information	In order to maintain the relationship and identification between business process, the preservation system should keep the original identification codes of any digital objects that exists as part of business process i.e. Database Tables of Product id
Use cases	Phaidra (D7.5) – PFR2
Cross References:	None.

Table 5: Authentic Object Rendering

ID: FR3	**Name**: Authentic Object Rendering
Description	Preservation system should be able to present the digital contents with original qualities and properties.
Additional Information	As part of long term preservation process there are many event set to happens such as platform migration, object format conversions, rendering tolls get changes, etc. However while going through these changes it is essential to maintain the originality of the digital contents for future rendering.
Use cases	Phaidra (D7.5) – PFR5, PFR8, PFR9 RCAAP (D7.5) – RNFR4 LNEC (D8.1) – FR14
Cross References:	FR10: Table 14: Digital Artefacts Format Recommendation

Table 6: Business Process Event History

ID: FR4	**Name**: Business Process Event History
Description	Capturing historical events of business process and preserving along the business environment.
Additional Information	Maintaining business process events are crucial for future investigation, trouble shooting of any process related issues or even process improvement purpose. Preservation system should be able to collect and maintain those data in the repository for future access i.e. log files
Use cases	Phaidra (D7.5) – PFR6
Cross References:	None.

Table 7: Logging Preservation Events

ID: FR5	**Name**: Logging Preservation Events
Description	System should be able maintain the event of preserved business process for future references.
Additional Information	During the preservation period the system might go through various activities such dependency library update, migration of storage system, hardware upgrade, format migration and conversion. DPES should log these events in a structured format to provide traceability in the future.
Use cases	Phaidra (D7.5) – PFR7 eHealth (9.3) – FR13, FR14
Cross References:	FR4: Table 6: Business Process Event History

Table 8: Preserving the Business Context

ID: FR6	**Name**: Preserving the Business Context
Description	System should be able import business context model developed and stored in the repository for future access
Additional Information	System must be able to preserve business context models developed during the preservation process along with the software tools used for rendering the model. These context models should be kept up to-date whenever changes taken place during the preservation lifecycle.
Use cases	RCAAP (D7.5) – RFR1, RFR2, RFR6 Open Source Workflow – (D7.7) – PFR1, PFR2, PFR2.1 LNEC (D8.1) – FR6 CAD and CAM (D8.4) – FR1, FR2 eHealth (D9.3) – FR1, FR2, FR3, FR4, FR5
Cross References:	FR4: Table 6: Business Process Event History **FR5:**

	Table 7: Logging Preservation Events

Table 9: Query and Browse Repository

ID: FR7	**Name**: Query and Browse Repository
Description	The preserved system should provide the search facilities on the preserved data.
Additional Information	System should provide interface to search and find preserved digital objects by entering key words, metadata and other search filters.
Use cases	RCAAP (D7.5) – RFR4, RFR5
Cross References:	NFR4: Table 25: Ease of Use

Table 10: Support Business Process Change Update

ID: FR14 (D6.4)	**Name**: Support Business Process Change Update
Description	The system must able to facilitate the partial update or replacement of the preserved digital representation of a business process. Updates will be triggered from context/process monitoring tools.
Additional Information	Business processes are dynamic. Their digital representations are even more so and change at least as frequently as the process they represent and at most each time a technical component within the IT system is updated, patched or reconfigured. Context agents continuously monitor the business process changes to trigger the risk evaluation. The system should be able to update the preserved business process digital artefacts for only the items which have been identified as having changed.
Use cases	LNEC (D8.1) – FR1, FR2, FR3, FR4 CAD and CAM (D8.4) – FR10
Cross References:	FR4: Table 6: Business Process Event History FR6: Table 8: Preserving the Business Context

Table 11: Redeploy for localised Analysis

ID: FR8	**Name**: Redeploy for localised Analysis
Description	The DPES should be able to redeploy any part of preserved system for testing and analysis
Additional Information	System should support modular and selective redeployment without restoring entire preserved system, but deployed system would still need to relate to its adjacent dependencies, process context and simulated data inputs.
Use cases	LNEC (D8.1) – FR7, FR8 CAD and CAM (D8.4) – FR5 eHealth (D9.3) – FR8
Cross References:	None.

Table 12: Detect Information Redundancy

ID: FR12 (D6.4)	**Name**: Detect Information Redundancy
Description	The system must be able to detect duplicates when the capturing information is already preserved.
Additional Information	Since a lot of BPs are duplicating data (the same operating system, patches, and even applications may be installed many times), the DP system must be able: • to detect such duplication and prevent multiple preservation of the same data; to handle different data types and work with structured and non- structured information
Use cases	LNEC (D8.1) – FR11
Cross References:	None.

Table 13: Digital Artefacts Format Database

ID: FR9	**Name**: Digital Artefacts Format Database
Description	The system must maintain a database table containing the entire data format of currently preserved digital artefacts in TIMBUS repository.
Additional Information	During digital curation period some data/file format may be obsoleted and no future comparability would be available. In such a case it is important to have databases of digital artefact format to search and identify the list of affected formats and their business processes for migration or conversion activities.
Use cases	LNEC (D8.1) – FR13 CAD and CAM (D8.4) – FR8, FR9
Cross References:	FR2: Table 4: Maintain Business Object Identity FR3: Table 5: Authentic Object Rendering

Table 14: Digital Artefacts Format Recommendation

ID: FR10	**Name**: Digital Artefacts Format Recommendation
Description	Recommend right format for long term preservation.
Additional Information	Format recommendations are essential during ingestion process in order to identify correct alternate format to eliminate risk of incompatibility issue while rendering content in the future. Preservation Identifier is being integrated with DPES to obtain recommendation on best file format long term preservation.
Use cases	LNEC (D8.1) – FR13 CAD and CAM (D8.4) – FR8 eHealth (9.3) – FR12
Cross References:	FR2: Table 4: Maintain Business Object Identity FR3: Table 5: Authentic Object Rendering

Table 15: Preserve Business Process

ID: FR11	**Name**: Preserve Business Process
Description	The system should preserve digital artefacts that support the business process.
Additional Information	Preservation system should precisely analyse context model, business process description and extract only information needed to preserve the business environment including dependencies rather than preserving the entire IT system.
Use cases	eHealth (D9.3) – FR6
Cross References:	**FR6:** Table 8: Preserving the Business **Context**

Table 16: Redeploy Business Process

ID: FR12	**Name**: Redeploy Business Process
Description	System must perform redeployment of a preserved system with required IT resources on existing IT environment.
Additional Information	Preservation should able to sequence the redeployment process with following steps. • Identify the Business process to redeploy • Identify the hardware infrastructure • Identify the software components • Retrieve preserved artefacts and perform integrity check • Choose redeployment methods i.e. Physical IT system, Virtual Machine, Container or emulations • Run redeployment script to copy and install tools and libraries • Install business process services and perform test.
Use cases	CAD and CAM (D8.4) – FR1, FR2 eHealth (D9.3) – FR9 Open Source Workflow – (D7.7) – PFR3

Cross References:	FR8: Table 11: Redeploy for localised Analysis

Table 17: Preserve External Functionalities

ID: DF14	**Name**: Preserve External Functionalities
Description	The preservation system should able to capture the interaction between external systems and services.
Additional Information	Some business process dependent on external third party system or services to perform its core tasks. Several interactions are happen during the operation of business process through various communication methods such as calling web services, importing libraries or consuming documents. Preservation system should be able to capture those interactions and sample data exchange for future verification purpose.
Use cases	Open Source Workflow – (D7.7) – PFR4, PFR4.1, PFR4.2,PFR4.3,PFR4.4
Cross References:	FR6: Table 8: Preserving the Business Context

4.2 Non-Functional Requirements

Table 18: Maintain the Integrity of the Digital Objects

ID: FR10 (D6.4)	**Name**: Maintain the Integrity of the Digital Objects
Description	The system should not alter the package or lose the data bits from the artefacts within packages during the lifecycle of preservation planning, maintenance and redeployment.
Additional Information	The system must be able: • to check and verify the integrity of sourced artefacts by comparing source provided fixity (i.e. checksum MD5, SHA1, etc.) data and known (calculated) fixity data. • to maintain the integrity of artefacts during maintenance such as migrating the repository, upgrades, business process change update (SW, HW, OS, or architecture changes) etc.
Use cases	Applicable to all uses cases.

Cross References:	FR3, FR2, DPES (D6.4) – FR16 FR3: Table 5: Authentic Object Rendering FR6: Table 8: Preserving the Business Context FR12 (D6.4): Table 12: Detect Information Redundancy

Table 19: Preservation and Redeploy Access Control

ID: FR13 (D6.4)	**Name**: Preservation and Redeploy Access Control
Description	The system must facilitate user access management during the entire preservation lifecycle and protect the repository from unauthorised access to the preserved business process digital artefacts.
Additional Information	The system should facilitate creating user accounts with key credential (name, id, email, and password) before requesting access to TIMBUS repositories and the preservation system (i.e. the DPES). The system administrator will perform audits on the registered user account and grant permission. The system should maintain the user account and relate business process ownership during preservation process. The system should also provide specific restrictions, where the user should be authorised to perform full or partial preservation or redeployment of business processes. • User accounts can be set to require periodic re-validation, re-authorisation for continued access in the event of employee role changes or employees leaving the organisation.
Use cases	Phaidra (D7.5) – PNF2 RCAAP (D7.5) - RNFR3 LNEC (D8.1) – FR12, NF3 CAD and CAM (D8.4) – NFR3, FR7
Cross References:	None.

Table 20: Logical Preservation of Digital Artefacts

ID: NFR1	**Name**: Logical Preservation of Digital Artefacts
Description	The system should be able collect, organise and maintain enough metadata to represent the digital object over long time.
Additional Information	Logical preservation involves preserving the intellectual content of the data in the event of future technological and knowledge changes [54]. It enables future interpretation of the preserved data by intended user that they may use technologies unknown today. i.e. attempting to view work document after several year without knowing the requirement for MS Office software.
Use cases	Phaidra (D7.5) – PNF4 Open Source Workflow – (D7.7) – PFR2.1 LNEC (D8.1) – NF7 CAD and CAM (D8.4) – NFR7 eHealth (9.3) – NF6
Cross References:	DPES (D6.4) - FR6 FR6: Table 8: Preserving the Business Context

Table 21: Secure and Reliable Storage

ID: NFR3	**Name**: Secure and Reliable Storage
Description	Storage system used for preserving digital object should be reliable, should have continued access for long period as required by business process.
Additional Information	Storage system should be protected from any natural disaster, malicious attacks, bit loss or file corruption during migration or hardware upgrade. Digital object should be preserved in distributed storage system across multiple geographical regions with appropriate raid features (Mirror) i.e. AWS S3 distributed cloud storage service for multiple backups.
Use cases	Phaidra (D7.5) – PNF4 LNEC (8.1) – FR10, NF4 CAD and CAM (D8.4) – NFR4

	eHealth (9.3) – FR11, NF3
Cross References:	DPES (D6.4) - FR6, FR15, FR11 FR4: Table 6: Business Process Event History

Table 22: Self-Contained and functional Completeness

ID: NFR2	**Name**: Self-Contained and functional Completeness
Description	Preserved business process should be self-contained to execute in future.
Additional Information	Preserved system should retain all the dependency i.e. libraries, tools and services in a self-contained way without needing any additional resources to deploy the system in the future.
Use cases	RCAAP(D7.5) – RNFR3, RNFR5 and all other use cases.
Cross References:	FR3: Table 5: Authentic Object Rendering FR6: Table 8: Preserving the Business Context DF14: Table 17: Preserve External Functionalities

Table 23: Scalability of Preservation System

ID: TR18 (D6.4)	**Name**: Scalability of Preservation System
Description	The system must be able to handle and support the increased workload without adding additional resources or altering the designed architecture.
Additional Information	Modern business environments are heavily dependent on IT infrastructure. The complexity and size of IT landscapes are continuously increasing as the business grows, and also due to the nature of business processes. The DPES system should be flexible enough to scale in such a way as to be able to digitally capture and preserve any size of IT landscape for a business process. The architecture of DPES should hold true of the definition by Charles B Weinstock “Scalability is the ability to handle increased workload by repeatedly applying a cost-effective strategy for extending a system’s capacity” [55].
Use cases	Applicable to all uses cases.

Cross References:	None.

Table 24: Preserve the security functionality

ID: NFR3	**Name**: Preserve the security functionality
Description	The security protocol exist on original system must be preserved.
Additional Information	Preservation system should capture and preserve the business process security such as user access control, roles and privileges to avoid unintended damage to the redeployment system.
Use cases	Open Source Workflow (D7.7) – PNF2 CAD and CAM (D8.4) – FR7
Cross References:	FR13 (D6.4): Table 19: Preservation and Redeploy Access Control

Table 25: Ease of Use

ID: NFR4	**Name**: Ease of Use
Description	The preservation system should be use friendly
Additional Information	System must present clean flow of preservation and redeployment tasks and easy to understand by users.
Use cases	Applicable to all uses cases.
Cross References:	FR1: Table 3: Preserving Original User Interfaces

Table 26: Non-intrusive and Transparent Capture

ID: NFR5	**Name**: Non-intrusive and Transparent Capture
Description	The preservation system should be use able to capture the system information in a non-intrusive and transparent way

Additional Information	Preservation system should not cause any disruption to running system while collecting metadata or digital artefacts of any business process.
Use cases	eHealth (9.3) – NF1 LNEC (8.1) – NF1
Cross References:	**D6.4 – FR5** : Table 6: Source Business Process Digital Artefacts

5 Digital Preservation Expert Suit - Architecture

This chapter recaps and updates the architecture and design of the Digital Preservation Expert Suite which was submitted in M30 in deliverable document D6.4. The scope of this topics lies under combined tasks of T6.4 and T6.6. Section 6.1 describes the application architecture driven by the functional requirements gathered in Section 4.1 and Section 5.2 discusses the technical architecture of the DPES system which is again driven by non-functional requirements extracted in Section 4.2.

5.1 Application Architecture of DPES

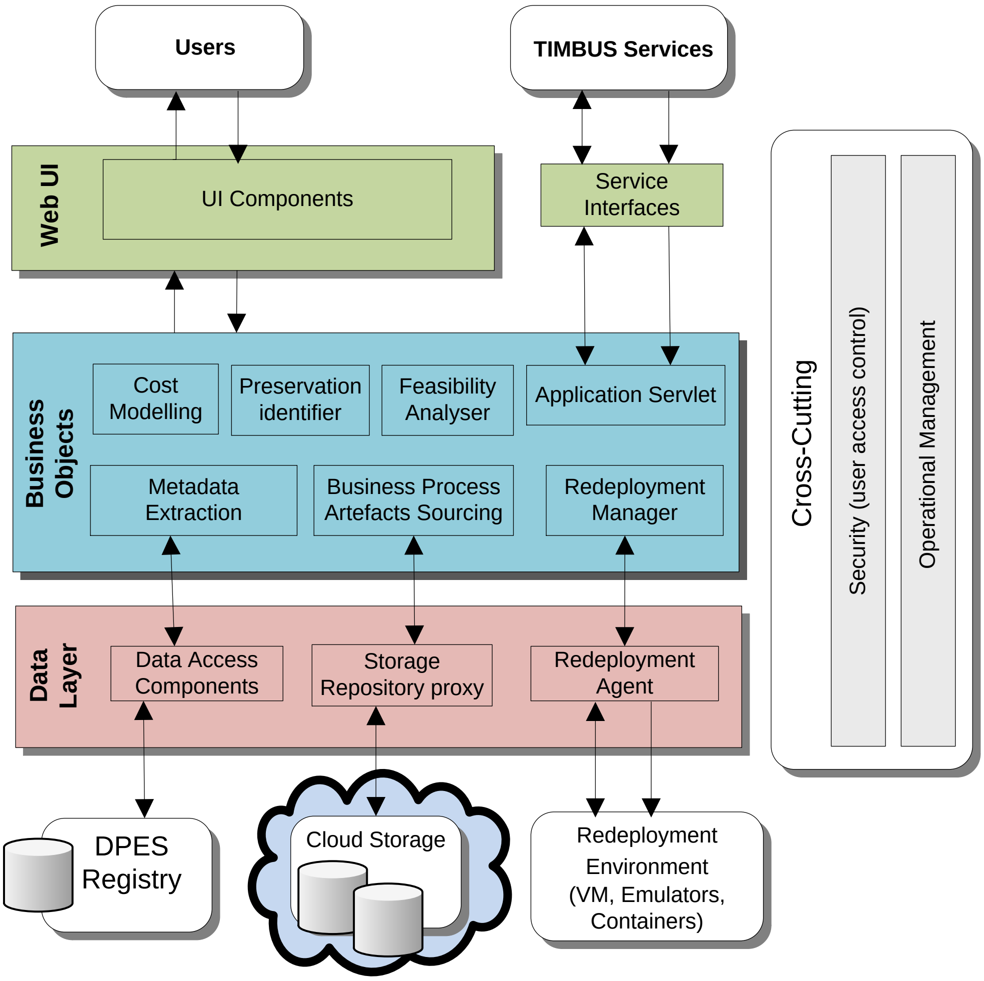

Figure 5-1: Application Architecture of DPES

Figure 5-1 illustrates the application architecture of DPES showing the internal components, the interaction with external TIMBUS services developed in other WP6 tasks and third party services such as cloud storage services provided by Amazon S3.

The application architecture consists of three main logical layers and a cross cutting vertical layer spans those and which implements access control and operational management of preservation system.

5.1.1 Web UI and Service Interfaces

This layer consists of a web application for the DPES UI and a service API for integrating the DPES with other TIMBUS tools such as iERM, the Extractors and the PI tools.

5.1.1.1 Web UI

Web UI has been developed using Flash and HTML UI framework to provide user friendly navigation support for the DPES user. DPES application can be accessed via this console for performing preservation and redeployment actives.

Fulfilled Requirement: NFR4: Table 25: Ease of Use

5.1.1.2 Services Interfaces

Within WP6, tools and services are designed and implemented in a distributed way for easy management of service development and deployment. Each service is hosted by the tool owner who developed the services as part TIMBUS WP6 implementation. In order to integrate these services with DPES, a set of API's are provided between the services. Example: see Section 7.1 for iERM and DPES integration through Restful API.

5.1.2 Business Object management

This layer consists of several key components for achieving preservation of a business process, namely the preservation feasibility analyser, cost modelling, metadata extractor, redeployment manager and business process artefacts sourcing. The remaining text in this section briefly describes the application of these components.

5.1.2.1 Preservation Feasibility Analyser

This module imports the business process resources in the form of an OWL file (Context Model) from the Preservation Identifier (PI), and then extracts the list of resources with alternative options as identified by the PI tool. Having extracted the list of resources, the user can review the list and interpret the feasibility of digitally preserving the resources. Initially, the system would heavily rely on that expert input for making a decision, however over a period time, an expert knowledge base could be built within DPES registry to aid the DPES user in making decisions automatically using combinations of PI recommendations and historical information.

The criteria for accessing the feasibility of preservation are already discussed in detail in Section 5.1 of D6.4 deliverable document submitted in month M30. However, for the convenience of reader the following paragraph revises those criteria in brief.

- *Technological* – the system should analyse each resource and identify the available solutions to digitally preserve them. In some instances, there may not be enough solutions available to digitally capture the resources which may affect the feasibility of preservation of that business process (BP).
- Business – The current business solutions heavily rely on the modern IT infrastructure. Whenever those change, as they are bound to over time, it leads to a re-evaluation of the business models and underlining BPs. Monitoring of the business model allows triggering the re-assessment and potential preservation or redeployment of the critical BPs if changes are detected.
- Operational – Changes within the business model have an impact on the company or organisational operations. To keep track of these changes is important for discovery relationships between BPs.

Fulfilled requirements: **FR1:** Digital Preservation Feasibility Analysis**,** defined in D6.4.

5.1.2.2 Cost Modelling

This module is used for estimating the cost of digitally preserving a selected business process described in the Context Model (T6.5) and the PI (Preservation Identifier) provided alternate resources. The cost model used is simplified abstraction from the LIFE3 [56] Long Term Preservation Cost model and recommendations from TIMBUS deliverable Section 6 of D4.8: DP & Intelligent Enterprise Risk management. The DPES adopts the abstracted cost model from LIFE3 and the model is implemented with a set of rules to determine the cost elements for each phase of the preservation process. These rules can be altered by the DPES administrator to reflect the costs of technologies such as hardware, software, licensing and other operational overheads. For simple illustration and ease of maintenance, the cost components are allocated to different measurable of a business process resource, for example, to the quantity of digital artefacts, size, license cost, conversion, etc. An example costing rule is described using Table 22 of deliverable document D6.4 [3] submitted in month 30.

This module has fulfilled requirement: **FR3:** Estimate Preservation Cost**,** defined in D6.4

5.1.2.3 Metadata Extractor

The metadata extractor included within DPES envisioned generating metadata of an individual digital artefact for long term digital preservation whereas the Extractor tools developed in Task T6.5 are aimed to extract meta information and dependencies details from the targeted business environment. Therefore it is useful to think of the metadata in TIMBUS existing at two different levels. The DPES metadata is generated at the file, or artefact, level for each object stored in the preservation storage repository.

There are several metadata extraction tools available to integrate with the DPES metadata extractor, e.g. Apache Tika [57], Fits [58], and JHOVE2 [59]. The following list of metadata elements are needed for long term digital curation.

- **Technical Metadata**: Most of the preservation metadata is "technical metadata," or metadata which describes the technical properties of digital files and bit streams. Some of these properties, such as size, format, and fixity information, are applicable to most materials and are included in PREMIS and other general preservation metadata specifications.

- ***Format identifications:*** example: MIME type of digital objects, and file name extension.
- **Environment to use:** Specifications of hardware, software and supporting libraries needed to render or use a digital object during redeployment.
- **Fixity**: Fixity information is essential for determining whether a file has been changed between two points in time (i.e. MD5, SHA1, etc).

In the DPES system the extracted metadata (at artefacts level) will be populated and kept synchronised with local DPES registry and context model annotation using the PREMIS metadata representation schema. An example of the PREMIS ontology is shown on the Figure 5-2 to illustrate the structure of metadata population at the XML document level.


```
– <files>
  – <file name="popeye_taxi-turvey_512kb.mp4" source="derivative">
      <format>512Kb MPEG4</format>
      <original>popeye_taxi-turvey.mpeg</original>
      <md5>2856b54fdfc72d211a9f9e5605c3c11b</md5>
      <mtime>1227210907</mtime>
      <size>26696485</size>
      <crc32>bba776e6</crc32>
      <sha1>b7183aa354a026c652e2409b1da27d4683528907</sha1>
    </file>
  – <file name="popeye_taxi-turvey.mpeg" source="original">
      <format>MPEG2</format>
      <md5>8f3f6378140db2eafc9291eac2cb0219</md5>
      <mtime>1110496353</mtime>
      <size>144445440</size>
      <crc32>46b0cbad</crc32>
      <sha1>e03c9ad81ff801e07c7c3964f395d3acb698fa5f</sha1>
    </file>
  – <file name="popeye_taxi-turvey_reviews.xml" source="metadata">
      <mtime>1288415696</mtime>
      <size>1670</size>
      <md5>f7295a5122da83ca59b2120d9a7208ed</md5>
      <crc32>f485688d</crc32>
      <sha1>f86455ed4ddcf2061eeec8fe7ead3f2dc1c71bb3</sha1>
      <format>Metadata</format>
    </file>
  – <file name="popeye_taxi-turvey.gif" source="derivative">
```


Figure 5-2: Sample File Metadata population [60]

This module has fulfilled requirement: **FR4** defined in D6.4 Virtualisation Manager

5.1.2.4 Preservation Identifier

The Preservation Alternative Identification (PI) module is capable of discovering potential alternative implementations for a given process, specifically regarding some of those components identified being at risk. For each of these risks that contain aspects of a process context, a number of potential preservation alternatives are identified. These alternatives can then be assessed for their feasibility in the subsequent steps of the TIMBUS preservation process, partly by tool support, and partly based on expert knowledge.

The PI was already described in Deliverable D6.4, here the current state of the implementation of the module, which will be completed in deliverable D6.9 in M45 is given.

Functionality Description

The preservation alternative identifier expects a context model instance of the business process to be analysed as input. Further, a list of risks identified in association with the business process needs to be provided. As an output, the PI provides a list of preservation alternatives, each of which contains of a modified context model instance, and a list of the changes needed to arrive at this new context model instance from the originally provided one. Due to the complexity involved, risks are treated independently from each other. For each risk that affects certain type of resources, a number of strategies to mitigate that risk can be proposed.

Each preservation alternative consists of one preservation action, covering data objects (documents and others) and software or systems involved in the process. For more details, please refer to Deliverable D6.4.

- For each data object (*Artefacts* concept in the Context Model) that is either produced or consumed in the process and for which the data format is at risk of becoming obsolete, an alternative for ensuring long-term access to this data object has to be produced.
- For each software component involved in the process (*SystemSoftware* or *Artefact* concepts in the Context Model), the alternatives proposed could consider the following options for preserving the software's execution environment (adapted from the Software Sustainability Institute, http://www.software.ac.uk/resources/approaches-software-sustainability)
    - *Replacement* of the software by an alternative providing the same functionality.
    - *Emulation* of the execution environment, i.e. utilising emulators that mimic the functionality and behaviour of the hardware and software environment.
- For external services (*InfrastructureService* or *ApplicationService* concepts in the Context Model, where there is no access to the implementation (the source code or even the binaries), alternatives can include
    - A mock-up service that can replay messages based on a data base captured during the existence of the service.
    - Migration to an alternative service that provides the same functionality, which could be still a (more reliable) external service, or in-house, e.g. moving cloud storage to a local storage.

5.1.2.5 Business Process Artefacts Sourcing

This module was previously named as Business Process Resource Manager in deliverable document D6.4, the primary objective of this module is to acquire the business process resources defined in the context model and also recommended by PI module in the form of digital artefacts. This module imports context model to identify the business processes and the list of digital artefacts to be sourced for preservation. Acquisitions of digital artefact are achieved from four main sources as shown in the Figure 5-3.

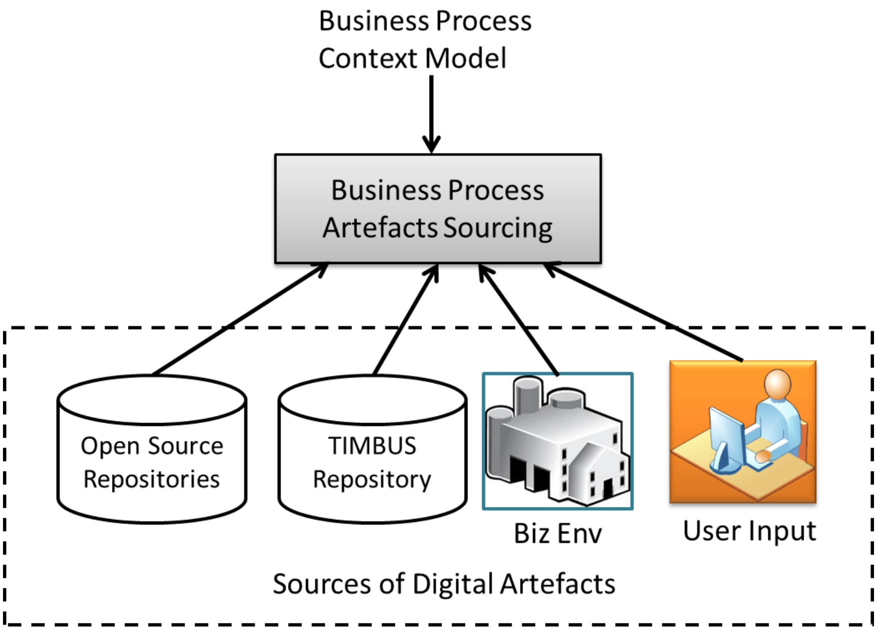

Figure 5-3: Sources of Business Artefacts

Sourced artefacts are organised in three categories as described below based on preservation requirement decision made by DPES user:

Resource Type 1: Metadata and Cloning of Physical Machine to Virtual disk

For this resource type, the Artefact sourcing module will generate a virtual machine (VM) image of all the computing systems from the live business environment. In order to create the cloned VM images from business computing systems, the following requirements need to be met:

- Exclusive access to physical machine and hard disk, in offline mode if needed. Some tools for this support online access which is highly preferable.
- Consent from users - preserve any personal data, email, and user login information left in the system during cloning process - privacy policy agreement.
- Ability to deploy and run scripts or third-party tools in the business computer to create cloned VM disks.

Resource Type 2: Metadata and VM Images of Newly Constructed Business Machine

In this resource type, the BPRM will facilitate the building of new virtual machines using the digital artefacts gathered in resource type 2. This newly built virtual machine(s) will replicate the live business environment. The requirements for resource type 3 are as follows:

- Hardware configuration of each machine
- OS type, version, build info
- List of applications and their dependencies, versions, vendors, license
- User account details
- Any business process data to prepopulate the system.

Resource type 2 is always needed for long term preservation of a selected business environment and for its future redeployment. Additionally, resource types 1 or 3 are needed as a back-up to resolve any unforeseen complexity during redeployment, preferably type 1. But, choosing type 3 over type 1 as a preservation/re-deployment strategy depends on following constraints:

- Live business machine disks are too large to clone
- Unable to access running/live business environment
- Business interruption during cloning process is unacceptable (expensive)
- Too much personal and unwanted information in the business IT system which requires complex privacy policy agreement and disk clean up.

Resource Type 3: Metadata and Extracted files (.dll, .rpm, .deb, .iso, .msi, .config, and etc.)

The redeployment of the business process will be based on the resources gathered in Type 2 as other two resource types (type 1 and 3) are just for back-up purpose to provide alternatives to the future designated community should the primary re-deployment mechanism prove unsuccessful. The module depends on the Windows and Linux metadata extractor tools and the business process owner to get the following information and resources:

- Package of extracted files and metadata from the extractor tools (Task T6.5)
- Software license keys sourced from business owner or purchased as part of the preservation process.
- Scripts to copy the extracted file back to the default location during deployment (i.e. the reverse of extraction process) provided by Task T6.5.
- Hardware and Software configuration of each Business machine should be present in the Context Model.

Packaging of Artefacts:

The packaging activity involves annotating the context model with details of sourced artefacts as per the tree structure presented below in Figure 5-4. Context Model developed in WP4 will provide placeholder within DSO for populating the preserved package details as well as extracted metadata. This also creates storage containers in the repository following the same grouping for simplified manageability.

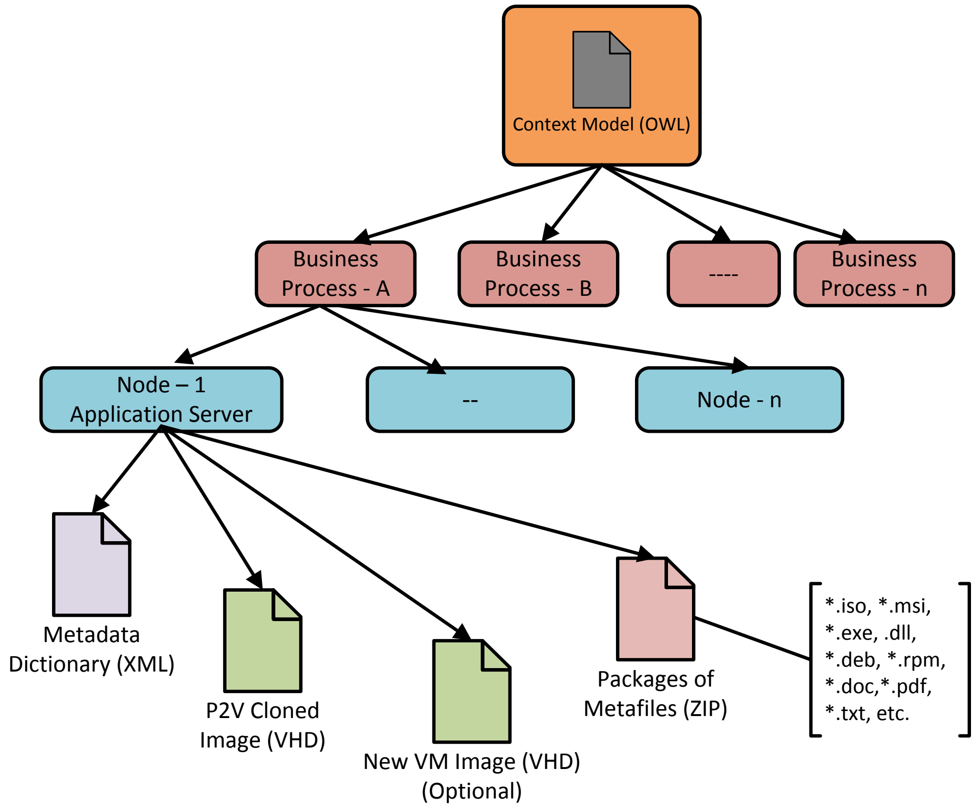

Figure 5-4: Packaging Structure of Business Artefacts

This module has Fulfilled Requirement:

NFR5 – see Table 26: Non-intrusive and Transparent Capture

FR14 (D6.4): Table 10: Support Business Process Change Update

FR11: Table 15: Preserve Business Process

FR10 (D6.4): Table 18: Maintain the Integrity of the Digital Objects

NFR2: Table 22: Self-Contained and functional Completeness

5.1.2.6 Application Servlet

The application servlet implements the Service interaction APIs, which act as gateway to external services to respond to requests by connecting to other module within business object management layer.

5.1.2.7 Redeployment Manager

This module serves as a gateway between the preservation environment and the redeployment suite (TIMBUS Task T6.6). It invokes the re-deployment server and instantiate the container for the re-deployed business process, copy packages from the repository to the VM builder, etc. The redeployment manager presents the DPES user with a planned redeployment strategy. Essentially these are pre-prepared scripts to enable making changes during deployment, or for future redeployment purposes. The redeployment manager invokes the redeployment process via a proxy which connects to the redeployment environment i.e. the T6.7 Integrated Test Bed in the case of the TIMBUS project.

This module Fulfilled Requirements:

D6.4 Virtualisation Manager - **FR13, FR14, FR15**.

FR12: Table 16: Redeploy Business Process

FR8: Table 11: Redeploy for localised Analysis

5.1.3 Data Layer

The proxy service module provides interfaces to the database server, the repository services and the virtualised/emulated redeployment environment. The following section describes each module in the data layer in further details.

5.1.3.1 Data Access Components

This component provides a connection to the local DPES registry database server to maintain the status of preservation requests, store key metadata concerning the preservation activities for search, locate and to continue the preservation process workflow. The DPES registry also stores the user account information for secure login and access to the preserved business process. Database schemas for this were presented in the previous deliverable D6.4 Virtualisation Manager in month M30.

This module has contributed fulfilling following requirements:

FR9: Table 13: Digital Artefacts Format Database

FR4: Table 6: Business Process Event History

FR2: Table 4: Maintain Business Object Identity

FR13 (D6.4): Table 19: Preservation and Redeploy Access Control.

5.1.3.2 Repository Proxy

This module acts as a gateway between DPES and the various storage repository services to submit and reclaim packages during the lifecycle of the preservation process. The repository proxy enables switching between various storage providers without altering the DPES system. A RESTful HTTP API is used to interact with the object storage service which provides cross protocol compatibility as shown in Section 3.1.1 of State of the Art study. The RESTful API exposes the object storage and allows the creation of multiple containers in storage structures and the creation of storage packages with metadata attachments. The object storage services allows the DPES system to upload larger files in multipart, this approach is convenient when uploading very large number of digital artefacts of file sizes in Gigabytes. Multipart enables parallel upload, pause, and stop and restarts the uploading activity at any time during the preservation process.

This module has fulfilled the following requirements:

FR8 (D6.4): Table 9: Create Storage Container in Repository

FR9 (D6.4): Table 10: Support Incremental uploading to remote repository

NFR3: Table 21: Secure and Reliable Storage

TR18 (D6.4): Table 23: Scalability of Preservation System

FR10 (D6.4): Table 18: Maintain the Integrity of the Digital Objects

FR12 (D6.4): Table 12: Detect Information Redundancy

5.1.3.3 Redeployment agent

This module serves as a gateway between the DPES redeployment manager and the redeployment suite (TIMBUS Task T6.6). It invokes the re-deployment server and instantiates the container for the re-deployed business process, copies packages from the repository to the VM builder, etc.

This module has fulfilled the following requirements:

FR13 (D6.4): Table 19: Preservation and Redeploy Access Control

FR12: Table 16: Redeploy Business Process

5.2 Technical Architecture of DPES system

This section of the deliverable will review the technical architecture presented in deliverable D6.4 M30. It also updates the technology and services integration into the DPES system. The technical architectures are primarily driven by the non-functional requirements stated in Section 4.2. The diagram shown Figure 5-5 has been modified with some changes to the storage repository and redeployment technologies. An Amazon S3 Storage service is added in addition to OpenStack to provide flexibility across the storage platform and to take advantage of an available and stable client side java development library [61] SDK.

On the redeployment side, two new technologies are being explored to meet the self-contained modular redeployment needs. OpenVZ [62] and Docker [63] linux containers are added in addition to Proxmox [64] virtualisation server and QEMU emulator presented in the previous deliverables D6.4 in M30. These two new containers are currently being evaluated to fulfil our requirements. The following content briefly explains the capabilities of these technologies with respect to fulfilment of the non-functional requirements stated in Section 4.2.

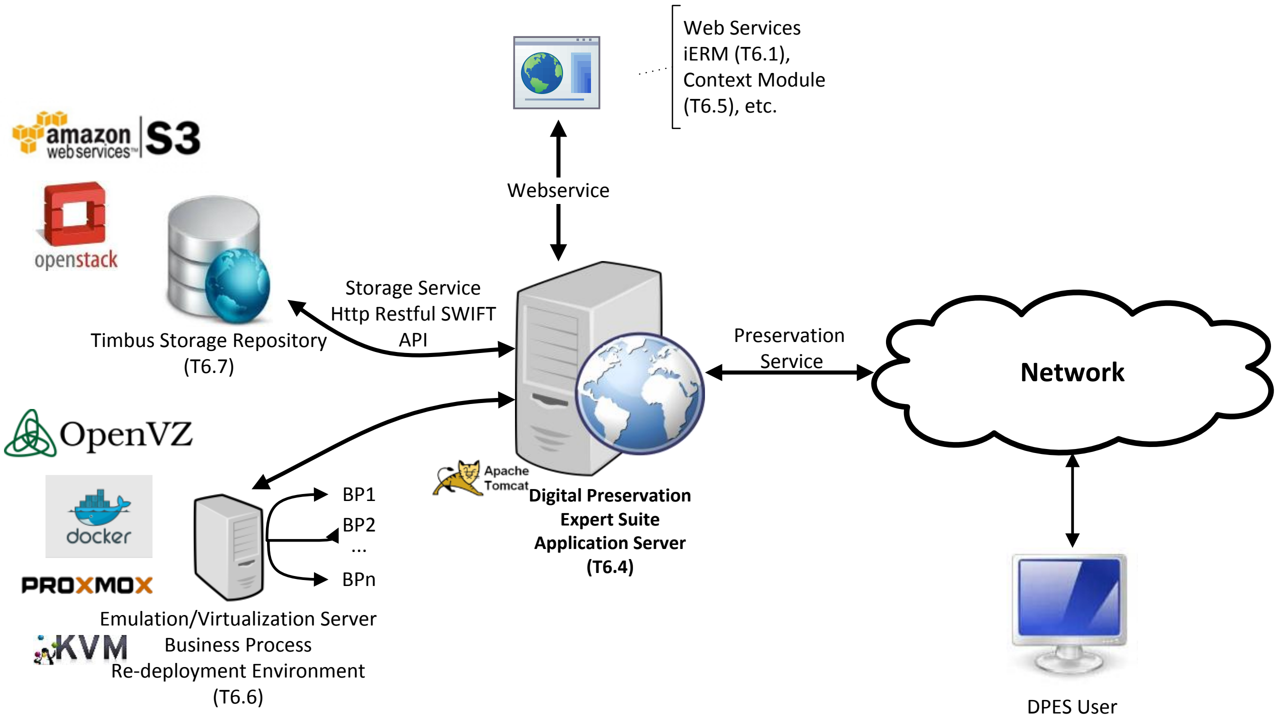

Figure 5-5: DPES High level Technical Architecture

5.2.1 DPES Web Application Server

The DPES web application server provides a run-time container for deploying and running the preservation process via a web client. The DPES functional modules are developed using the JAVA programming language and are deployable inside the Apache Tomcat web servlet container. Figure 5-6 shows the updated diagram of the DPES web application deployment where a Service API layer is added to enable interaction between DPES and other TIMBUS services.

Figure 5-6 illustrates the deployment of the DPES application on a Linux machine running the Ubuntu Server 12.04 operating system. Since Apache Tomcat is a java implementation and is cross platform compatible, it uses JVM (java virtual machine) as a run time/execution environment.

This module has fulfilled the following requirements:

TR18 (D6.4): Table 19: Scalability of DPES System

TR19 (D6.4): Table 20: Flexibility and Interoperability

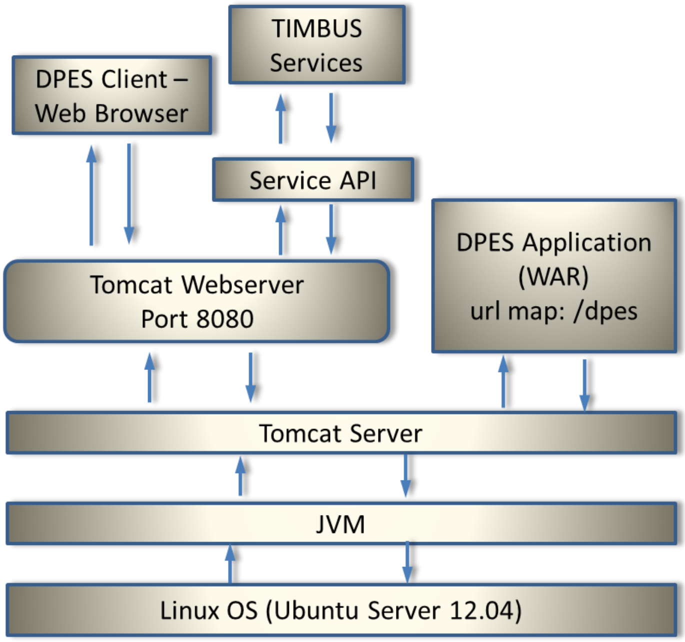

Figure 5-6: DPES Web Service Deployment

5.2.2 TIMBUS Storage Repository

The DPES needs to provide a storage solution. The two classes most readably available are object storage and block storage. As detailed in Section 3 of this deliverable, object storage is becoming very popular as an archival solution because it allows the use of commodity storage systems while still supporting data redundancy and fixity checking. Object storage is also vastly more scalable than traditional file systems because of its simplified usage model. While the TIMBUS architecture supports any back-end storage medium, object storage systems are used in the current prototype to provide a long-term storage solution. TIMBUS has developed and tested its initial storage solution using OpenStack Swift [65]. However, OpenStack needs extensive initial setup and in-house hardware capability to manage clustering / replication of storage nodes. As a more expedient solution we also decided to investigate Amazon S3 and Amazon Glacier cloud storage as a TIMBUS storage solution.

The exploitation deliverables, D2.2 and D2.3 have pointed out the many shortcomings of cloud offerings in terms of providing rounded, robust long-term preservation solutions. However, those shortcomings are due to the *service provider* not offering them. This does not prevent a 3rd party from implementing those services itself and back-ending their storage onto a cloud providers service. This does not solve all the issues of using cloud providers for long term archival purposes, but as an expedient for TIMBUS, it is sufficient to accelerate our progress.

The Amazon S3 API and Java SDK are more mature and well documented than OpenStack, but are also compatible with OpenStack's swift implementation. Object Stores provide HTTP interfaces which allow for fast, easy access to files by users from anywhere in the world using a unique URL for each object stored in the repository. In the DPES, TIMBUS is developing the storage proxy as shown in the DPES application architecture (Figure 5-7) which implements the HTTP API provided by storage services to perform basic operations such as store, retrieve, copy, and delete files, as well as control which users can do which tasks.

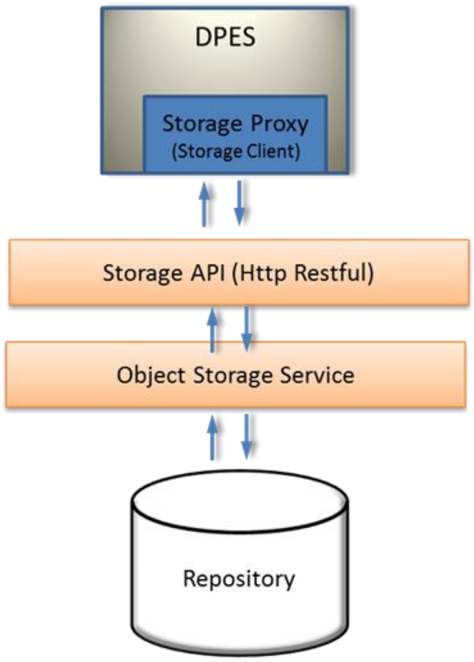

Figure 5-7: DPES Repository Communication

This module has fulfilled the following requirements:

TR19 (D6.4): Table 20: Flexibility and Interoperability

NFR3: Table 21: Secure and Reliable Storage

5.2.3 Redeployment Server

The Digital Preservation Expert Suite (DPES) will be interfaced with an emulated redeployment environment described in T6.6 for re-deploying the preserved business processes. The redeployment server is based on Proxmox Virtual Environment [64], an open source virtualisation solution. Proxmox VE is based on KVM virtualisation and container-based virtualisation and manages virtual machines, storage, virtualised networks, and HA (High Availability) Clustering. Proxmox CPU emulation is performed by QEMU and supports emulation of almost all the modern x86 processors for business process computing needs during redeployment. Figure 5-8 shows the Proxmox KVM virtualisation server display hardware resources, container and virtual instance creation dialog menu for deploying business process services. Further details of redeploying business process using Proxmox VE are discussed in Section 7.5.2.

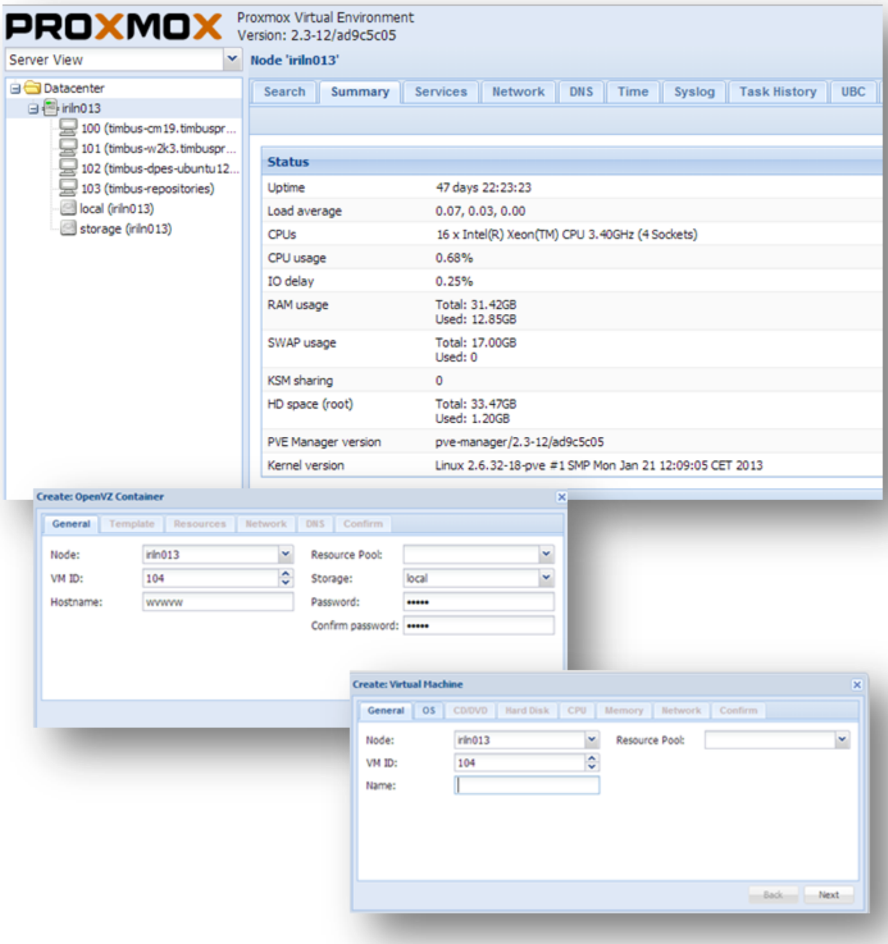

Figure 5-8: Proxmox Web GUI

6 Use Cases – Preservation Perspective

This section discusses the use case scenarios presented by the TIMBUS consortium partners in terms of the preservation and redeployment strategy using tools and services being developed within WP6. Each use case has its own deliverable document except for the Intel IoT scenario which is introduced recently. Most of the content discussed in this section is extracted from use cases to gather information for DPES implementation and developing preservation strategies.

6.1 IoT – Internet of Things (T6.7)

KaZSM (Kafka [66], ZooKeeper [67], Storm [68], MQTT [69]) forms part of an Intel Quark [70] based sensor data acquisition and analysis for an end-to-end IoT platform code named 'Palmerfield'. This is also discussed in more detail as part of Intel's exploitation plan in deliverable D2.3. To quickly summarise that, this work is part of a planned product and TIMBUS has the possibility of including concepts and features in that product. The product is acting as a test platform for our TIMBUS tools with a view to integrating long-term data preservation concepts into the product when it launches. Further details of this use case scenario are presented in the Appendix 10.2.

6.1.1 Digital Preservation Requirements

An archimate representation of the KaZSM ontology has been designed but cannot be shared in this deliverable because it is confidential and this document is public. From a TIMBUS perspective, KaZSM constitutes an end-to-end IoT business process which we want to preserve because the data contained within it potentially poses a risk to customers if it is not available and accessible in the long term. It also acts as a very good challenge for the TIMBUS tools as it contains a mix of sensors, M2M (Machine to Machine devices - Intel™ Galileo™'s), cloud services, cloud storage, analytics functions and end user API's and web based UI's. To preserve the end-to-end system, an ERM analysis of the KaZSM business process identified following risks:

6.1.1.1 Hardware node failure

KaZSM is a highly distributed system meaning it is running on many hardware nodes in order to be scalable. Due to this, the risk of a hardware failure is higher than that for systems running in smaller clusters. Impact due to a hardware failure will be low since it is a distributed system and redundancy is built-in but there will impact to the performance temporarily. The risk of a hardware failure by itself does not justify digital preservation due to low impact.

6.1.1.2 Data Loss

Database failure or failure of storage may lead to data loss. The main function of KaZSM is to acquire sensor data, aggregate it and provide it to the consumers of the data. Consumers of the data break in to many categories. Some are interested in making real time operational decisions in, for example, crisis response in a city. In essence, those consumers are primarily interested in the current data and not historical data. So loss of historical sensor data will not be of much impact to their business. There are however other users of the data which may also include persisted time scale based aggregation of the original data and for these, loss of historical data will cause an impact. For example, this group would include city planners who want to understand traffic flows or environmental impacts from infrastructure development in the city. This risk gives rise to using preservation to help mitigate data loss because the systems used to analyse the data depend on the current components being in place to enable the analysis of data. However, these analysis tools are not directly part of KaZSM, they are part of 'Palmerfield' but hence they are not considered a driver for preserving KaZSM. If our task was to preserve 'Palmerfield', then data loss as discussed here would be a driver for preservation.

6.1.1.3 Network

Since KaZSM is a distributed system, access to the network and performance of the network in the hosting environment are of primary importance. Unless each node is in communication with others, the data acquisition and processing pipeline will be disrupted bringing down the reliability of the entire system. In such a scenario, it may be necessary to migrate the entire system to run in a different data centre that has no network issues. This is a risk that impacts the primary function of the system and hence the impact justifies that a business continuity plan is implemented. The possession of such a capability proves to the consumers of the service that the solution is robust and can be relied upon and by inference, the organisation providing the service retains its credibility and the trust of its customers. This in turn helps to ensure the future prosperity of the organisation and therefore digital preservation is recommended to mitigate this risk.

6.1.2 Preservation approaches for IoT use case.

6.1.2.1 Technical components

Most of the supporting software components and underlying frameworks of the KaZSM prototype use open-source license models. The compelling offering of KaZSM is that it has integrated these components and implemented a process on top of them to perform a required function to end users. The licensing model however means source code and binaries for the components are available on many public repositories today. While it may be easy to source the latest versions of the software components from the public repositories, it may be time consuming to locate the older versions of these components and there is no guarantee that they will always be available into the future. For quick restoration when demand arises, it is recommended to preserve the software components of the exact version used by KaZSM. This also brings in other elements of the TIMBUS research as it will touch on legal aspects as to whether or not we are *legally* allowed to preserve the source code of these components. This is a separate consideration than if we possess the *technical* capability to be able to preserve the components.

All of the hardware used by the KaZSM prototype is virtual. However, the builds of the virtual hardware has been customised and is therefore not limited to standard images available with public cloud providers. Due to this, it is recommended that a single copy if each hardware build is preserved.

6.1.2.2 System configuration

Digital preservation of KaZSM is required primarily to ensure quick deployment of the entire system into a new hosting environment when the existing hosting environment fails to provide the required network performance. In a distributed computing environment, each node plays a bit part in the system. Each of the nodes, and all the other nodes, has to be aware of the role of the node for the system to function and perform. In KaZSM, Zookeeper does the job of centrally maintaining the configuration of all the nodes and monitoring, coordinating the activity. Zookeeper is the backbone of the system. Hence the specific version of Zookeeper, and its source code should be preserved with the configuration maintained by it. Again, we would make the distinction for the purpose of this discussion that TIMBUS gives us the technical ability to preserve these but we should never assume we have the legal right to preserve them so an assessment based on the legalities tool developed in TIMBUS is recommended for all of these.

6.1.2.3 Data

As discussed above, loss of historical sensor data causes only a minor impact to the business. Hence no preservation action is recommended for the Sensor data. An access control list for Data Hosting API needs to be preserved for quick restoration of service to data subscribers.

6.1.3 Redeployment Plan

Redeployment should be done in the following order (again, we will assume no legal impediment to this for the purposes of this discussion):

- *Create Virtual Machines*: Using the preserved virtual machine images, re-create the virtual machines in the new environment. Deploy software components: Install software from the preserved software artefacts as per the dependency graph.
- *Update zookeeper configuration*: Zookeeper is the key component which configures the entire system. Using the preserved configuration, update the zookeeper and prepare the system for re-start.
- *Restore the access control list*: For data consumers to be able to subscribe to the data, they will need their access configured. By restoring backed up ACL data, data consumers will gain access to the Data Hosting API.
- *Update DNS records*: For the sensors in the field to be able to post the data to KaZSM, the new server addresses have to be mapped to the domain address of KaZSM. Another assumption with networking is that at the time we re-deploy KaZSM, the concept of DNS and DDNS (dynamic DNS) will still exist as they do today and that IPv4 will still be understood by network infrastructure or that we will possess the ability to emulate a network layer which does understand IPv4. This

capability is available today in many virtualisation products and increasingly through SDN (Software Defined Networking) development we would expect such capabilities to not be 'show-stoppers' for our re-deployment in the long-term future.

- *Test*: Execute all the verification and validation tests and make sure all the system components are functioning.

6.2 WP7 – Use Cases

This section of the deliverable details out the interactions that the DPES Tasks T6.4, T6.6 and T6.7 have had in year 3 in interpreting the TIMBUS use cases from WP7 and our contribution to supporting their objectives through the development of the D6.6 tools to meet their requirements.

6.2.1 Introduction

Phaidra (Permanent Hosting, Archiving and Indexing of Digital Resources and Assets) and RCAAP ("Repositório Científico de Acesso Aberto de Portugal" – Portuguese Scientific Repository for Open Access are Open source repository systems, both OAIS compliant. Phaidra development is led by the University of Vienna, and is used in around 10 different universities in Central Europe. The use case in WP7 focuses however on the installation at the University of Vienna itself. RCAAP is a governmental effort to host the digital repositories of numerous academic and research institutions in Portugal. Phaidra is a digital asset management system with capabilities for long-term preservation, based on Fedora Commons.

6.2.2 Phaidra Technical Architecture

Running on a virtual VMWare ESX-Cluster, Phaidra comprises of a back-end and a front-end. The back end uses Fedora commons running on Red Hat Linux to manage the data. Digital objects are stored in a database, which has its data stored on a SAN (Storage Area Network). The front-end provides a user interface which is web-based, running on Apache Tomcat. Some of the key components of the system are listed below:

1. Red Hat Enterprise Linux 6: Server OS on which all the other components run.
2. Fedora Commons 3.3: Repository management system for archiving.
3. Apache Tomcat 6: Webserver to host the user interface application and the web services.
4. MySQL 5: Database management system used for storing the objects.
5. Catalyst 5.8: Perl based User Interface framework
6. Java 1.6: Java runtime for Fedora Commons
7. Perl 5.14: Perl runtime for Web application for User Interface which is scripted in Perl.
8. ImageMagik: Image conversion library used to provide a preview the images to the users.

In addition to above, Phaidra also utilises a set of Web services to get external content and also to render content such as videos to the users. High level architecture of Phaidra is illustrated in Figure 6-1 below.

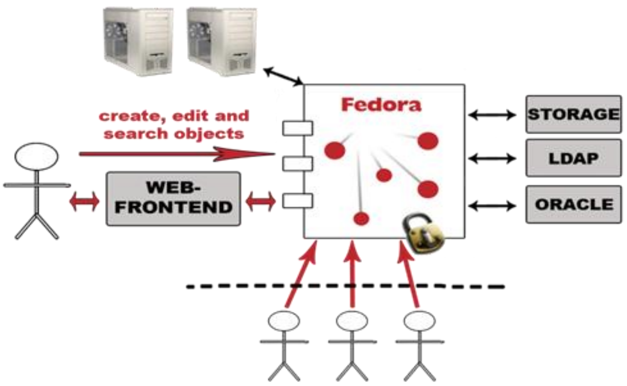

Figure 6-1: Phaidra High Level Architecture

A detailed picture of the Phaidra architecture in the context of University of Vienna is below in Figure 6-2.

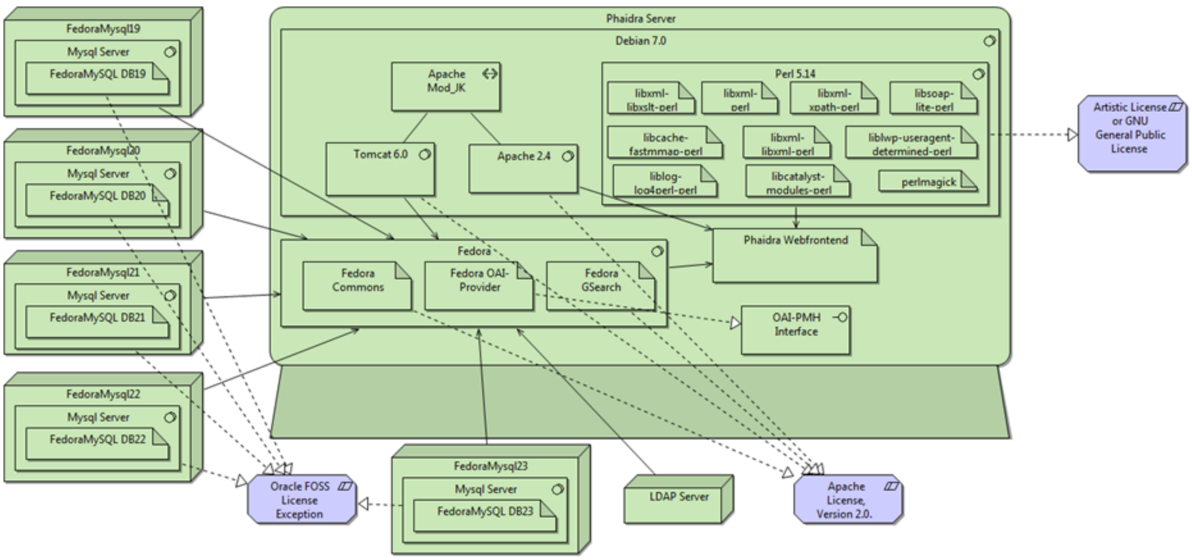

Figure 6-2: Phaidra Architecture

6.2.3 Phaidra Business Processes

The key role of Phaidra for the business is as a Repository. Usage of Phaidra can be broadly categorised into five business processes namely 1) Search Repository, 2) Retrieve Object, 3) Deposit Object 4) Metadata Editing and 5) Create Collection.

The average object size in a Phaidra instance in University of Austria is 25.60 MB while the largest object is 9.43 GB. Total repository size is about 1686 GB.

6.2.4 RCAAP Technical Architecture

RCAAP hosts multiple digital repository services to serve various groups of communities by providing hosting for specific types of digital assets. For example, The Scientific Data Hosting Service (SARDC) provides data sharing services for research institutions across the country. Similarly, The Scientific Journal Hosting Service (SARC) provides service to publish scientific journals. RCAAP provides common repository for all the institutions which do not have the capability and funding to host their own repositories.

RCAAP runs on the hardware infrastructure listed below:

- Six servers running DSpace components
- Two Linux virtual machines for Load balancing
- Two PostgreSQL database servers
- Two MySQL servers

RCAAP is a distributed system, primarily based on DSpace, running on a server cluster with redundancy. DSpace is mainly developed in Java and provides two user interfaces – one based on Java Server Pages and another based on Apache Cocoon. It uses PostgreSQL for data storage. In total RCAAP uses 27 instances of DSpace. In addition, RCAAP also uses an instance of OJS to provide services for SARC. OJS is an open source journal management system. It is based on PHP and uses MySQL for storage. The RCAAP services, and the infrastructure supporting them are depicted in the Figure 6-3 below.

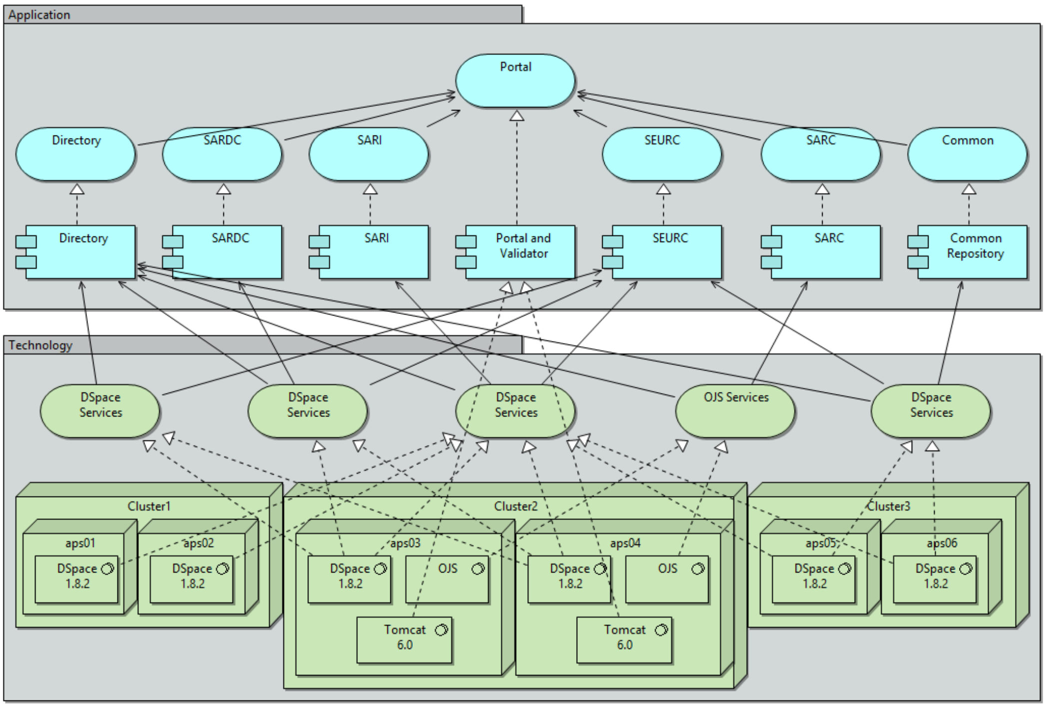

Figure 6-3: RCAAP services and the infrastructure

Business processes in RCAAP

The deposit process can be configured based on the requirements of the subscribing institutions. Broadly, the processes can be classified into categories shown in Table 27 below.

Table 27: RCAAP Business Process

S.No	Business Process	Description	Role
1.	Simple Process	This process consists of deposit of digital object followed by an approval and editing/addition/validation of metadata. In some cases approval is not required and object is available as soon as it is deposited.	Depositor, Collection/Community Co-coordinator
2.	Complex Process	It comprises multiple processes followed by the deposit. Some of these might be peer review, metadata editing etc followed by a final acceptance.	Depositor, Reviewer, Collection/Community Co-coordinator

DSpace contents are organised in a hierarchical manner as Communities at highest level and Collections at next level. Below that there are individual items.

6.2.5 Preservation approaches for Phaidra

6.2.5.1 Technical components:

Phaidra uses many external services to handle primarily, the presentation of digital object to the user. The core functions of deposit and retrieval of objects do not depend on external services. When the system is redeployed, if the services on which Phaidra are unavailable, the user experience is impacted. Hence it is desirable that all the services on which Phaidra is dependent are also preserved with the Phaidra core components. In cases where a service cannot be preserved due to lack of access/control, alternatives should be identified and preserved.

6.2.5.2 Data

In Phaidra, all the digital objects and the metadata should be preserved so that the resources are retained when restored.

6.2.6 Preservation approaches for RCAAP

6.2.6.1 Technical Components:

RCAAP itself has a federated search that contacts individual repositories, thus has external dependencies. If these are not there the system might not fail, but behave differently (i.e. have different results & contents). Hence, it is important to capture those external service dependencies but repositories are customised for specific organisation need. So, without the need of external services the system can be redeployment in the future time to redeem the basic functionalities.

6.2.6.2 Data

RCAAP has more complex workflows than Phaidra and a digital object could change during these workflows. Hence it is a requirement to preserve the change history of the object as well as the metadata to be able to satisfy any questions on trustworthiness of the data.

6.2.7 Re-deployment plan for Phaidra

Redeployment should be done in the following order:

- *Create virtual machines*: Using the preserved virtual machine images, re-create the virtual machines in the new environment
- *Deploy software components*: Install software and service dependencies from the preserved software artefacts as per the dependency graph.
- *Update configuration*: Update the configuration required to establish network communication
- *Update DNS records*: For the users to be able to use the system, the new server addresses have to be mapped to the domain address
- *Test*: Execute all the verification and validation tests and make sure all the system components are functioning.
- *Stakeholder communication*: As needed communicate to stakeholders

6.2.8 Re-deployment plan for RCAAP

Redeployment should be done in the following order:

- *Create virtual machines*: Using the preserved virtual machine images, re-create the virtual machines in the new environment
- *Deploy software components*: Install software using the preserved software artefacts as per the dependency graph
- *Update network configuration*: Update the configuration required to establish network communication
- *Update repository configuration*: Update the repository configuration to restore services as required by various organisations
- *Update DNS records*: For the users to be able to use the system, the new server addresses have to be mapped to the domain address

- *Test*: Execute all the verification and validation tests and make sure all the system components are functioning

6.3 WP8 – Use Case

Preservation of CAD/CAM services of Civil Engineering Project (LNEC). This work is based on our understanding of the work package 8 use cases, their requirements and how our work in D6.6 can support and meet their requirements.

6.3.1 Introduction

The preservation of CAD/CAM designs presents some unique challenges. The Portuguese National Laboratory for Civil Engineering (LNEC) is involved in the construction and maintenance of large civil engineering infrastructures, namely dams. Many of their business processes are dependent on CAD/CAM. These are of high impact in terms of safety of the people and goods with business continuity being very important. It is also highly impactful to LNEC future business undertakings if they cannot demonstrate an unerring ability to design, test, manage and contribute to the operational excellence of these facilities and thus maintain the trust and confidence of their customers long in to the future.

LNEC work entails the development of monitoring and analytics solutions for the structural behaviour of dams, not just new dams under construction but also dams under operation. Figure 6-4 below shows a selection of photographs from a visit of the TIMBUS consortium to one of these dams in Portugal in year 2. Through sensors installed on the structures, parameters such as temperature, displacement and stress are measured to help evaluate stability and safety of the dam. Data from the sensors is manipulated and applied on the CAD design to get the understanding of the impact through a graphical representation.

Figure 6-4: Photos from a TIMBUS consortium trip to a dam in Portugal hosted by LNEC in year 2

Historically CAD was aimed at making the design process more productive by making it easier to re-use existing designs. These days, with the development of 3D techniques, CAD is used much more widely and in many areas of study for analysis and research. In many cases, these CAD designs are used over long periods, such as in the case of LNEC and require long term preservation strategy.

6.3.2 Technical Architecture

A key issue with the CAD models is the dependence of each design on particular CAD software and/or a specific version of the software. However there have been efforts to standardise the design formats. In the context of LNEC dam monitoring, key technical components used are:

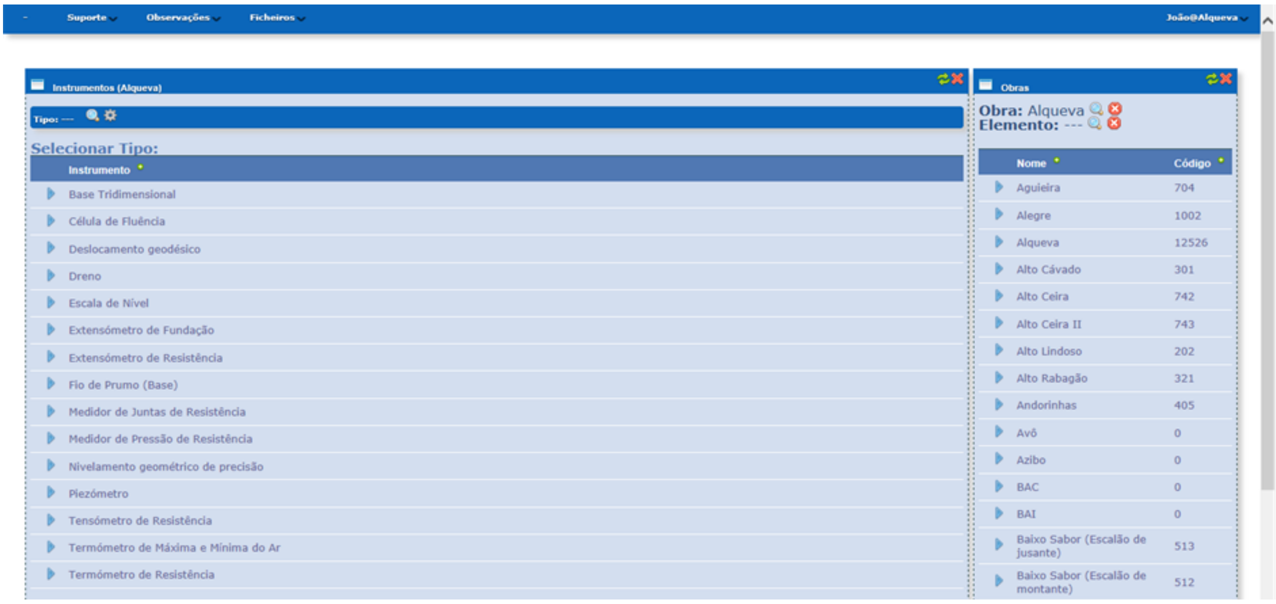

Figure 6-5: Screen Capture of GestBarragens UI

GestBarragens: This is a modular system which manages data from the dam Sensors, visual inspections, physical models and mathematical models. It has capabilities to manage technical documents and has a set of analytical tools to analyse the data. Based on the data obtained from a dam, GestBarragens can create CAD designs for further analysis. GestBarragens has a set of components built using VB.Net, Matlab & FORTRAN to achieve this. A screen capture of GestBarragens UI is illustrated in Figure 6-5.

Auto CAD: It is a popular CAD software package used to create and analyse CAD designs. It is commercial software developed by Autodesk Inc.

Paraview, GiD: Paraview and GiD are data visualisation software packages used to visualise results of CAD analysis.

FORTRAN: It is a programming language that is popular in scientific and engineering communities. Many commercial and free distributions of FORTRAN compilers are available.

.Net: It is a widely used software development framework developed and provided by Microsoft Corporation. It is used in LNEC for creating CAD from Sensor data.

6.3.3 Business Process

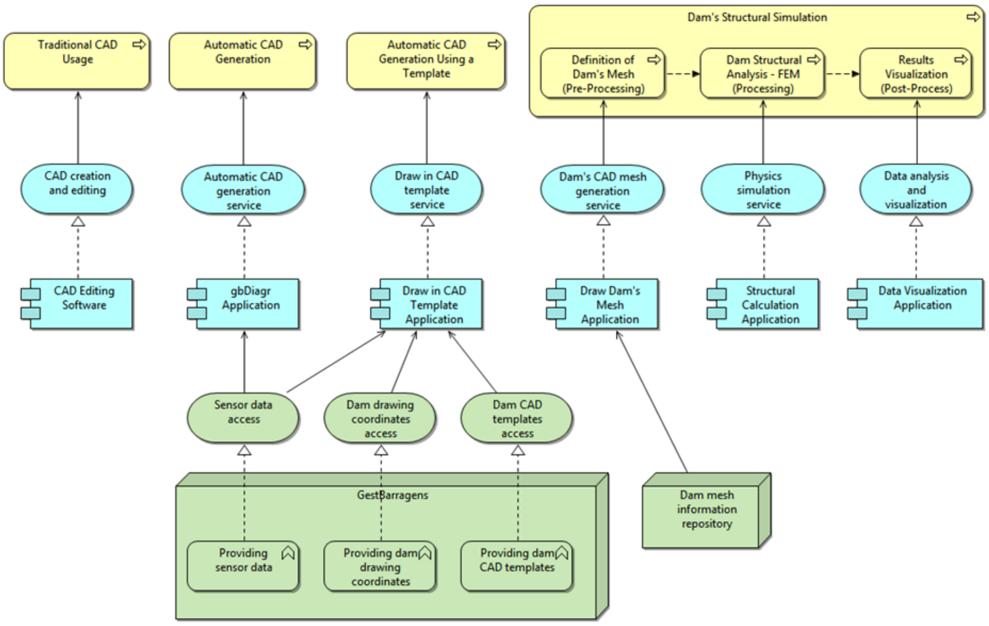

Figure 6-6: LNEC DAM Monitoring Business Process

Figure 6-6 shows a representation of the LNEC dam monitoring business processes. All of these are built around GestBarragens and involve creating or modifying CAD designs. Details of these are shown in Table 28 below:

Table 28: LNEC Business Process

Name	Description	Tools
Traditional CAD Usage	For new dams, the dam owner provides a CAD design which is then opened in CAD software such as Auto CAD and converted to a Template for usage later with sensor/observation data gathered from the dam. Another scenario is where a generated CAD design is opened in Auto CAD and modified.	Auto CAD, GestBarragens
Automatic CAD Generation	In GestBarragens, users can request a CAD to be generated by selecting a dam and output options. GestBarragens then uses the sensor data and generates a CAD. A line diagram with sensor data on a time scale is generated.	GestBarragens
Automatic CAD generation	In this case, sensor data is represented on the design of the dam, rather than as a line diagram. This is required in situations	GestBarragens, VB.Net application

using a template	where a spatial analysis needs to be carried out.	
Dam's structural simulation	User can request GestBarragens to do structural simulation of a dam. Using the finite element mesh data and dam's structural information, a CAD is generated that is then visualised in GiD or ParaView.	GestBarragens, FORTRAN, GiD, ParaView

6.3.4 Preservation Requirements

LNEC business processes are of critical nature with safety of people and goods dependent on these processes running smoothly. This factor indicates that a strong, reliable and trustworthy business continuity plan has to be in place.

In this context, preservation needs to meet some key requirements:

- Trust – The processes need to re-execute and provide exactly same results upon re-deployment.
- Correction – It should be possible to re-execute process with modified data and get valid results.
- Reanalysis – Business process should re-execute using any new data available combining it with the previously available data.
- Continuity – Business processes needs to preserve with capability for long-term re-execution i.e. it should be possible to re-execute it after a very large time gap in terms of decades.

6.3.5 Preservation Approaches for CAD/CAM Environment

6.3.5.1 Technical Components

GestBarragens is the most critical component of LNEC dam monitoring system. It also holds all the sensor data that is critical to continuing operations. Binaries of GestBarragens and the VB.Net/FORTRAN components which are used to generate CADs should be preserved. Binaries of CAD tools in use should be preserved along with licenses. This is to mitigate risk of non-availability of tools at a future date.

OS images of all the server hardware used should be preserved. Since creating full server images the actual hardware might cause an interruption to the business process, it is advisable to preserve vanilla OS images. This means during the redeployment, all software components will have to be re-installed and configured.

6.3.5.2 Data

All the CAD designs in the GestBarragens data store should be preserved. Before preservation, the CAD designs that are in proprietary format should be converted to a standardised format to avoid any later interoperability issues. Another more flexible option to prevent interoperability issues is to preserve the conversion software along with the designs. This way conversion can be optionally performed if required when redeploying and doesn't need to be done during preservation planning and execution. Sensor data needs to be preserved as time based analysis of the data is very important for the business.

6.3.5.3 Configuration

All configurations including access control list, should be preserved to allow for expeditious redeployment in the event of disaster recovery.

6.3.6 Redeployment Plan

Redeployment should be done in following sequence:

1. *Create virtual machines*: Using the preserved virtual machine images, re-create the virtual machines in the new environment
2. *Deploy software components*: Install software from the preserved software binaries for GestBarragens and as per the dependency graph
3. *Restore the data*: All the sensor data and CAD data should be restored to GestBarragens system. If conversion of CAD format is necessary, do the same
4. *Restore access control list*: In order for the users to be able to use the system, access needs to be granted using the preserved ACL
5. *Update network configuration*: For various components to communicate with each other, new network addresses will have to be updated in the configuration
6. *Test* : Execute all the verification and validation tests and make sure all the system components are functioning

6.4 WP9 – Use Case

The information in this section of the deliverable reflects our thoughts and analysis of how the T6.4, 6.6 and 6.7 tools can best assist the eHealth use case from work package 9 and how we can go about such a preservation task. It follows a similar structure to the previous use case examples already covered in this section of the deliverable.

6.4.1 Introduction

European legislation mandates that all the prescription drugs contain information on how they work and what the intended effects are. Often a drug may have an adverse reaction on the patient. This is termed an Adverse Drug Reaction (ADR). The study of ADRs is conducted in the field known as pharmacovigilance. A more general term Adverse Drug Event (ADE) is used to refer to any injury caused by the drug and the harm associated.

ADE is a serious problem that medical practitioners need to be aware about when prescribing medicines. The e-Health use case aims to provide a solution to help address this problem. The solution offers capability to discover and search ADE rules. It provides subscription based services to doctors and pharmacists. It provides information on possible risks for a particular combination of drugs. In the discovery process data is obtained from CMRDP and a complex set of algorithms are run to create information that can then be used in the search process. The search process is also based on complex algorithms. Every time a recommendation is provided, results of the recommendation are used to tune the discovery algorithms.

6.4.2 Technical Architecture

The technical infrastructure of the ADE rules discovery and search system is distributed between three entities that are involved in the process. The infrastructure at DrugFusion is primarily based on Dell PowerEdge series servers, DataMole infrastructure is based on IBM servers and SemanTech hosts HP ProLiant based server infrastructure. Further details and purpose of each server is listed in the use case deliverable D9.3.

An illustration of the technical environment is provided in the Figure 6-7 below. The system depends on CMRDP web service for data for the discovery process. It also uses a rules validation service provided by Drug data companies to validate ADE rules.

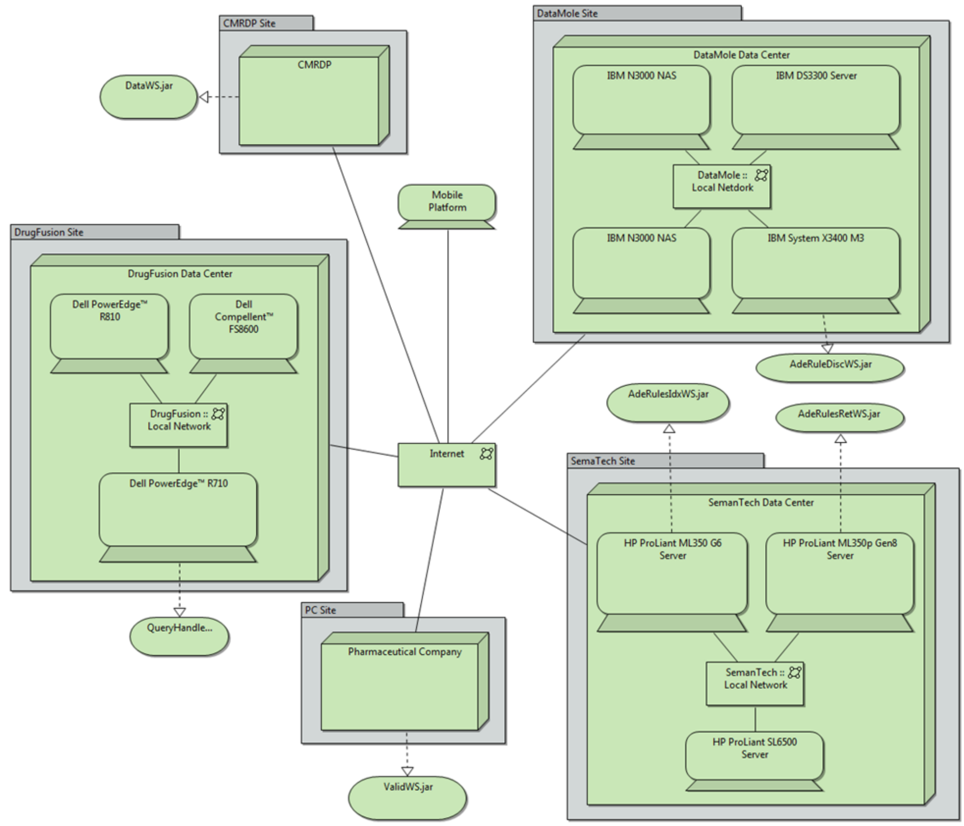

Figure 6-7: eHealth Technical Environment

Software used is mostly Java based and the Table 30 below lists the software packages and the server hardware on which they run.

Table 29: eHealth Software Resources

Platform	Installed Software Components	Entity
Dell PowerEdge™ R810	AdeWorkflow.jar	DrugFusion
Dell Compellent™ FS8600	FileZilla Packages	DrugFusion
Dell PowerEdge™ R710	QueryHandlerWS.jar QueryHandler.jar AdeRulesSearch.jar ResultNavigator.jar InternalCache.jar	DrugFusion
IBM System X3400 M3	AdeRuleDiskWS.jar	DataMole

	AdeRuleDisk.jar	
IBM N3000 NAS	MySQL Packages	DataMole
IBM DS3300	FileZilla Packages	DataMole
HP ProLiant SL6500 Server	MySQL Packages	SemanTech
HP ProLiant ML350 G6 Server	MySQL Packages	SemanTech
HP ProLiant ML350p Gen8 Server	AdeRulesRetWS.jar AdeRulesRet.jar	SemanTech

6.4.3 Business Process

The three entities involved – DrugFusion, DataMole and SemanTech run different steps in the system, which together, provide end to end capability of the ADE Rules Discovery and Search system. Further details about eHealth business process can be learnt from Deliverable D9.3.

6.4.4 Preservation approaches for eHealth

6.4.4.1 Technical Components

Key software components of the ADE Discovery and Search system are based on Java. This makes them highly portable across the operating system platforms and also the hardware platforms. Data platform used (MySQL) is also highly portable. Thus there is no hard dependency on the hardware or OS.

The software components however are highly customised and preserving these along with the ADE Rules database with guaranteed integrity is crucial. All JAR files should be preserved.

6.4.4.2 Data

ADE Rules Database should be preserved with all the data. In addition, a quarterly maintenance activity should be carried out to update the preserved data with data from live system.

6.4.4.3 Configuration

The system is distributed with network access required over local network as well as Internet. Hence the entire network related configuration should be preserved.

6.4.5 Redeployment Plan

Redeployment should be done in following sequence:

- *Create virtual machines*: Choose an OS configuration that can host a Java Virtual Machine and create Virtual machines to deploy software components
- *Deploy software components*: Install software from the preserved JAR files as per the dependency graph
- *Restore the data*: Install MySQL server and restore the ADE rules database
- *Update network configuration*: For various components to communicate with each other, new network addresses will have to be updated in the configuration.
- *Update DNS records*: For doctors and pharmacists to access the system over public network, addresses of new servers should be updated in the public DNS records.
- *Test*: Execute all the verification and validation tests and make sure all the system components are functioning.

7 Implementation and Integration

This section reports out the implementation work completed to date on Tasks T6.4, T6.6 in addition to implementation work submitted in chapter 9 of deliverable D6.4 on month M30. The following is the new or updated implementation work since the last deliverable. Refer:

7.1 iERM and DPES API

DPES provides the DP Expert with a UI and tools to manage a preservation project from start-to-finish. For this DPES provides API capabilities to integrate various tools such as context modellers, risk managers, artefact extractors etc. that provide inputs needs for preserving a business process.

These external tools interact with DPES through a Restful API. Details of the interaction between DPES and iERM (Risk Analysis tool) are described below as an illustration in sequence diagram shown in

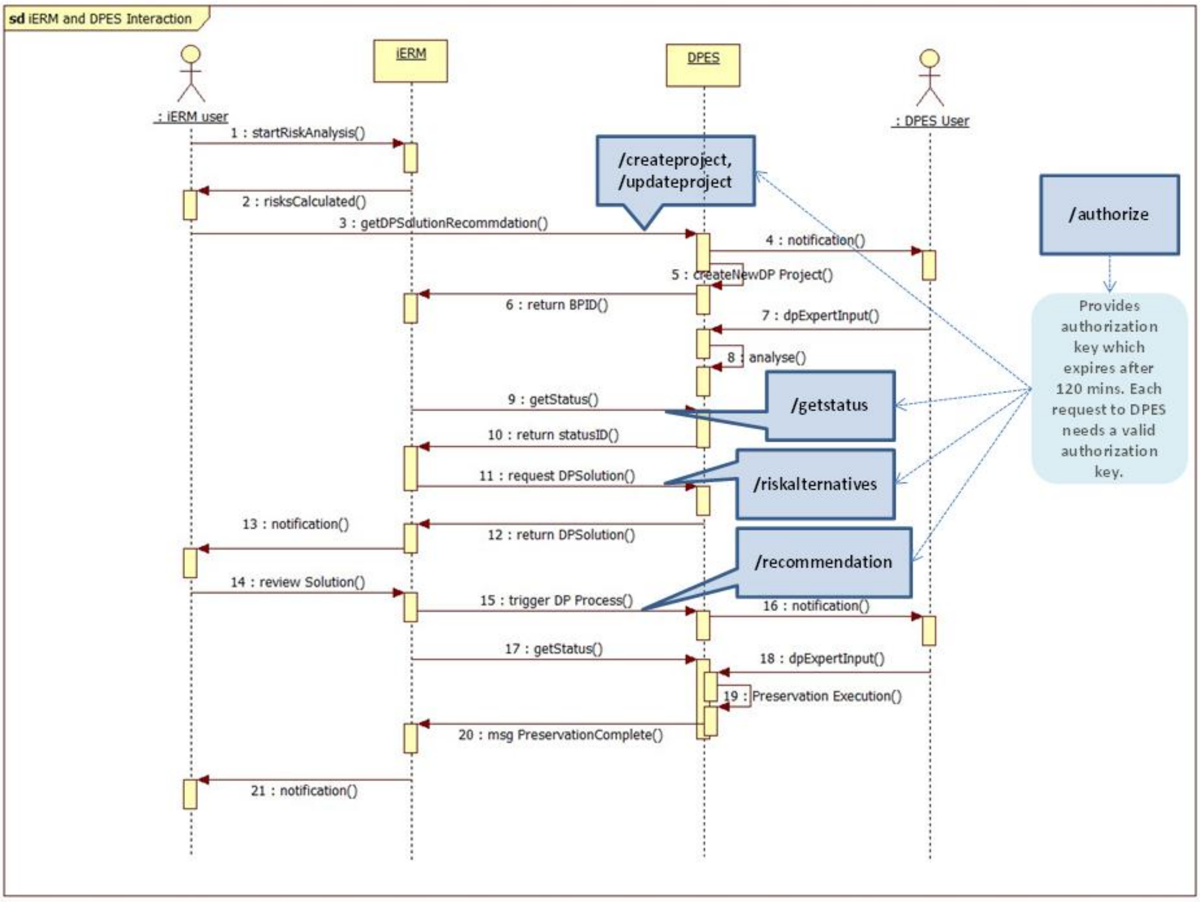

Figure 7-1:

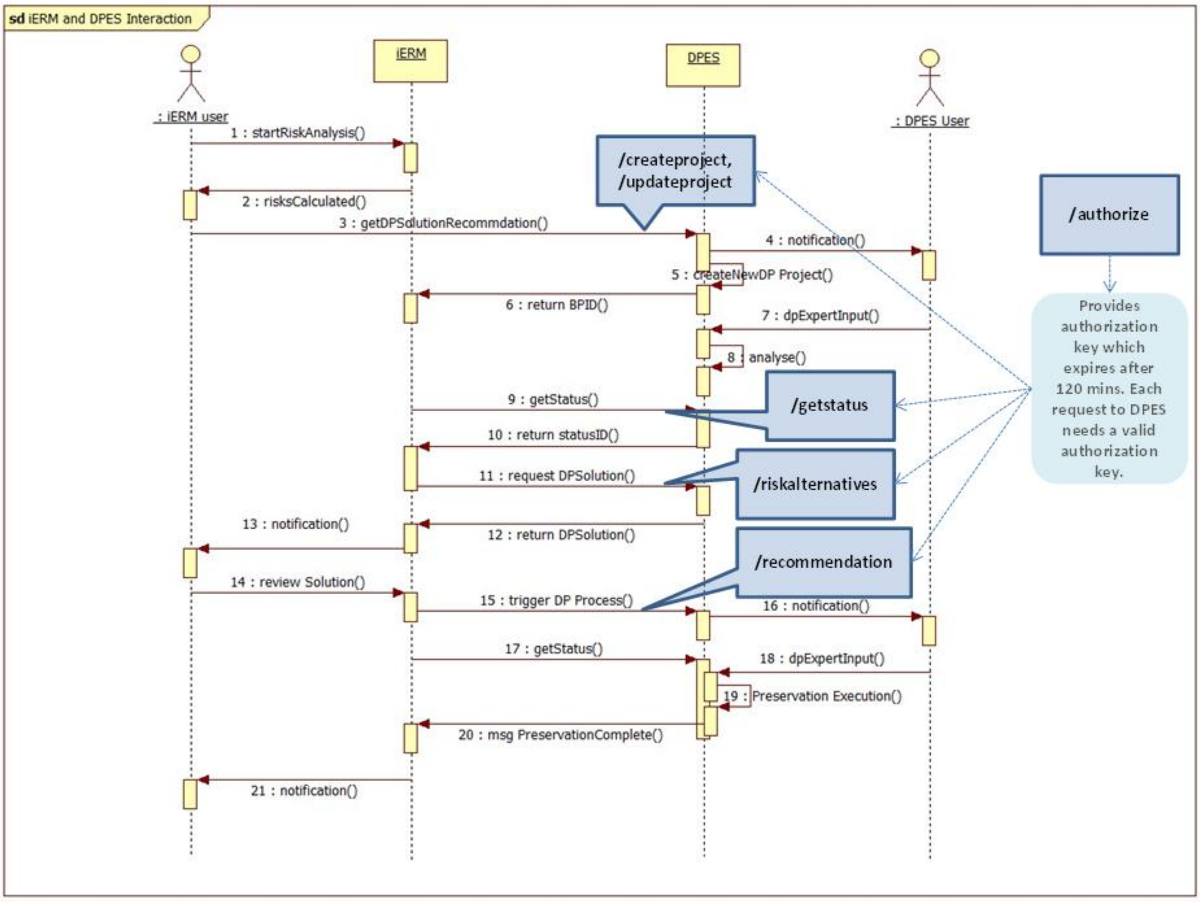

Figure 7-1: Sequence Diagram of DPES and iERM Tool

Table 30: DPES – iERM API calls

API Method	Description
createproject	Creates a business process preservation project. A Risk document or a Context model needs to be provided to initiate a project.
updateproject	Used to update risks or context model for an existing project.
Getstatus	Provides the current status of a preservation project
riskalternatives	Provides cost projection for preserving the business process and any preservation alternatives identified. This will have the information required by Risk manager to decide if the business process should be preserved.
/recommendation	Risk Analysis Tool uses this method to communicate the decision/recommendation of the Risk manager to DPES.

Details of the DPES API for Risk Analysis Tool

Table 31: DPES API to create project

Endpoint	/api/project/create
Input	• Authorisation Token (required as HTTP header) • Preservation Project Name (required) • Risk Document (xml) (optional) • Ontology Document (xml/rdf) (optional)
Output	• Project id (String)
Sample output	25
Notes	1. A new Preservation Project will be created and the ID of the new preservation project will be returned. 2. If name is duplicate, a warning message also will be in the output. 3. Either Risk Document or Ontology document is required to create a project.

Table 32: DPES API to update Project

Endpoint	/api/project/[project id]/update
Input	• Authorisation Token (required as HTTP header) • Project Id (required) • Risk Document (xml) (optional) • Ontology Document (xml/rdf) (optional)
Output	• Project id (String)
Sample output	25
Notes	1. Error if Preservation Project ID is not found or if Project status is 'pre-served'.

Table 33: DPES API to get Status

Endpoint	/api/project/[project id]/status
Input	• Authorisation Token (required as HTTP header) • Project Id (required)
Output	• XML with status code and status text
CURL example	`curl -X POST -H "authenticationKey:6473c86c-1f10-4c16-a1bd-f10a7c7ee1a9" http://134.191.240.68:16080/dpesDev/api/project/2/status`
Sample output	`<?xml version="1.0" encoding="UTF-8" standalone="no"?><status><statusCode>50</statusCode><statusText>ALTERNATES RECEIVED FROM PRESERVATION IDENTIFIER</statusText>`

	<statusUser>svoorakk</statusUser> <statusTime>2014-01-20 16:38:26</statusTime> </status>
Notes	Status codes : 10 - CREATED 20 - CONTEXT MODEL UPDATED 30 - RISK DOCUMENT UPDATED 40 - RISKS SENT TO PRESERVATION IDENTIFIER 50 - ALTERNATES RECEIVED FROM PRESERVATION IDENTIFIER 60 - COST ANALYSIS IN PROGRESS 70 - COST ANALYSIS AVAILABLE 80 - RISK ANALYSIS RECOMMENDATION - PRESERVATION RECOMMENDED 81 - RISK ANALYSIS RECOMMENDATION - PRESERVATION NOT RECOMMENDED 90 - ARTIFACTS ACQUISITION IN PROGRESS 100 - ARTIFACTS ACQUIRED 110 - METADATA PREPARED 120 - PRESERVATION CONTAINER(S) CREATED 130 - PRESERVATION COMMENCED 140 - PRESERVED TO REPOSITORY 150 - DELETED

Table 34: DPES API for Risk Alternative

Endpoint	/api/project/[project id]/riskalternatives
Input	• Authorisation Token (required as HTTP header) • Project Id (required)
Output	• XML string with alternates, costing and feasibility information
CURL Example	`curl -X POST -H "authenticationKey:6473c86c-1f10-4c16-a1bd-f10a7c7ee1a9" http://134.191.240.68:16080/dpesDev/api/project/2/riskalternatives`
Notes	This endpoint will return links to all the alternative ontologies identified by Preservation Identifier for all the risks in the risk document and also links to the costing document for each alternative for which DP has provided cost inputs.

Table 35: DPES API for Recommendation

Endpoint	/api/project/[project id]/recommendation
Input	• Authorisation Token (required as HTTP header) • Project Id (required) • Recommendation to preserve (Yes/No)
CURL Example	`curl -X POST --data "preserve=true" -H "authenticationKey:6473c86c-1f10-4c16-a1bd-f10a7c7ee1a9" http://134.191.240.68:16080/dpesDev/api/project/2/recommendation`
Notes	1. Recommendation is 'Yes' or 'No' for going ahead for preservation. 2. If recommendation is 'Yes', it is expected that iERM returns the Alternative ontology that is recommended to be used for preservation. 3. Error if Preservation Project ID is not found .

Table 36: DPES API for user authorisation

Endpoint	/api/[user id]/authorise
Input	• User Id (required) • Password (Required)
CURL Example	`curl -X POST --data "password=xxAAA" http://134.191.240.68:16080/dpesDev/api/user/test_user/authorise`
Sample Output	`<?xml version="1.0" encoding="UTF-8" standalone="no"?><authToken><authenticationKey>6473c86c-1f10-4c16-a1bd-f10a7c7ee1a9</authenticationKey><expiryDate>2014-01-20 13:59:15</expiryDate><loginDate>2014-01-20 11:59:15</loginDate></authToken>`
Notes	1. It is a pre-requisite that a user registers in DPES before any access to DPES occurs through iERM. 2. Authorisation token will have a definite expiry duration (proposed to be 120mins). 3. Any method requiring Authorisation Token will return error if an ex-

	pired token is provided.

7.2 Storage Client Implementation

The UML diagram shown in Figure 7-2 implements the TIMBUS storage using AMAZON S3 SDK. DPES create an account with S3 service to store artefacts of preserved business process. Secret code and access codes are generated after creating account with S3 to integrate S3 service with DPES. DPES will automatically manage storage task from GUI based user action rather logging into Amazon S3 system.

Classes: IRespositoryClient, Preservation, S3Client

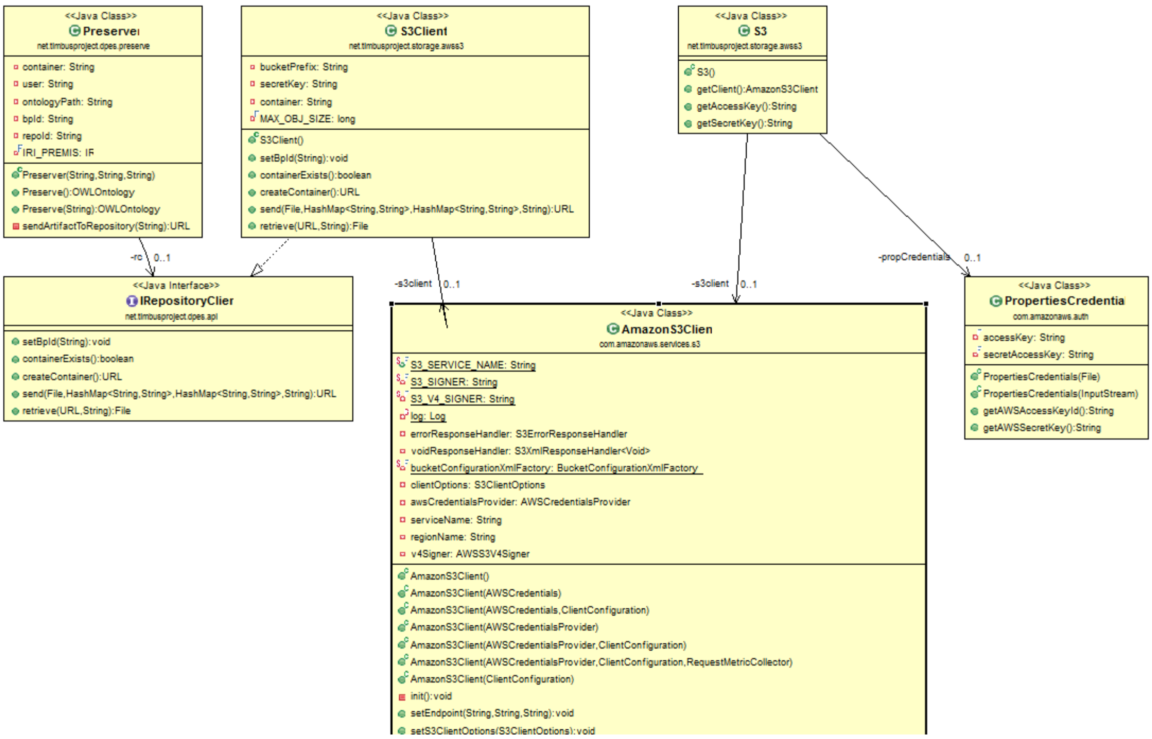

Figure 7-2: DPES Storage Client Side Implementation

7.3 DPES User Interfaces

The DPES software user interface has been redesigned from previous version submitted in the deliverable D6.4 in M30 to provide better navigation experience to the user during preservation process. A list of screen captures from UI is presented in appendix 10.1 of this deliverable describing the complete preservation with mock data.

7.4 Re-Run Environment (T6.6)

There are two main redeployment approaches considered, the following sections describe each type in more detail. During the preservation process, a suitable redeployment approach will be identified and tested in the specific redeployment environment before preserving artefacts in the repository to ensure that all the business artefacts are captured and also verifies the chosen redeployment model is functionally complete.

7.4.1 Linux Container Based Redeployment

Linux containers are discussed in the SoTA Section 3.5 software containers, where it describes and compares the features and benefits of various containers. Docker [63] is a component considered in an example of constructing a container based redeployment model for TIMBUS because it is an open source project to create lightweight, portable self-sufficient run time containers for any application. Several community contributed Docker based images are readily available to run the application.

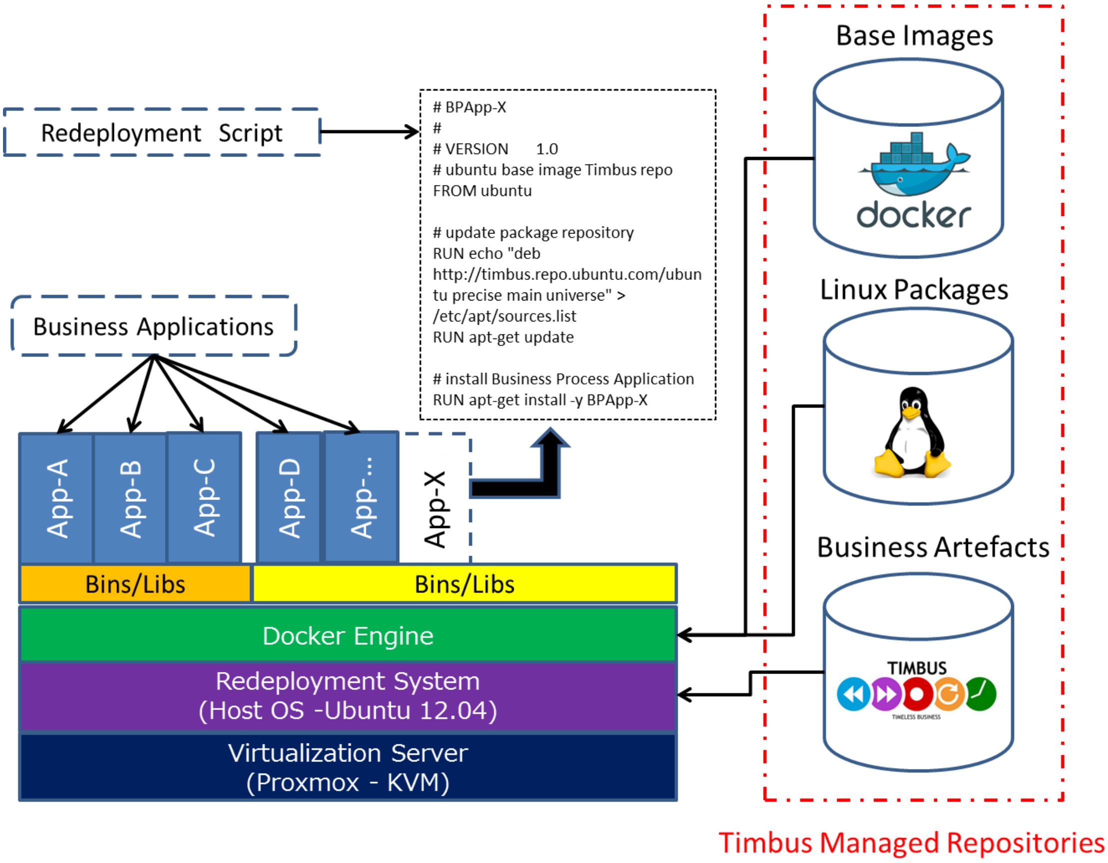

Figure 7-3 Linux Container Based Redeployment Model

Timbus Managed Repositories

Figure 7-3 shows the redeployment model where business process specific Docker base images are imported as part of artefacts sourced from community repositories and stored in the TIMBUS repository. The redeployment process is executed by running a script on the host machine (Test Bed); the script mounts base images of business process operating systems and installs libraries and business artefacts from the preservation repository. PROXMOX virtualisation server shown in the diagram is optional as it just reflects the existing TIMBUS Test Bed environment. In general Docker container can be run on any (virtualised or physical OS) compatible host operating system.

7.4.2 Redeploying in a Virtual Machine

In this deployment model, a new virtual machine will be created using the hardware specification and operating system described in the context model on the virtualisation server. Business artefacts and dependent libraries are installed semi-automatically using scripts and tested for function completeness using process verification checks provided by business owners. Figure 7-4 shows the model which imports full operating system ISO images to build a virtual machine using KVM on a virtualisation server. Hardware emulation can be used for specific parts of the IT system to replicate the original business configuration such as emulation of computer processers using QEMU. Figure 7-5 shows the list of emulated CPU types available in PROXMOX virtual server. This type of redeployment is suitable for business environments that are heavily dependent on original system hardware resources.

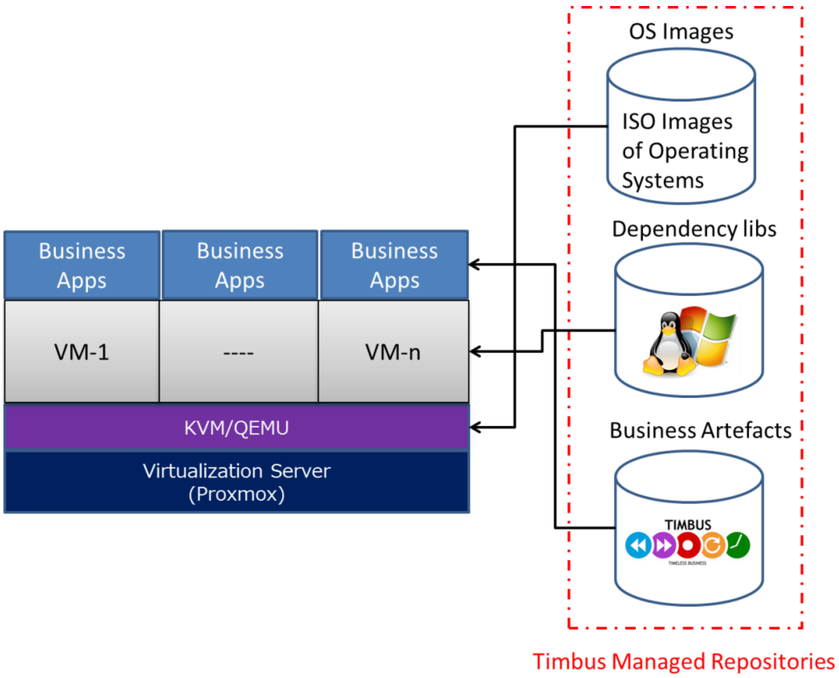

Figure 7-4: Redeployment using Virtual machines

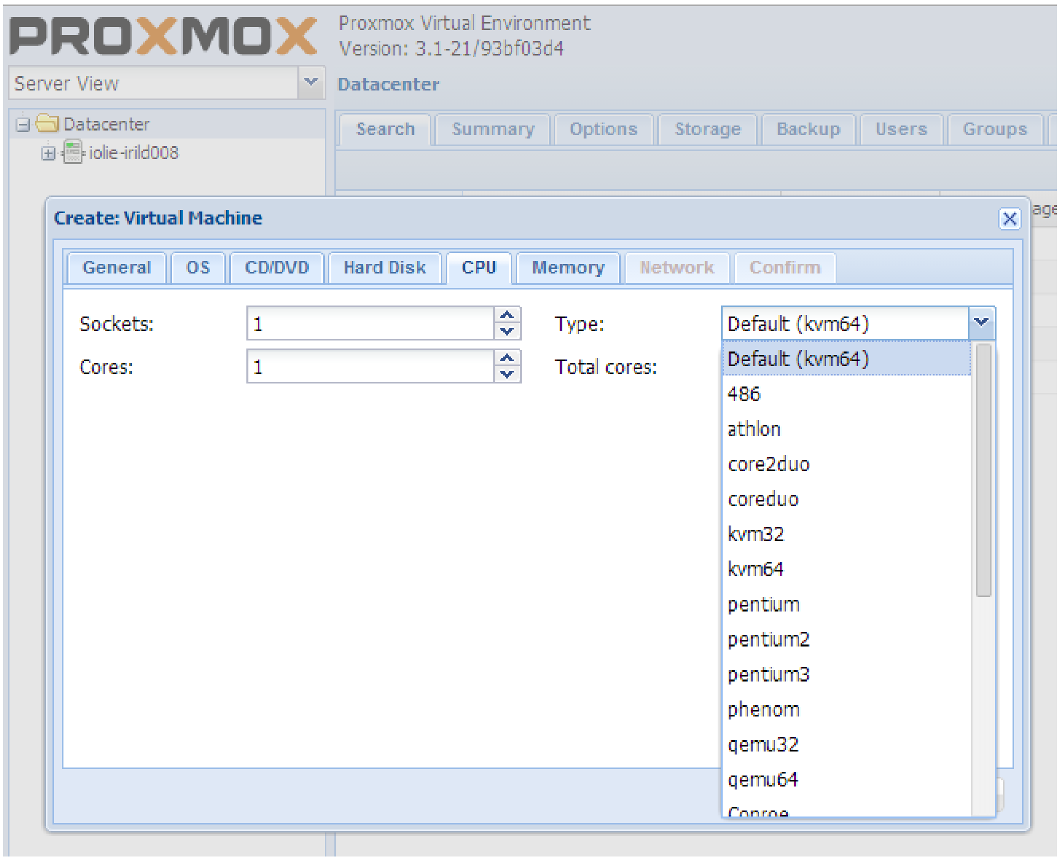

Figure 7-5: Virtual machine using Emulated CPU

7.5 Status of Tools and Services Development

This section discusses the current implementation status of the DPES system and also the future implementation plan for remaining features.

Table 37: DPES Software Implementation Status

No.	Module Name	Status	Implementation Plan
1	Feasibility Analyser	Implemented	M30
2	Cost Modelling	Implemented	M30
3	Preservation Identifier Integration	to be Implemented	M42

4	Metadata Generation	Implemented	M30
5	Redeployment Server Integration (PROXMOX KVM-QEMU)	Implemented	M30
6	User account Management	Implemented	M30
7	Local Registry	Implemented	M30
8	Use case Requirements gathering	Completed	M36
9	Redeployment Model	Completed	M36
10	DPES GUI redesigned	Implemented	M36
11	Storage Service Integration	Implemented	M36
12	iERM and DPES API Integration	Implemented	M36
13	Integration with Extractors and Context Query	To be Implemented	M42
14	Local Registry for Docker / Linux packages	To be Implemented	M42
15	Metadata packaging and PREMIS owl file annotation	To be Implemented	M42
16	Management UI function	To be Implemented	M42
17	Redeployment container automation	To be Implemented	M42

8 Building, deploying and Testing DPES application

This chapter provides updated instructions on using the DPES software.

8.1 Development Environment

The DPES development environment comprises of the following development tools:

- Java Development Kit (JDK) (version 1.7_17)
- Apache Maven (version – 3.0.5)
- Eclipse IDE (version – Juno)
- Flex builder 3 for GUI Framework
- Tomcat Web Servlet Container (version – 7)
- SVN plugin for eclipse
- H2 data base (version – 1.3.173)

8.2 Deployment Requirement

The following services and resources are needed for deploying DPES application.

- Java Run Time JRE-JVM (version - 1.7)
- Apache Tomcat Server (version - 7)
- DPES War bundles files
- MySQL database Server (version - 5.5)
- Amazon S3 account to configure storage credentials
- Network connectivity to access to TIMBUS Services and Ontology
- KVM+QEMU Virtualisation Server (PROXMOX)

8.3 Installation and Testing

Follow the below instruction to download DPES source code, compile and deploy them into the Test Bed.

- svn checkout https://timbus.teco.edu/svn/dev/digital-preservation-expertsuit-intel/DPES-Flash-UI for flash UI project code, then import into Flex builder for any further development.
- svn checkout https://timbus.teco.edu/svn/dev/digital-preservation-expertsuit-intel/preservation-manager-intel/trunk/dpesWebapp for java back-end implementation software.

- Alternately source code can be imported directly using the Eclipse svn plugin as a maven project using the above URL.
- In the command line interface, go to the project root directory and type the *mvn package* command to download dependencies and perform packaging.
- Finally from Eclipse, choose export and select type as *WAR* file and then type file name folder location to save the DPES web application file as shown in Figure 8-1.

NOTE: Several alternate methods are available for all or some of the installation steps. For example any command executed through a GUI interface may also be possible to be executed through a command line interface. We have outlined above the methods popularly utilised by TIMBUS developers.

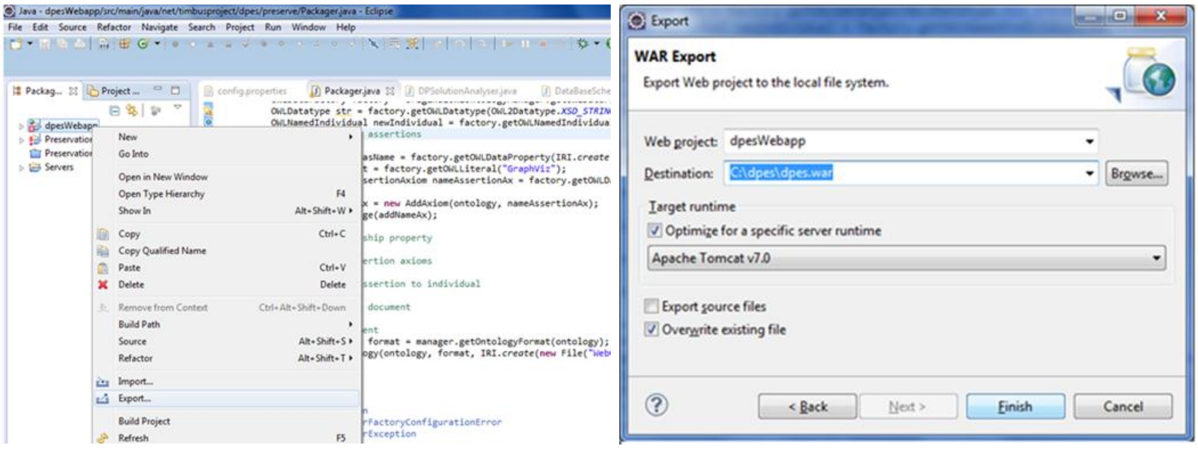

Figure 8-1: Export DPES WAR file

- To install the DPES web application into the Tomcat server, go the Tomcat admin URL (TIMBUS DPES Test Bed URL: http://134.191.240.68:12080) and click *manager app* button as shown by red arrow in the Figure 8-2 and then click choose file under Deploy ->WAR file to deploy.
- Use the application tab from the Tomcat admin console to provision (stop, start, reload and undeploy) the DPES app and monitor the status. Use the following TIMBUS Test Bed URL to browse the application.
    - DPES Development site: http://134.191.240.68:16080/dpesDev/
    - DPES Release site: http://134.191.240.68:12080/dpes/

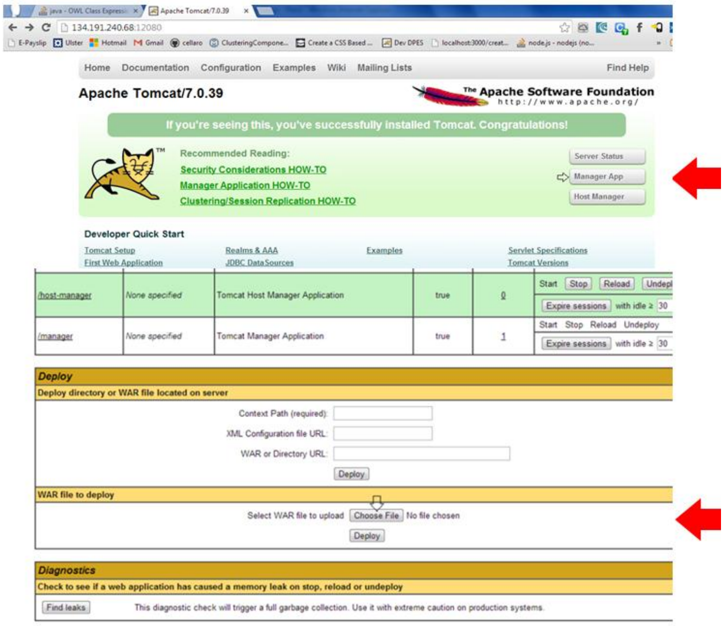

Figure 8-2: Deploy DPES WAR file into Tomcat Server

9 Conclusion and Future Work

At the end of year 3, the development of tools and services under Tasks T6.4 and T6.6 are well underway to demonstrate the preservation process of use cases presented in WP7/8/9 as well as in Intel's IoT use case. During the last six months, since the last update on these tasks in M30, there are a list of works completed that are detailed in Table 37, namely analysing the use case scenarios, understanding DPES requirements from use case perspective and studying the redeployment approaches using technologies such as Linux containers and virtual machines and designing a redeployment model. Also in addition to the software implementation work reported in month M30, new implementation work has taken place on the storage service and iERM. The DPES service API has been implemented and a redesign and development of a flash-based DPES GUI has been completed.

As we face in to the closing stages of the TIMBUS project, these tasks continue to execute with further implementation of the remaining components as shown in Table 37. Further integration of the TIMBUS tools such as extractor, context model querying and annotation with storage Meta information is also planned.

Future work will also focus on improving the existing implementations while testing them on use case preservation and redeployment activities, and discovering any shortfall in the system for further enhancement. All this work supports partners' exploitation plans. These plans are dependent on WP6 being able to produce impressive prototypes which integrate all the TIMBUS tools and provide capabilities not otherwise available to consumers today.

10 Appendix

10.1 DPES Screen Captures

Figure 10-1: DPES Login

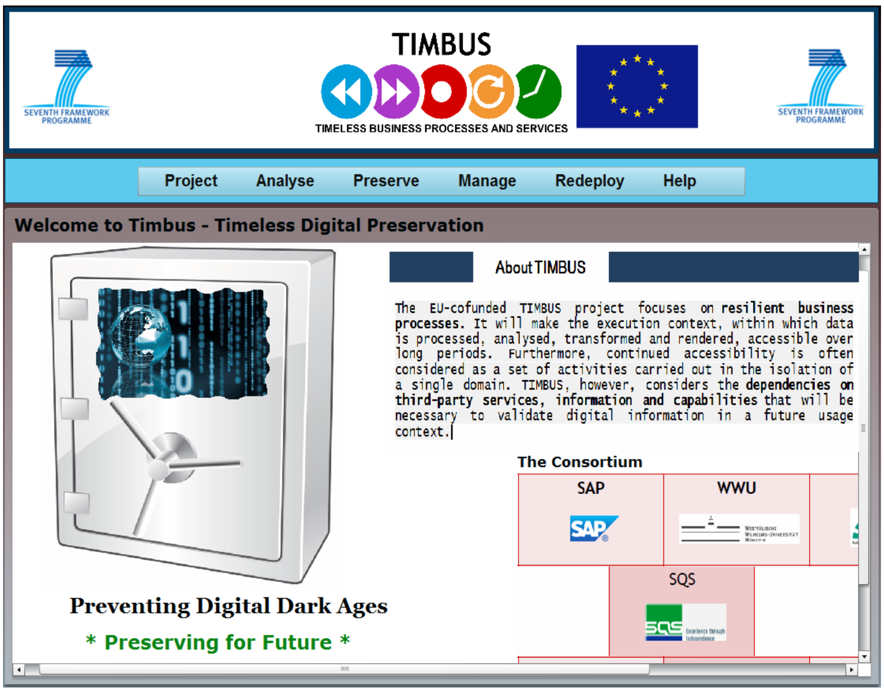

Figure 10-2: Home Page

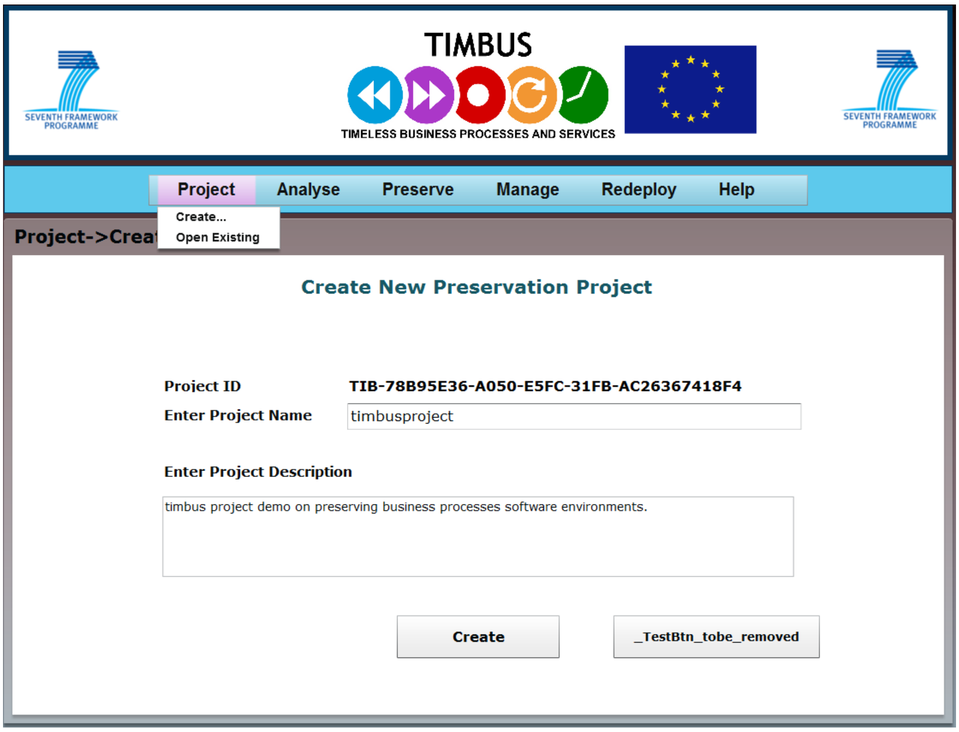

Figure 10-3: Create Project Screen

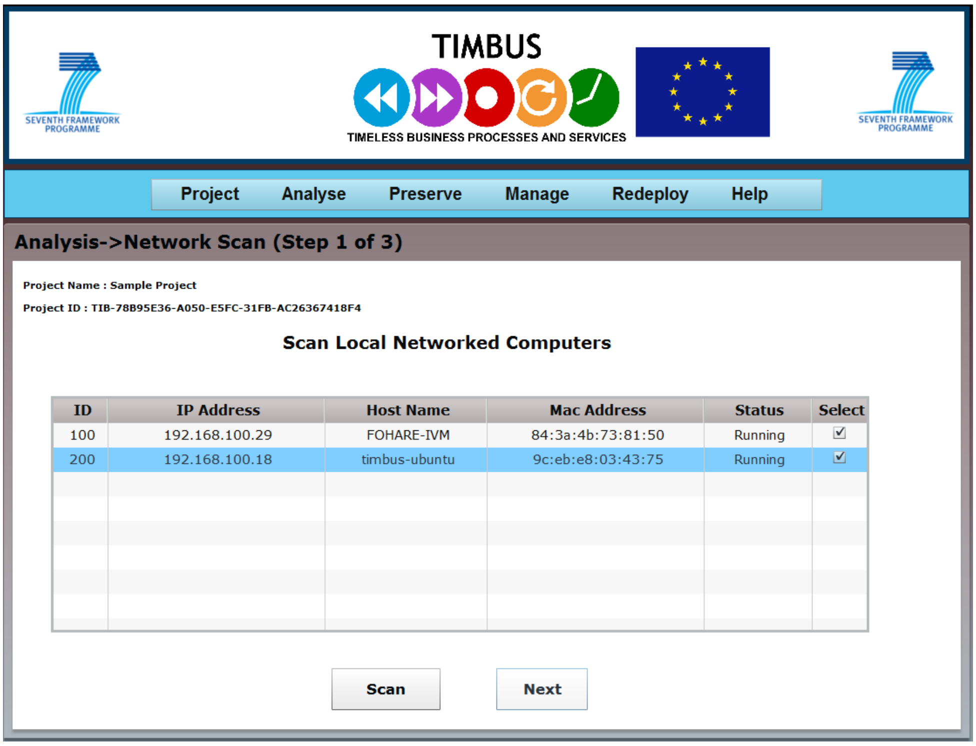

Figure 10-4: Network Scan

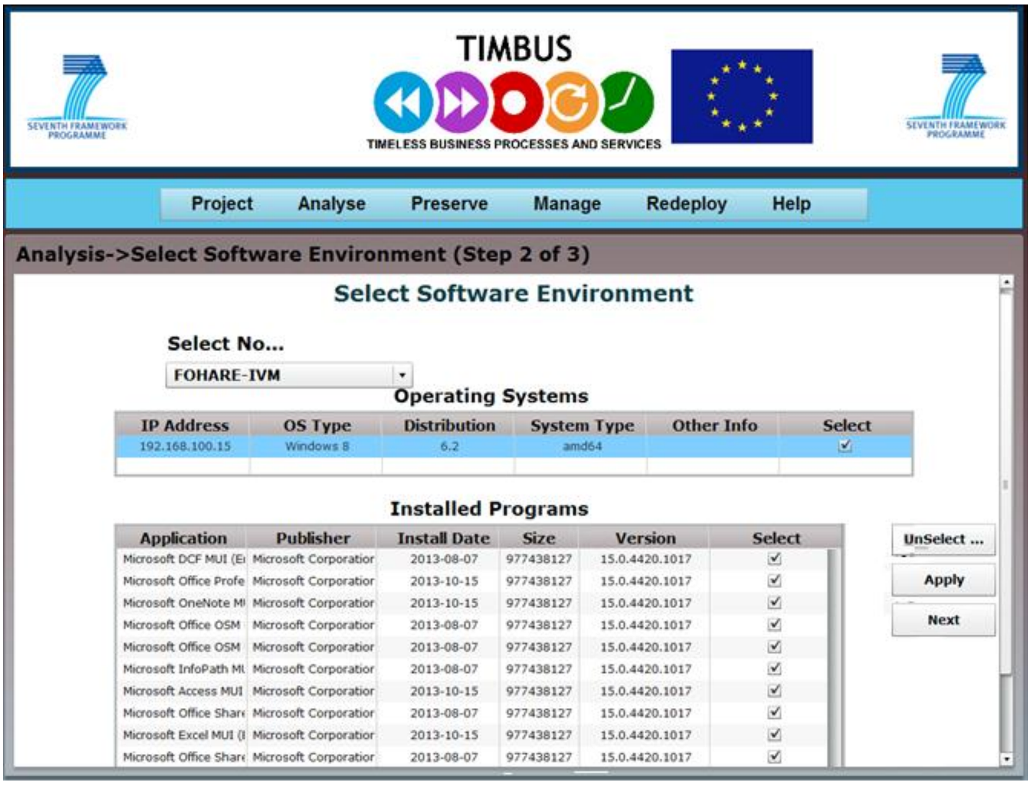

Figure 10-5: Software Environment Selection

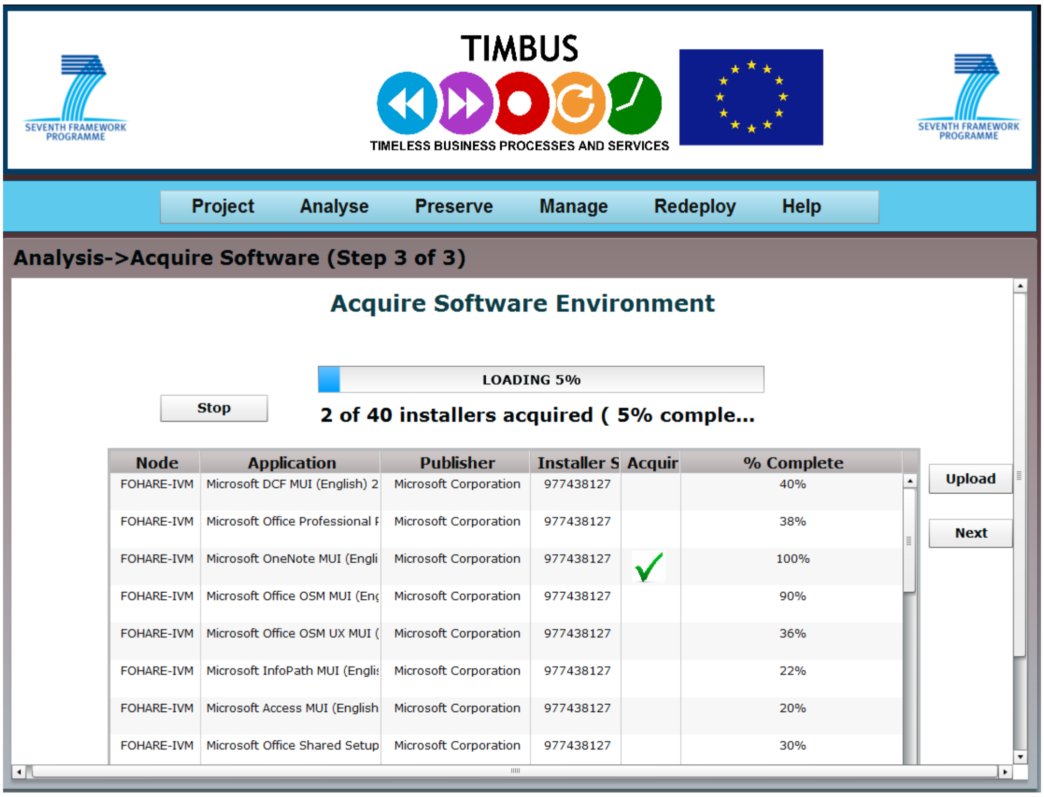

Figure 10-6: Extract Software Environment

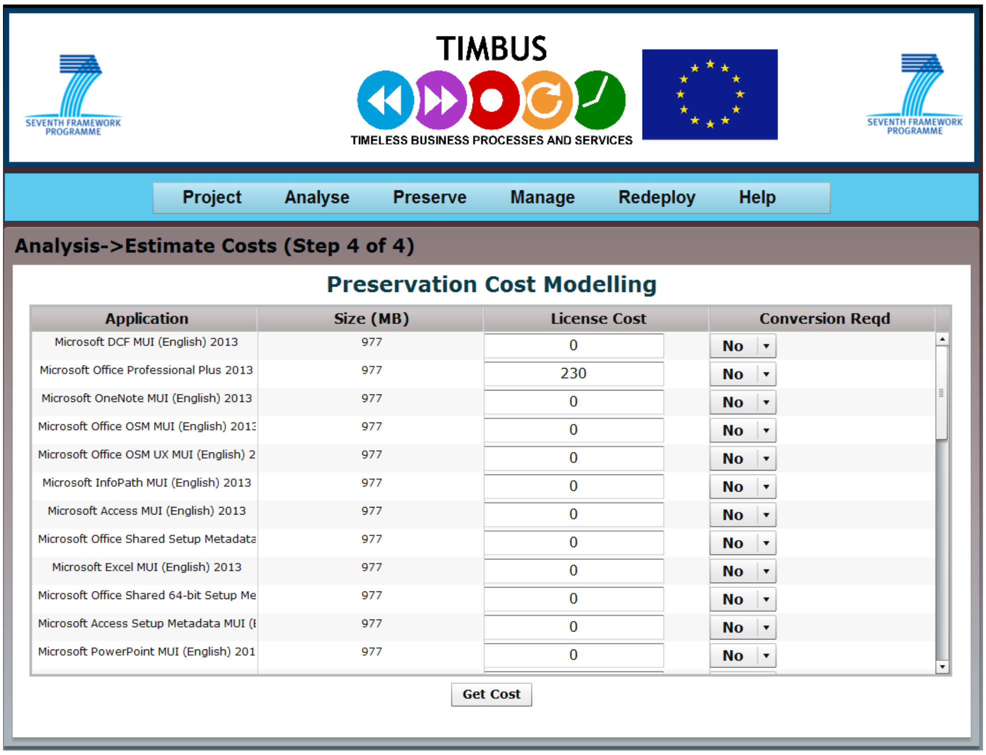

Figure 10-7: Cost Modelling Input

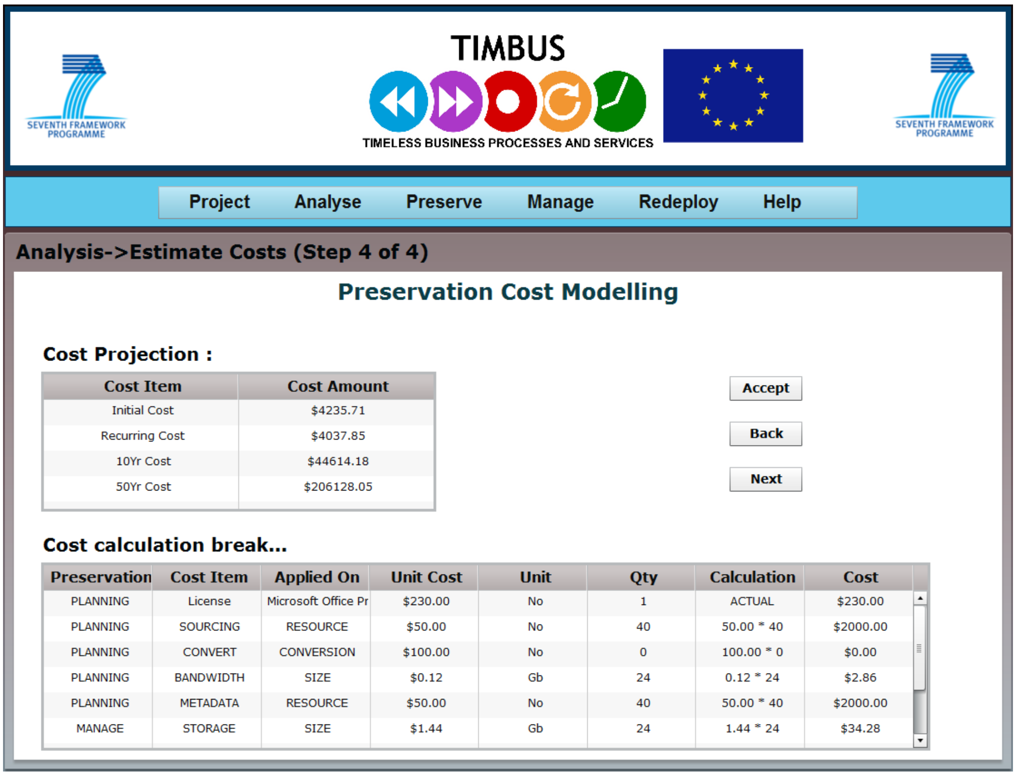

Figure 10-8: Cost Modelling Result

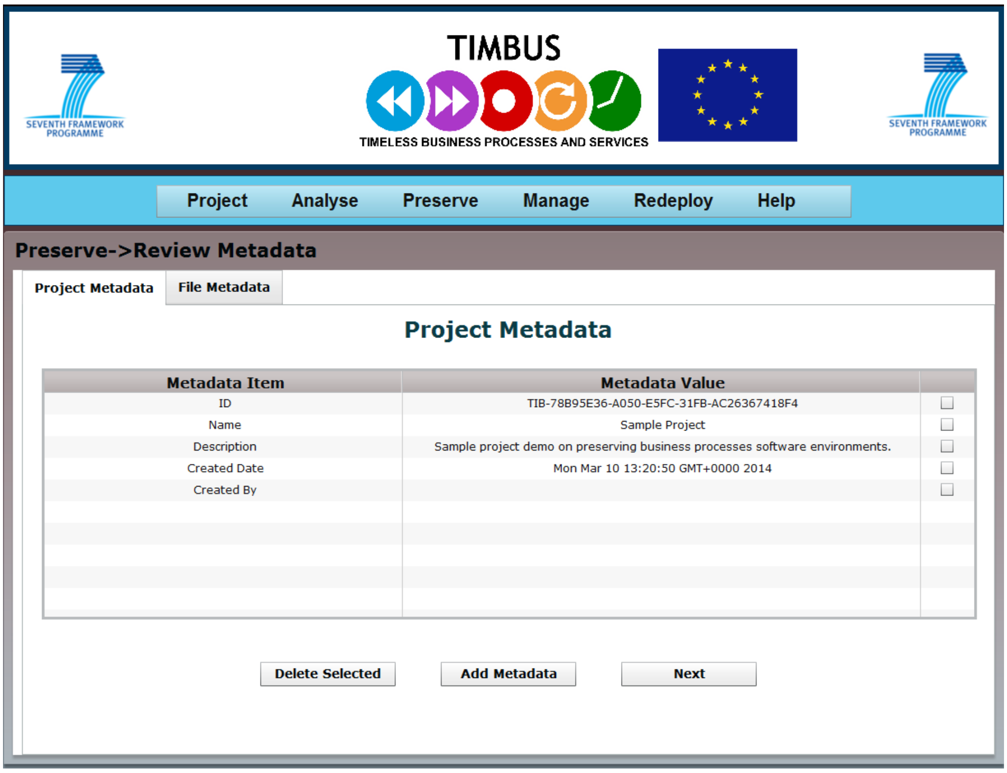

Figure 10-9: Edit Project Level Metadata

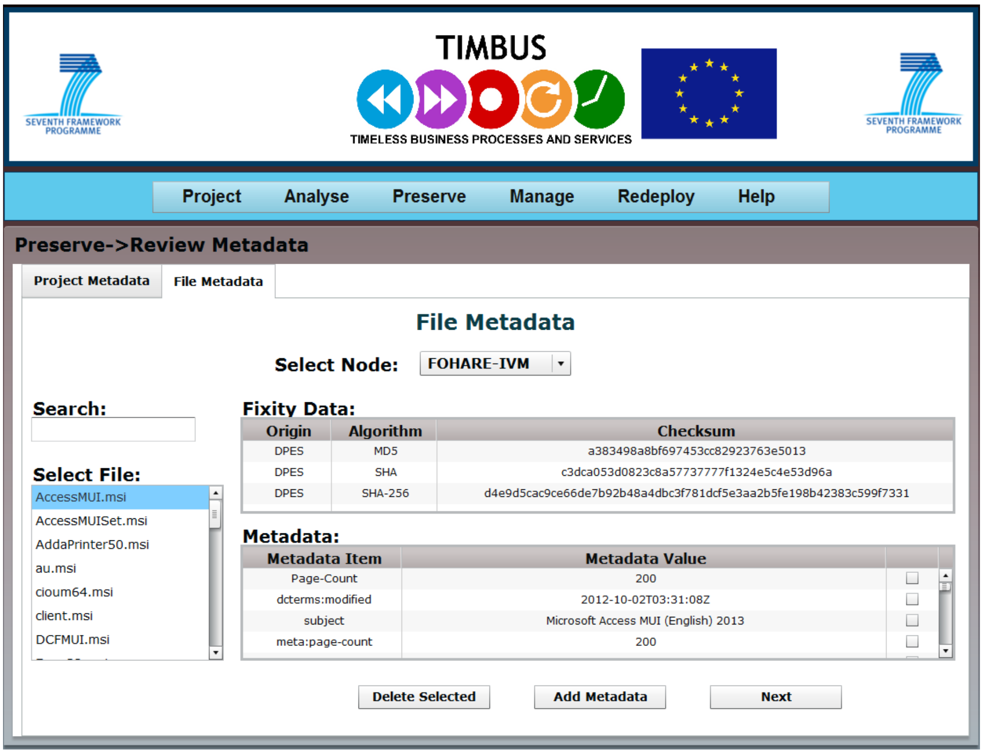

Figure 10-10: Edit Artefacts level Metadata

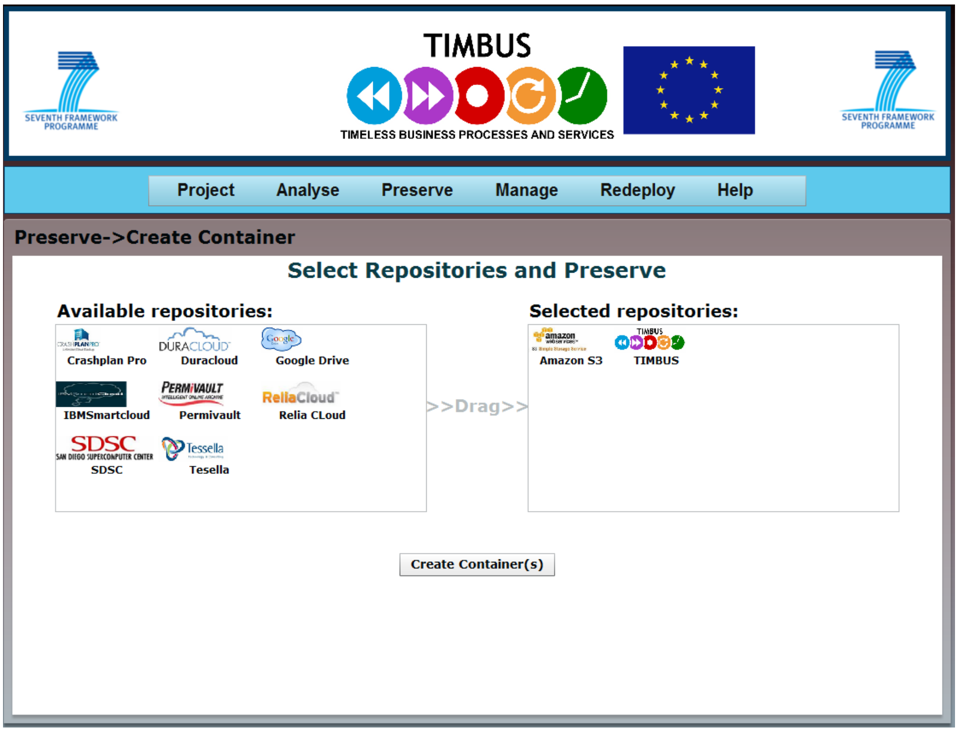

Figure 10-11: Storage Repository Selection

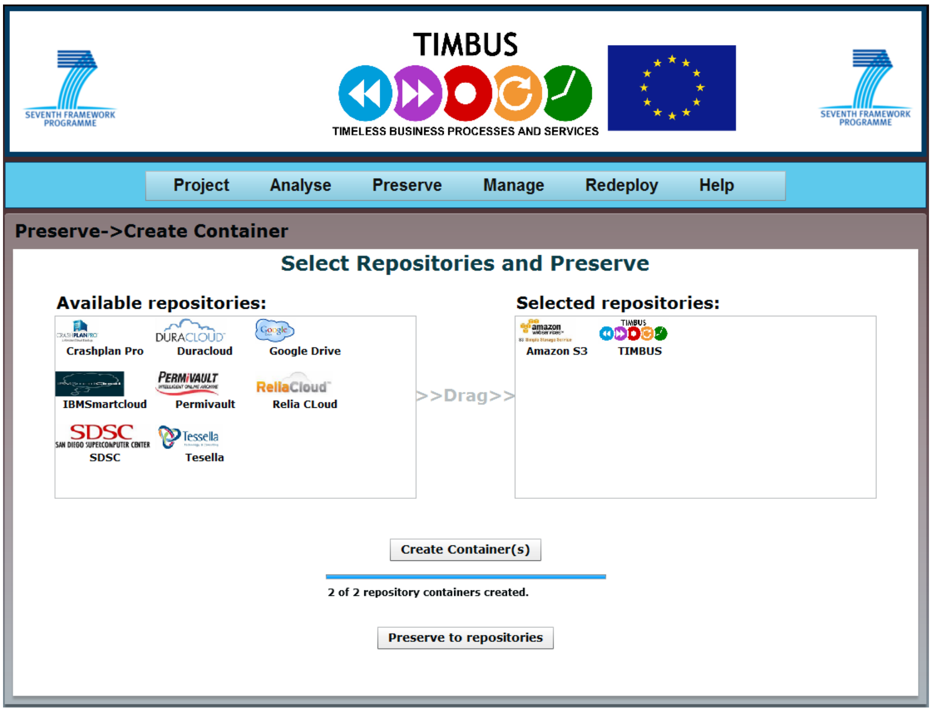

Figure 10-12: Storage Repository Containers creation

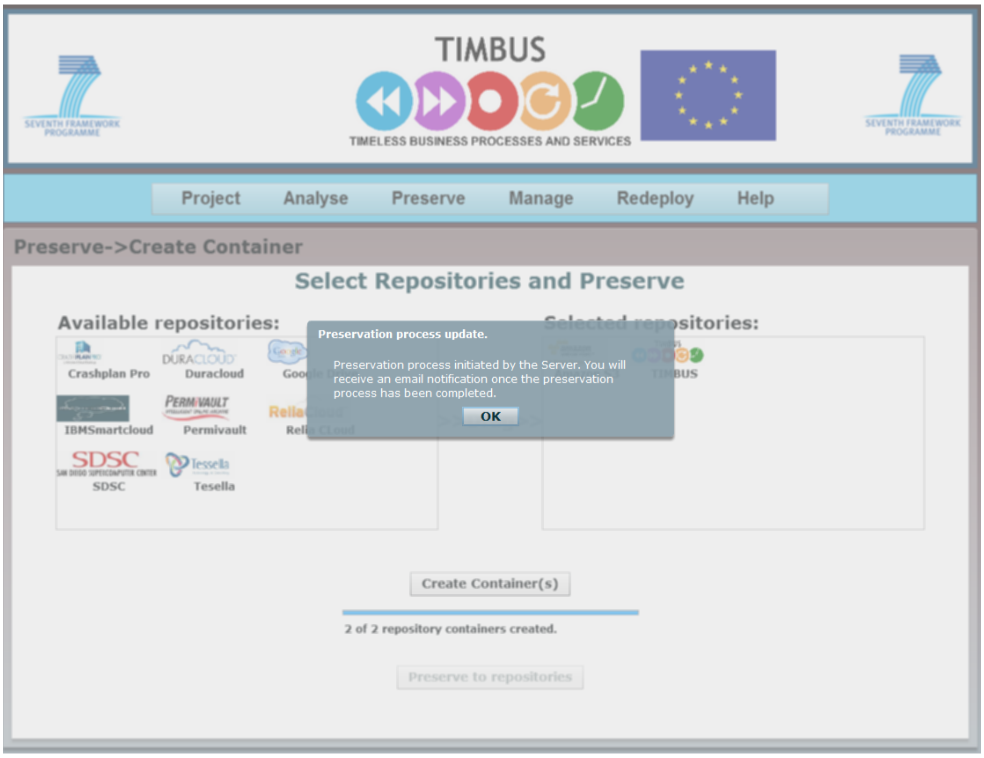

Figure 10-13: Preservation Process Initiated

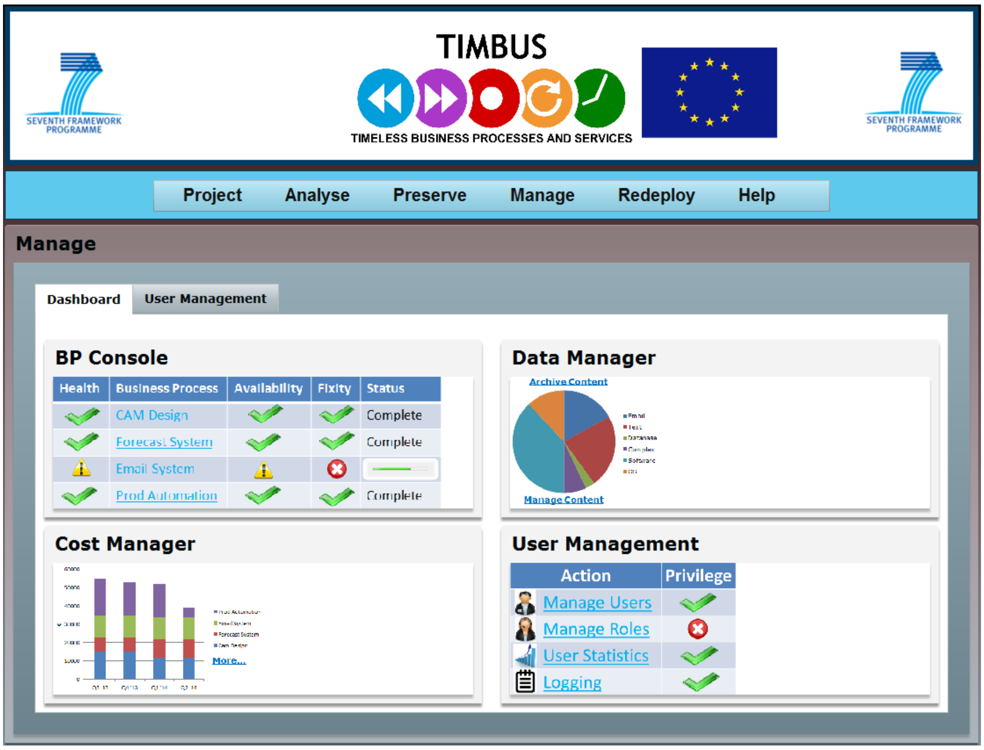

Figure 10-14: DPES Management Screen Concept

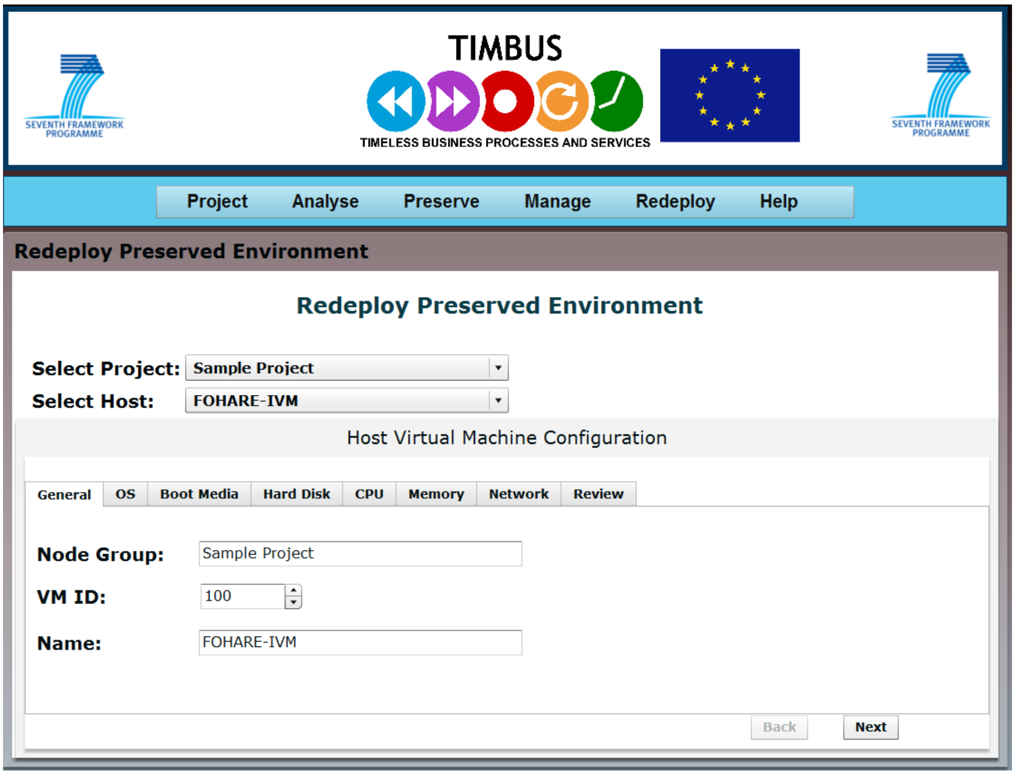

Figure 10-15: Redeployment – General Options

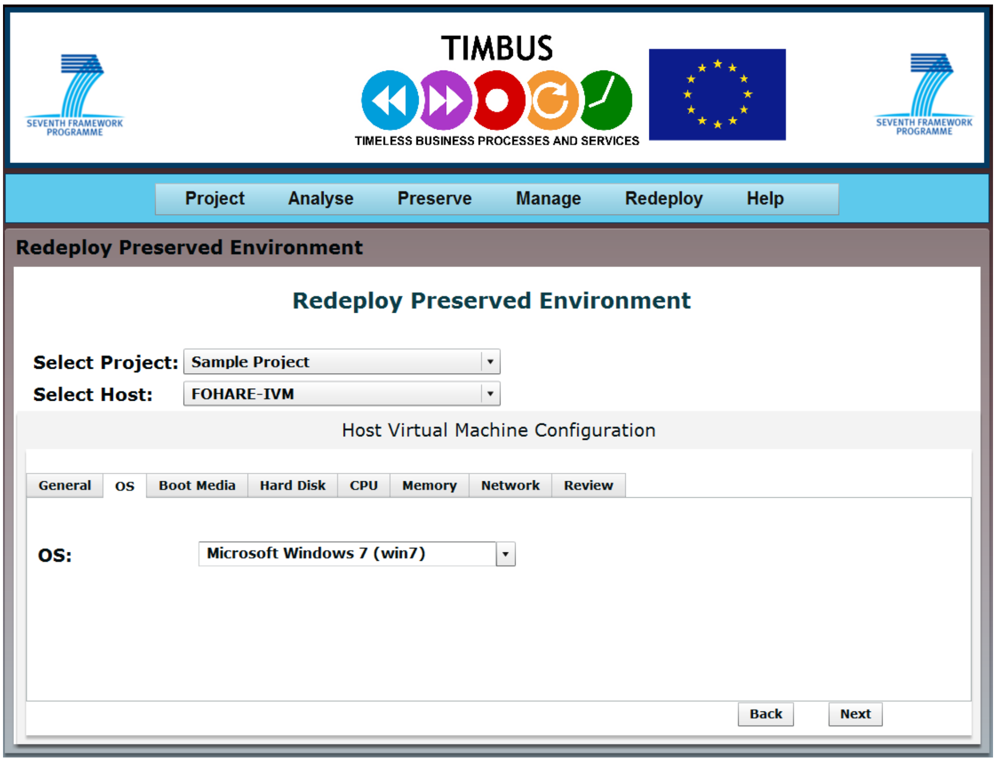

Figure 10-16: Redeployment OS Selection

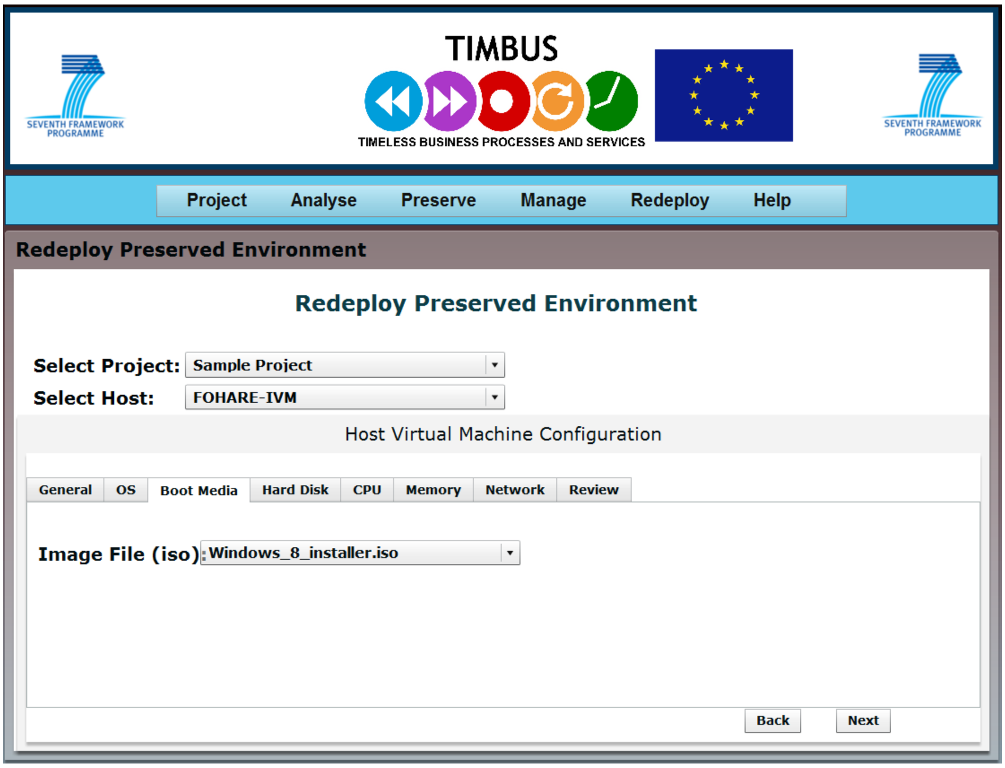

Figure 10-17: Redeployment Boot Options

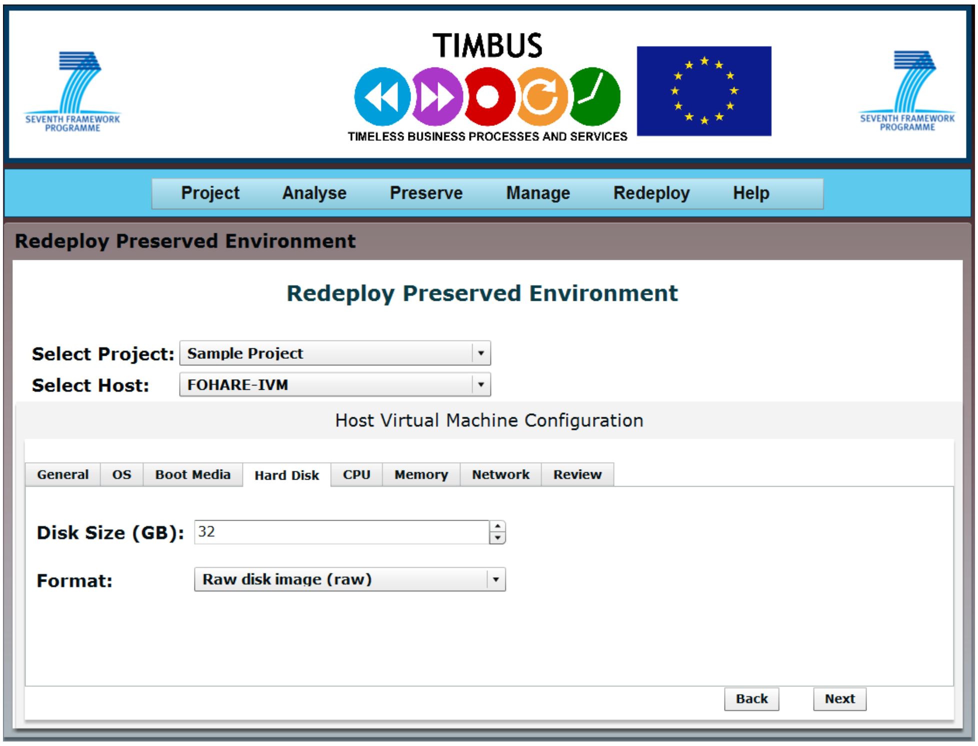

Figure 10-18: Redeployment – Disk Options

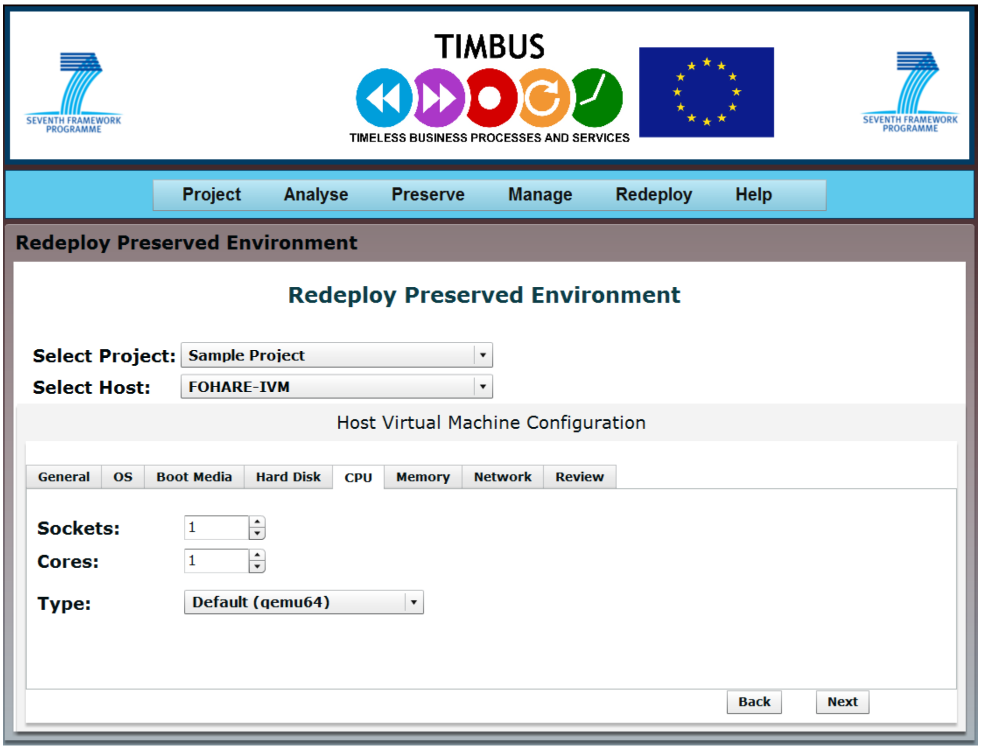

Figure 10-19: Redeployment CPU Options

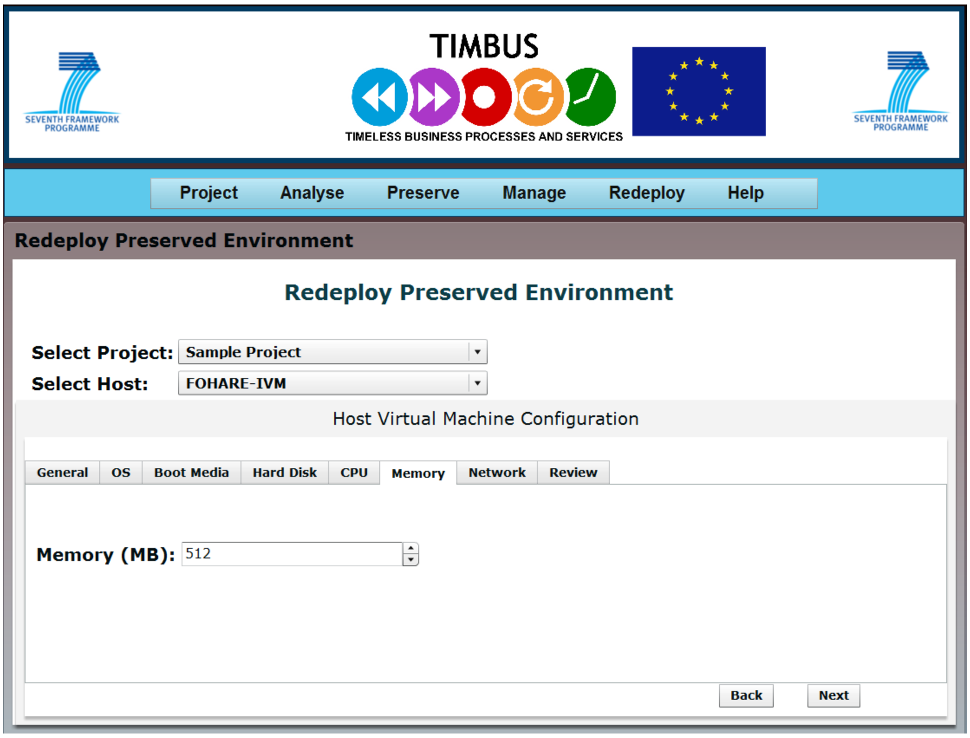

Figure 10-20: Redeployment – Memory Options

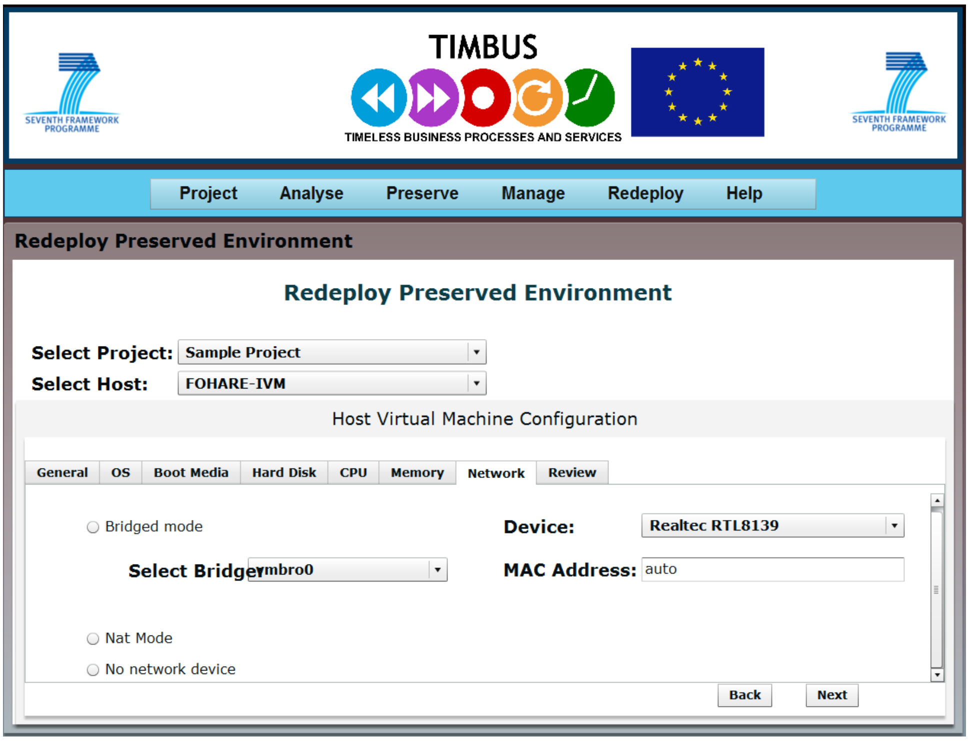

Figure 10-21: Redeployment – Network Options

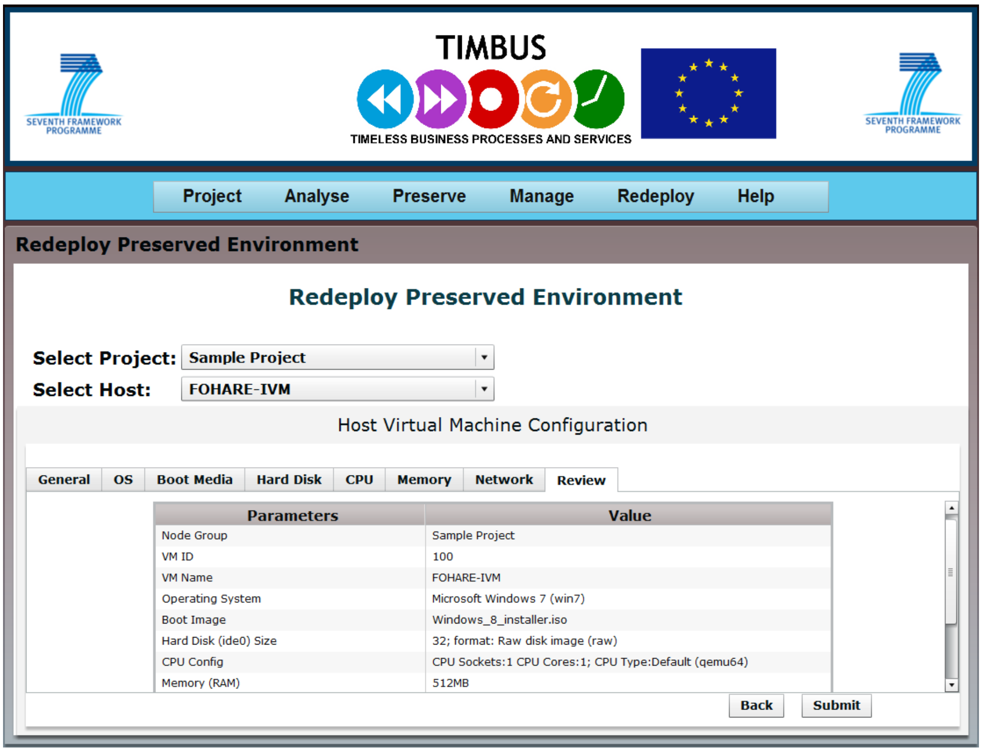

Figure 10-22: Redeployment Summary

10.2 IoT Use case

10.2.1 Introduction to the TIMBUS IoT Scenario

The amount of data that is being created every day is increasing exponentially. Figures on this have been included in the TIMBUS exploitation deliverables D2.1, D2.2 and D2.3. It is becoming more and more challenging to build systems/platforms that can process this huge volume of data and convert it into information. Big corporations that handle such huge volumes of data such as Google, Netflix etc., have built custom solutions optimised to their needs, some of which are now available as open source tools. These corporations had time on their side as they started early and have built their platforms as the needs evolved. Today, however, the picture is different. For someone make a start with big-data, they will have to hit the ground running since eco-system is such that data to be handled and processed from day one will be huge. These platforms present an interesting use case to evaluate the TIMBUS approach and this is one of the purposes of reporting this effort as part of this deliverable.

The TIMBUS research team in Intel has assisted in designing a platform we called KaZSM to solve many problems of processing big-data. The proof of concept prototype is actually deployed and receives data from a large number of sensors in the field (located across Dublin, London and Liverpool). It processes this data and makes the processed data available to consumers via a web based UI. Thus we have a very exciting opportunity to test out the TIMBUS tools on a live system with a potential for productisation further down the line as detailed in the Intel exploitation plan in deliverable D2.3.

10.2.2 KaZSM business services

Data Subscription: The primary purpose of KaZSM is to convert a huge volume of discrete sensor inputs into pieces of information that are useful and which can be persisted in a data store. Once the information is available in the data store, it needs to be made available to data consumers. This is done through a data hosting API which allows consumer systems to subscribe to data that is of interest to them.

Data Acquisition: On the other end of data acquisition, KaZSM provides the capability for devices to post sensor data through an MQTT [71] (Message Queue Telemetry Transport) based API. MQTT is a protocol that is built for machine-to-machine communication. This API lets KaZSM discover the capabilities of the devices and acquire the data.

Data Processing: This is the most important part of KaZSM and the one adds value to the data. KaZSM uses Storm to process the data. Storm allows multiple data flows and data processing steps using a series of Spouts and Bolts. This is illustrated in Figure 10-23. Through Storm in combination with a metadata layer, KaZSM provides a highly configurable and scalable data processing, which works as follows:

(1) Each spout provides data to one or more bolts.

(2) Each bolt processes data and

(3) makes it available to a spout or another bolt.

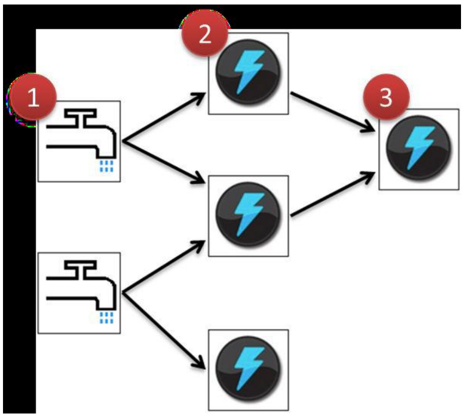

Figure 10-23: An Example of Storm Topology

11 References

[1] Wiki - Digital Dark Age - http://en.wikipedia.org/wiki/Digital_dark_age

[2] Amazon Web Services (AWS) - http://aws.amazon.com/

[3] TIMBUS Deliverable D6.4 Business Process Virtualisation – https://timbus.teco.edu/svn/timbus/Deliverables/FinalM30/D6.4/D64_M30_VirtualisationManager.pdf

[4] TIMBUS Deliverables D5.3 Architecture for VSRI – https://timbus.teco.edu/svn/timbus/Deliverables/FinalM24/D5.3/D5.3_M24_Architecture_for_VSRI.pdf

[5] Amazon Simple Storage service (S3) - http://aws.amazon.com/s3/

[6] Dropbox cloud data Storage - https://www.dropbox.com/

[7] Block level Storage vs. File level Storage – http://www.stonefly.com/resources/File-Level-Storage-vs-Block-Level-Storage.asp

[8] Object Storage - http://en.wikipedia.org/wiki/Object_storage#cite_note-1

[9] Mesnier, Mike; Gregory R. Ganger, Erik Riedel (August 2003). "Object-Based Storage". *IEEE Communications Magazine*: 84–90. http://www.storagevisions.com/White%20Papers/MesnierIEEE03.pdf

[10] Object Storage Architecture – http://en.wikipedia.org/wiki/File:High_level_object_storage_architecture.png

[11] SNIA's Comparison of Storage Standard – http://snia.org/sites/default/files2/SDC2013/presentations/Cloud/ScottHoran_Lessons_Learned_Implementing.pdf

[12] Measurement & Analysis of Large - Scale Network File System Workloads, University of California Santa Cruz, 2008

[13] Object Storage A fresh Approach to Long- Term File Storage, A Dell Technical white paper – http://partnerdirect.dell.com/sites/channel/en-ca/documents/object_storage_overview_whitepaper.pdf

[14] File Level Storage - http://www.stonefly.com/resources/File-Level-Storage-vs-Block-Level-Storage.asp

[15] Block, File and Object Storage Comparison – https://developers.seagate.com/display/KV/Block,+File+and+Object+Storage

[16] Wiki- Cloud storage - http://en.wikipedia.org/wiki/Cloud_storage

[17] ENSURE EU FP7 Project - http://ensure-fp7-plone.fe.up.pt/site/

[18] Vernik, Gil, et al. "Data On-boarding in Federated Storage Clouds." Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing. IEEE Computer Society, 2013.

[19] Kemme, Bettina, et al. "Consistency in Distributed Systems (Dagstuhl Seminar 13081)." (2013).

[20] "4 reasons why cloud and on-premises storage are different, but equally good for people data". 2013-09-09. Retrieved 2013-09-09.

[21] O'Brien, J. A. & Marakas, G. M. (2011), Computer Software and Management Information Systems 10th ed. 145. McGraw-Hill/Irwin

[22] PDS Cloud: Long Term Digital Preservation in the cloud – https://www.research.ibm.com/haifa/projects/storage/datastores/papers/PdsCloudPaper_CameraReady.pdf

[23] Digital data archive and Preservation in the cloud – http://www.snia.org/sites/default/education/tutorials/2013/spring/cloud/ChadThibodeau-SebastianZangaro_Digital_Data_Archive_and_Preservation.pdf

[24] File Storage Costs Less in the Cloud than In-House, Andrew Reichman, Forrester 2011

[25] Object storage gains steam as unstructured data grows," Beth Pariseau, Storage Magazine, November / December 2009

[26] IDC White Paper sponsored by EMC, As the Economy Contracts, the Digital Universe Expands, May 2009

[27] SNIA's Cloud Best Practices – http://www.theregister.co.uk/2011/11/01/snia_cloud_archive_best_practices/

[28] IT Virtualization definition - http://www.gartner.com/it-glossary/virtualization/

[29] O'Brien, J. A. (2002) *Management Information Systems: Managing Information Technology in the E-Business Enterprise*, 5th ed. McGraw-Hill, New York.

[30] Book By: Sangeeta Dhamdhere, Title: Cloud Computing and Virtualization Technologies in Libraries, ISBM-978-1-4666-4631-2

[31] Wiki – Hypervisor http://en.wikipedia.org/wiki/Hypervisor

[32] Ivor Milosevic, Hrvoje Stancic (2011) – Usage of Virtualization Technologies in Long-Term Preservation of Integrity and Accessibility of Digital Data.

[33] Book by Martha Young and Michael Jude, Title: Case for Virtual Business Processes, ISBN-10: 1-58720-087-2

[34] Wiki – Bare machine - http://en.wikipedia.org/wiki/Bare_machine

[35] Hardware assisted Virtualization Image source - http://forums.techarena.in/guides-tutorials/1104460.htm

[36] Hypercall - http://wiki.xen.org/wiki/Hypercall

[37] Para-Virtualisation - http://searchservervirtualization.techtarget.com/definition/paravirtualization

[38] Para-Virtualisation - http://www.virtualizationsoftwares.com/what-is-paravirtualization/

[39] Chroot on Unix Operating systems - http://en.wikipedia.org/wiki/Chroot

[40] Dirk von Suchodoletz, Albert-Ludwigs University Freiburg, A Future Emulation and Automation Research Agenda, http://eprints.rclis.org/15239/1/future-emu-dp.pdf

[41] Dirk von Suchodoletz, et al. Albert-Ludwigs University Freiburg, QEMU – A Crucial Building Block in Digital Preservation Strategies, http://eprints.rclis.org/15406/1/qemu-in-ltp.pdf

[42] QEMU Emulator - http://wiki.qemu.org/Main_Page

[43] Jeffrey van der Hoeven, Hilde van Wijngaarden, - Modular emulation as a long-term preservation strategy for digital objects, Koninklijke Bibliotheek, the National Library of the Netherlands The Hague, The Netherlands - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.106.2100&rep=rep1&type=pdf

[44] Holdsworth, D., Wheatley, P.R.: Emulation, Preservation and Abstraction. In: RLG Diginews, vol. 5 (4), (2001).

[45] Software Sustainability Institute - http://www.software.ac.uk/resources/approaches-software-sustainability

[46] Rothenberg, J.: An Experiment in Using Emulation to Preserve Digital Publications. Koninklijke Bibliotheek, The Hague, The Netherlands, 2000. Available at: http://www.kb.nl/coop/nedlib/results/emulationprese rvationreport.pdf (accessed 8 September 2005)

[47] Wiki – UVC-Based Preservation - https://en.wikipedia.org/wiki/UVC-based_preservation

[48] Nigamanth Sridhar, Jason O. Hallstorm (2006), A Behavioral Model for Software Containers. Springer-Verlag Berlin Heidelberg 2006 - http://selab.csuohio.edu/~nsridhar/research/Papers/PDF/abfms.pdf

[49] Software Containers Comparison - http://en.wikipedia.org/wiki/Operating_system-level_virtualization

[50] Yves Kemp - Use of Virtualization Techniques for data preservation and long term analysis.

[51] TIMBUS deliverables D8.1 – Use case Definition and Digital Preservation Requirements – http://timbusproject.net/component/docman/doc_download/138-d81m12usecasedefinitionanddigitalpreservationrequirementspdf

[52] TIMBUS deliverables D9.1 – Use case definition and Digital Preservation Requirements – http://timbusproject.net/component/docman/doc_download/137-d91m12usecasedefinitionanddigitalpreservationrequirementspdf

[53] Wiki – Internet of things - http://en.wikipedia.org/wiki/Internet_of_Things

[54] Michael Factor, Ealan Hennis, Et. al, Authenticity and Provenance in Long Term Digital Preservation: Modeling and Implementation in Preservation Aware Storage, IBM Research Lab in Haifa, Israel and

University of Urbino, Italy. https://www.usenix.org/legacy/event/tapp09/tech/full_papers/factor/factor.pdf

[55] Charles B. Weinstock, et, al. (2006), On System Scalability, Performance Critical System, from http://www.sei.cmu.edu/reports/06tn012.pdf

[56] Wheatley, P. and Hole, B. (2009). LIFE3: *Predicting long-term digital preservation costs, from* http://www.life.ac.uk/3/docs/ipres2009v24.pdf

[57] Apache java library for metadata - http://tika.apache.org/

[58] FITS Tools - https://code.google.com/p/fits/wiki/tools

[59] JHOVE2 – metadata extraction tools - https://bitbucket.org/jhove2/main/wiki/Home

[60] File metadata population - http://blog.archive.org/2011/03/31/how-archive-org-items-are-structured/

[61] AMAZON SDK for Java - http://aws.amazon.com/sdkforjava/

[62] OpenVZ linux container - http://openvz.org/Main_Page

[63] Docker a linux container - https://www.docker.io/

[64] Proxmox Virtual Environment - https://www.proxmox.com/proxmox-ve

[65] Object Storage openstack swift - https://wiki.openstack.org/wiki/Swift

[66] Kafka 0.7.2 - http://kafka.apache.org/documentation.html#introduction

[67] Zookeeper 3.3.5 - http://zookeeper.apache.org/doc/r3.3.5/releasenotes.html

[68] Storm 0.9.2-rc2 - http://storm.incubator.apache.org/

[69] MQTT - http://activemq.apache.org/mqtt.html

[70] Intel Quark - http://en.wikipedia.org/wiki/Intel_Quark

[71] Hadoop - http://hadoop.apache.org/

[72] MQTT Protocol Specification - http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html

[73] KEEP - Keeping Emulator Environment Portable - http://www.keep-project.eu/ezpub2/index.php