www.forgetit-project.eu

ForgetIT

Concise Preservation by Combining Managed Forgetting
and Contextualized Remembering

Grant Agreement No. 600826

Deliverable D8.2

Work-package	WP8: The Preserve-or-Forget Reference Model and Framework
Deliverable	D8.2: The Preserve-or-Forget Reference Model Initial Model
Deliverable Leader	F. Gallo (EURIX)
Quality Assessor	B. Logoglu (TT)
Dissemination level	PU
Delivery date in Annex I	M15 (April 2014)
Actual delivery date	2015-01-30
Revisions	12
Status	First Release
Keywords:	Preserve-or-Forget Reference Model, OAIS

Disclaimer

Revision History

Version	Major changes	Authors
v0.01	Created ToC, assigned contributions	EURIX
v0.02	Added preliminary text for introduction, foundations of reference model, functional and information model	L3S, LTU, EURIX
v0.03	Completed foundations of reference model, added mapping to the architecture	LUH, EURIX
v0.04	Created diagrams and added description of functional entities	LUH, LTU
v0.05	Added mapping to OAIS, updated other Sections	LTU, EURIX
v0.06	Completed description of model layers and workflows	LUH, LTU, EURIX
v0.07	Added section about extensions of information management systems	LUH
v0.08	First complete version for comments from all partners	EURIX
v0.09	Implemented comments from all partners, version ready for internal QA	EURIX
v0.10	Implemented comments from internal QA	EURIX
v0.11	Implemented final comments from all partners, first release version	EURIX
v0.12	Final checks and clean-up, first release version for submission	LUH, EURIX

List of Authors

Partner Acronym	Authors
LUH	Claudia Niederee
LTU	Ingemar Andersson, Jörgen Nilsson
IBM	Doron Chen
DFKI	Heiko Maus
USFD	Mark Greenwood
UEDIN	Robert Logie
EURIX	Francesco Gallo

Table of Contents

Executive Summary

This document describes the initial version of the Preserve-or-Forget (PoF) Reference Model. The Reference Model aims to encapsulate the core ideas of the ForgetIT approach into a a re-usable model. Therefore, it is inspired by the core principles of the Preserve-or-Forget approach: synergetic preservation, managed forgetting and contextualized remembering.

Based on this, we have identified five characteristics for a PoF Reference Model, which considers Active System and Digital Preservation System (DPS) as a joint ecosystem: *integrative*, *value-driven*, *brain-inspired*, *forgetful* and *evolution-aware*. Those characteristics have driven the design of the functional part of the PoF Reference Model and the associated Information Model. The Functional Model is made up of three layer: *Core Layer*, *Remember & Forget Layer* and *Evolution Layer*. For each layer we describe the main functional entities and the representative workflows.

This deliverable is not limited to the PoF Reference Model. It also presents mappings and interactions of the model with other building blocks of an intelligent DPS. Firstly, the PoF Framework developed in WP8 is compliant to the model. Actually its design was inspired by the experiences gathered in the creation of the framework. For clarifying the relationship, this deliverable contains a mapping of the PoF Reference Model functional entities to the relevant architecture components. Furthermore, the relationship with OAIS is also discussed: with its focus on synergetic preservation, the PoF Reference Model complements and goes beyond the OAIS Reference Model by stressing the aspects bringing the Active System and the DPS closer together. Finally, it also discusses the extensions required in any Active System for following the approach outlined by the PoF Reference Model.

The model presented in this deliverable is based on the work done for the PoF Middleware in WP8, by the concepts developed in WP3-WP7 as well as by the foundational work in WP2 and WP5. Furthermore, the PoF Reference Model builds upon the lessons learned so far in the ForgetIT project - including the experiences from the work with the interdisciplinary partners and the application partners.

The PoF Reference Model serves as conceptual guideline for further integration process: the current version is still considered preliminary and will be further developed in the remaining time of the project. A final version of the PoF Reference Model will be reported in deliverable D8.5.

1 Introduction

In this deliverable we present a model for the novel approach to intelligent preservation management taken by the ForgetIT project. Since ForgetIT stresses the smooth interaction between information management and preservation management, the Preserve-or-Forget (PoF) Reference Model described here pays special attention to the functionality, which bridges between the Active System and the Digital Preservation System (DPS) (see deliverable D8.1[ForgetIT, 2014c]). This includes the selection of content for preservation, the transfer of content between the systems, contextualization for easing long-term interpretation, processing during preservation time in reaction to changes, as well as access to the joint information space populated both by preserved content and content in active use.

The PoF Reference Model presented in this deliverable is based on the work done for the PoF Middleware in WP8, the components and especially the concepts developed in WP3-WP7 as well as the foundational work in WP2 and WP5. Furthermore, the model builds upon the lessons learned so far in the ForgetIT project - including the experiences from the work with the interdisciplinary partners and with the application partners. The PoF Reference Model aims to encapsulate the core ideas of the ForgetIT approach into a a re-usable model.

The PoF Reference Model serves as conceptual guideline for further integration process and will be further refined during the project. The refined and extended version of the model will be presented in deliverable D8.5 [ForgetIT, 2016].

1.1 Purpose of the PoF Reference Model

It is the idea of the ForgetIT project to follow a forgetful, focused approach to digital preservation, which is inspired by human forgetting and remembering. Its goal is to ease the adoption of preservation technology especially in the personal and organizational context and to ensure that important content is kept safe, useful, and understandable on the long run. For this purpose, concepts, technologies, and an entire framework (the PoF Framework) have been developed in the ForgetIT project so far. Furthermore, the framework has been implemented in a first version in the form of the PoF Middleware (see deliverable D8.3 [ForgetIT, 2014d]).

While the architecture and the implemented PoF Middleware ease adoption on the technology level, the PoF Reference Model leverages the ideas developed so far in the ForgetIT project to a conceptual level: required conceptual functionalities and relevant concepts are collected in a systematic way and are related to each other.

Similar to the OAIS Reference Model [CCSDS, 2012], the PoF Reference Model defines the terminology and concepts for the approach, which can be used as a basis for the implementation of the ForgetIT approach as well as for the further discussion and development of a forgetful approach to preservation.

The PoF Reference Model is also intended as the basis for the implementation of the ForgetIT approach in other ways than the one given by the PoF Middleware and in the context of other types of Active System and/or DPS, as they are considered for the PoF Middleware implementation. For fostering this aspect we also stress the idea of considering information management and preservation management as a *joint ecosystem* for the PoF Reference Model.

It is worth noting that the ForgetIT model does not intend to replace the OAIS model. As mentioned above, it has a different focus than the OAIS model and can interact with OAIS compatible approaches as it is also outlined in Section 6 of this deliverable.

The PoF Reference Model especially targets personal preservation settings as well as organizational preservation settings, which are not covered by legal regulations such as deposit laws. The focus is not on supporting memory institutions, although parts of the ForgetIT ideas (e.g., contextualization) might be applicable in this area as well. The selected settings (personal and non-mandatory organizational preservation) can especially benefit from the ForgetIT approach, since a) there is a big gap to preservation adoption in those fields, b) there are explicit preservation choices to be made, and c) there is a need for automation of the processes, in order to reduce the amount of investment required for preservation.

For the representation of the preliminary model described here we provide simple graphical representations as starting point for deeper analysis. If necessary, a more formal approach will be used for the final version of the model, using UML notation and machine-processable formats (e.g. ontology or XML schema specification), supporting the implementation of the PoF Framework. Such information will be reported in deliverable D8.5 [ForgetIT, 2016].

1.2 Target Audience

The target audience of this deliverable are practitioners and developers in the area of content and information management, who consider to get started with content preservation. The deliverable is also targeting practitioners, developers, and researchers in the area of preservation management.

In order to serve this diverse audience, the deliverable considers the presented PoF Reference Model from different perspectives. In addition to the presentation of the model itself in terms of functional model and information model, the deliverable also presents a mapping to the OAIS model and the interaction of the PoF Reference Model with a content management system. Furthermore, the presented mapping to the PoF architecture is also meant to ease adoption for system developers.

1.3 Structure of the Deliverable

The rest of this deliverable is structured as described in the following. In Section 2, we discuss the foundations of the PoF Reference Model, covering the characteristics of the model as well as the requirements, which did influence its development. In Section 3, we present the PoF Reference Model for an integrated information and preservation management system, which considers this combination as a joint ecosystem. In more detail, we describe a functional model and an information model. In Section 4, we relate the PoF Reference Model to the reference architecture of the PoF Framework. This architecture, which is based on the system architecture presented in deliverables D8.1 [ForgetIT, 2014c] and D8.3[ForgetIT, 2014d], maps the joint ecosystem of building blocks from the reference model on an architecture with three main parts: the Active System, the Digital Preservation System (DPS) and the PoF Middleware. In the second part of the deliverable, we relate the PoF Reference Model to existing models and approaches, in order to ease adoption. This includes a discussion of the extensions required in an information system for becoming part of an information and preservation management ecosystem (Section 5) and a discussion on how to map the relevant parts of the model on an OAIS based DPS as well as suggested extensions to such a system (Section 6). We conclude the deliverable with a summary of the main insights, with ideas for future work and an assessment of the results compared to success indicators reported in the project proposal.

2 Foundations of the PoF Reference Model

Five main characteristics have been identified for the proposed PoF Reference Model, supporting the ForgetIT approach for a sustainable and smooth transition between information and preservation management. The model is:

1. **integrative** - bridging the gap between information and preservation management for easing the adoption of preservation technologies;

2. **value-driven** - acting upon short-term and long-term information value based on careful multi-faceted information value assessment;

3. **brain-inspired** - learning from the way humans forget and remember for a better more focused management of digital memories;

4. **forgetful** - using the idea of forgetting in the digital memory for staying focused and supporting preservation decisions;

5. **evolution-aware** - embracing the long-term perspective by dealing with change on various levels.

Those partially related characteristics and their facets as well as the requirements and ideas, which inspired those characteristics, are discussed in more details below. They provide the foundations for the PoF Reference Model discussed in Section 3.

2.1 Integrative: Bridging the Gap between Information and Preservation Management

It is the aim of the ForgetIT approach to preservation to create a smooth transition between active information use and preservation of information, which so far are some quite separate worlds. For this purpose, the ForgetIT Reference Model should be *integrative*, bringing the Active System and the DPS closer together. Further details about foundations of synergetic preservation can be found in deliverable D5.1 [ForgetIT, 2013b] and following WP5 deliverables.

However, due to the inherent long-term perspective of preservation-related solutions it is not the aim to build a strongly integrated, monolithic system. In the long run, it has to be foreseen that the Active System used as well as the employed DPS will change. Therefore, the idea is a flexible integration, which enables smooth bi-directional transition of information between the Active System and the DPS and, at the same time, is also prepared for major changes in the overall environment. [Afrasiabi Rad et al., 2014]

A core part of integration is to enable the smooth transition of content to be preserved into the DPS and the senseful reactivation of content back into the Active System after a -possibly very long - period into the DPS. An integrative solution should also embrace the idea of a *joint information space*, where the information in the DPS stays conceptually accessible, e.g., visible in search results, even if the content is only available in the DPS.

One part of achieving a smooth transition is to act as a pre-ingest system (or pre-access) and prepare information packages for delivery in either way. With highly automated procedures for preparation of these packages according to agreements or requirements, the quality of the information packages becomes more consistent, which alleviate the burden of information package handling on both sides. [ForgetIT, 2014b]

In addition, an integrative system should also support the decisions on what to preserve for easing the integration of preservation into the information management workflows, finally aiming for an integrated information and preservation management workflow. The automated selection process (appraisal) will be discussed in more detail in the context of managed forgetting (see Section 3.1).

2.2 Value-driven: Acting upon Short-term and Long-term Information Value

One of the core ideas of the ForgetIT Reference Model is to deviate from the general *keep it all* model, which makes the implicit assumption that all information has the same value with respect to being kept. In general the value of information is multifaceted and can be considered from different perspectives, e.g., the short-term value for current activities vs. long-term value of a resource for an organization. For the combined information and preservation management system, short-term value and long-term value of information has to be considered separately, since it is driven by different factors. For a review of short-term and long-term memory and forgetting see deliverables D2.2 [ForgetIT, 2014a] and D2.3 [ForgetIT, 2014a].

Short-term value The short-term value refers to the value of content for the current focus of activity, e.g., documents used for a task at hand are of high short-term value. In short-term value of information, we will see a high dynamics in the information value (due to changing interests and tasks) and a high influence on interaction-based evidences on the information value. In terms of the human brain, this is roughly comparable to the working memory (see Section 2.3), although human working memory has an even higher change frequency. Further details about foundations of managed forgetting can also be found in deliverable D3.1 [ForgetIT, 2013a] and following WP3 deliverables.

Identifying the short-term value of a document is of high interest for creating immediate benefit in information management, e.g. by de-cluttering the desktop, which is one of the goals of the ForgetIT project. The idea here is to give higher priority or visibility to resources with high short-term value. In the ForgetIT project the term *Memory Buoyancy* has been coined for this purpose. It is inspired by the idea of resources of decreasing importance to the user are sinking away from the user.

Long-term value The long-term value of a resource is obviously relevant in the context of preservation. It refers to the value a resource has on the long run. Such long-term value can be used to decide about the investment to be made into preservation for the respective resource [Andersson et al., 2014]. Long-term value is expected to be more difficult to compute, since it includes estimating future use of resources [Rauch et al., 2013]. Furthermore, long-term value is driven by (at least partially) different factors than short-term value. It is expected that more objective values such as diversity, coverage and quality will play a more important role here (see deliverable D9.3 [ForgetIT, 2014e] for further details about the dimensions used for resource value). In the ForgetIT project, we have coined the term *Preservation Value* for the long-term value of a resource.

2.3 Brain-inspired: What can we Learn from Human Forgetting and Remembering?

Humans are very effective in forgetting irrelevant details and things no longer relevant. This enables them to focus on the relevant things and to efficiently make decisions in their current life situations. This ability of the human brain inspired the idea of considering a form of forgetting for digital memories in the ForgetIT project. It is investigated, if similar mechanisms in digital memory might be helpful in better coping with the general situation of information overload, i.e., helping to focus on the really relevant digital resources. However, if we just copy human remembering and forgetting, the digital memory would just forget the same things as the human. Clearly, this is not the goal. Rather, the methods in digital memory should learn from and *complement* the human memory: it is desirable that a computer focuses on ”remembering” those things that the human might forget, but which are still useful or required later.

The idea of complementing human memory suggests to take an embracing perspective on human and digital memory, considering them as a joint system. In such a system, the two parts are not considered in isolation: their interactions and mutual influence are taken into account as well. Figure 1 shows models of the human memory and a digital system as well as interactions (joint system perspective). Further information can be find in deliverables D2.2 [ForgetIT, 2014a] and D2.3 [ForgetIT, 2015a]. In particular, the high-level model of the functioning of the human brain described above is depicted in Figure 1. The model represented in Figure 1 is focused on the synergy between human and digital memory, and derived from the text of deliverables D2.2 [ForgetIT, 2014a] and D2.3 [ForgetIT, 2015a]. The human memory (on the left) and the digital memory (on the right) together contribute to a form of virtual memory a human can rely on.

On the side of the human memory three main type of memory are distinguished: *working memory*, *episodic memory* and *semantic memory*. Together with the currently activated episodic and semantic memory, the *verbal short term memory* (things just heard) and the *visual short term memory* (things just seen) form the working memory, which frames the current situation. Knowledge is activated on demand from the semantic and episodic

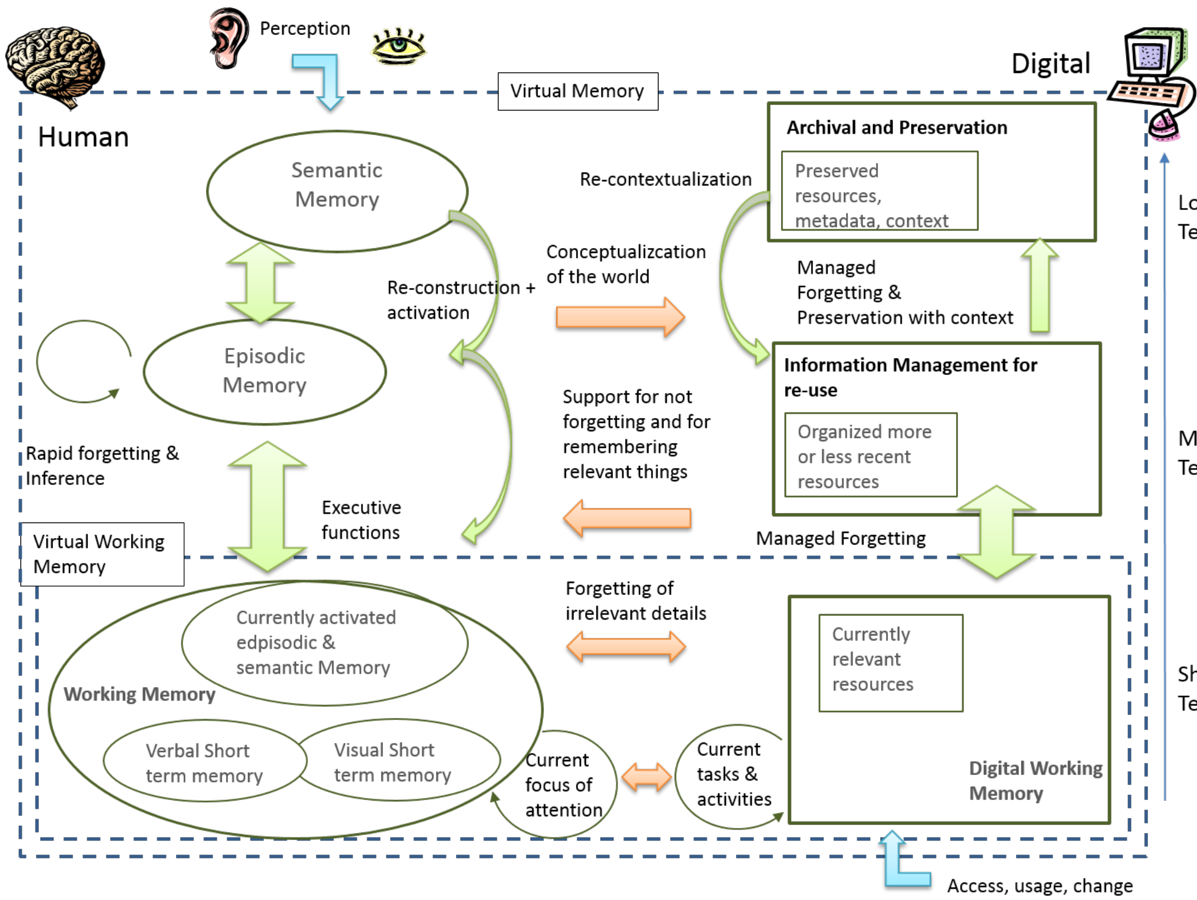

Figure 1: Joint Perspective on Forgetful, Interacting Human and Digital Memory, Derived from Material in D2.2 [ForgetIT, 2014a] and D2.3 [ForgetIT, 2015a].

memory according to current needs via the so-called *executive functions*.

Perception is one driver for such activation. It is worth noting that perceived signals do not directly become part of the verbal or visual short term memory, which are constantly updated, but they are rather filtered and interpreted by things already in the memory for making sense of the perceived signals.

Similarly, we also foresee a working memory within the digital memory in our model. This is composed of the digital resources currently relevant, e.g., used or relevant for current tasks or activities. The idea here is to clearly distinguish those resources from the rest of the digital resources and to keep them as close as possible to the user and easily accessible as possible (see also Memory Buoyancy). In an automated digital working memory signals from resource usage, pattern of usage and change as well as relationships between resources will be used to update the digital working memory. For this purpose, in our model, we introduce managed forgetting functions, which control the transitions between the different parts of the digital memory.

Together, the working memory and the digital working memory form the virtual working memory. Clearly, there is an influence between both of them. In the ideal case, the digital working memory would show to the user just all the information that the user needs in the current situation, but does not have in her working memory. Note, that it is also possible that there is an influence of the way the digital memory works on the human (working) memory (joint system perspective). For example, with the easy storage of phone numbers in mobile phones, humans no longer have to remember phone numbers.

Managed forgetting functions are also used to identify content that is of long-term value (see Section 2.2) and should, therefore, be preserved. In Figure 1, we distinguish (a) information management for re-use, as it is, e.g., done on a desktop computer or a server and (b) the system for archival and preservation (on top of the Figure). When content is transferred to *Archival and Preservation*, it makes sense to add context information to it (*contextualization*). This prepares the content for re-contextualization, which is required , when preserved content is brought back at a (much) later time. The idea of re-contextualization is to connect the re-activated content with the current environment or to - at least - make it understandable in the current environment. The idea of re-contextualization as an active situation-dependent process of bringing back things ”stored” in the memory is again inspired from human memory: when we as humans remember things this is also a re-construction process, which depends upon the current situation. Further details about foundations of contextualized remembering can be found in deliverable D6.1 [ForgetIT, 2013c] and following WP6 deliverables.

Episodic memory is mainly a detailed storage of events. It is typically subject to fast forgetting as well as blurring between the memory of similar events due to interference. Here, digital memory complementing human memory, e.g. via photos, can serve as a reminder of things that are forgotten, but that one might remember or refresh in a later point in time, e.g., reminiscing about past events (see also the discussion in deliverable D3.1 [ForgetIT, 2013a]). For this purpose it is crucial to select the adequate content to preserve, so that it can help remembering, e.g. considering coverage and diversity of preserved content.

Semantic memory is a more conceptual storage of memory, which stresses on patterns, abstractions and lessons learned. Here, the strongest interaction between human memory and digital memory is that the organization of digital resources does or should reflect the conceptualization of the world of the user, which is linked to her semantic memory. A more explicit modeling of the conceptualization of the world and a richer annotation of resource with this knowledge in *Information Management for Re-use* and in the *Archival and Preservation* (see Figure 1), as it is, for example, proposed by the Semantic Desktop approach (see WP9), can ease navigation and search of the user and thus re-finding things in the digital memory.

2.4 Forgetful: Focus on the Important Things

The ForgetIT project introduces the idea of an forgetful approach to information and preservation management as an alternative to the dominating *keep it all* approach. The forgetful approach opts for conscious decisions about what is important and thus should be kept (and preserved) replacing the often random form of forgetting (or losing) information as it can be often found with the *keep-it-all* approach (e.g. disk crashes, obsolescence of formats and technology, etc.).

Since preservation comes at a cost [Rauch et al., 2013] [Kejser et al., 2011] [Bote et al., 2012], it is important to make conscious decisions about what to preserve or how much to invest in the preservation of which part of the information space. For this need a forgetful approach is a good fit.

A forgetful approach is based on *Information Value* assessment, i.e. computing and predicting the value of information resources. Value for different purpose can be considered. In the context of this Reference Model, short-term and long-term values are important (see also Section 2.2). Effective information value assessment, especially for long-term information value, is a complex task involving a variety of parameters and heuristics. Based on such value, preservation decisions can be taken. On a high level, these decisions could include choice of preservation provider and/or service as well as decisions about redundancy and transformation.

2.5 Evolution-aware: Embracing the Long-term Perspective

Since we are targeting integrated information and preservation management systems, we are operating in a long-lived context, covering a time perspective of several decades. Even things that are considered relatively stable in the current setting of an information system - such as the type or class of content management system in use - will change over time. For being prepared for sustainable operation it is important to be prepared for such changes.

It is one of the core ideas the ForgetIT approach to keep the important information accessible and usable even in case of large changes in the setting and context of operation. For incorporating evolution-awareness into our Reference Model, several types of evolution with different impacts on the reference model have to be considered:

1. **Changes in conceptual model of the Active System**: this could be due, for example, to changes in the organizational ontology underlying the content structuring as well as processes described in the content. This creates a semantic gap between the archived content (relying on the old implicit or explicit ontology) and the active content (structure by new ontology). This gap has to bridged, at latest when preserved content is brought back into the active environment, in order to enable correct interpretation of the re-activated content;

2. **Active system evolution and exchange** (Migration): the used Active System might be subject to major changes or might even be completely replaced by another system, if we look at time frames of several decades. In spite of such changes the content should stay accessible and usable;

3. **DPS evolution or exchange**: in the same way, the chosen DPS might evolve or could be exchanged over time. This implies the migration of content into a new DPS. In the ideal case this should have as little impact on the Active System as possible;

4. **Change in best-practices and technology**: formats as well as employed technologies might become obsolete over time. This requires the identification of such changes as well as adequate actions to react to those changes, such as format transformations.

The last item in the list above is a classical issue of any DPS. It is, therefore, not covered in too much detail in our Reference Model , since we focus on the things that go beyond current best practices in digital preservation.

3 Preserve-or-Forget Reference Model

Following the ForgetIT approach, we are aiming for a smooth transition between active information use in the respective information management system (environment), which we call Active System, and the Digital Preservation System (DPS) (the PoF Framework architecture is discussed in deliverable D8.1 [ForgetIT, 2014c]). Therefore, in the Preserve-or-Forget (PoF) Reference Model we are considering active information use and preservation as part of a *joint ecosystem*, which stresses the smooth transitions and the synergetic interactions rather than the system borders. This also is a core distinction from the OAIS Reference Model [CCSDS, 2012], which is restricted to the DPS.

In Section 4, we discuss the mapping of the PoF Reference Model to the ForgetIT architecture. A possible mapping of this joint ecosystem onto three separate systems, i.e. the Active System (in the form of adapters and/or system extensions), the DPS (typically OAIS based) and the PoF Middleware, which couples both system in a flexible way is described in next Sections (see Section 4 for the mapping to the PoF architecture, Section 6 for the relationship with OAIS and Section 5 for the extensions of the Active System).

In describing the PoF Reference Model, we took inspiration from the way the OAIS Reference Model is structured. In this Section, we describe the functional view of the PoF Reference Model (functional model) and provide some preliminary ideas for the information model of the PoF Reference Model.

3.1 Functional Model: Layers, Workflows, Functional Entities

As in the case of the OAIS model, the function view of the PoF Reference Model, or the functional model for short, considers the main functional entities of the proposed reference model. Furthermore, we also describe the main workflows in the model and how the functional entities contribute to those workflows. Again the stress is on the parts, which connect the two types of systems, Active System and DPS, with each other.

The proposed PoF Functional Model is made up of three layers, namely the Core Layer, the Remember & Forget Layer and the Evolution Layer:

- the **Core Layer** considers the basic functionalities required for connecting the Active System and the DPS;
- building upon this layer, the **Remember & Forget Layer** introduces the brain-inspired and forgetful aspects into the PoF Reference Model implementing more advanced functionalities for the preservation preparation and the re-activation workflow;
- finally, the **Evolution Layer**, is responsible for all types of functionalities dealing with long-term change and evolution such as implementing the idea of contextualized remembering.

The different workflows and functional entities in the PoF Functional Model are associated to the three model layers above, as summarized in Table 1 and depicted in Figure 2. The description of the layers, workflows and functional entities is provided in the next Sections.

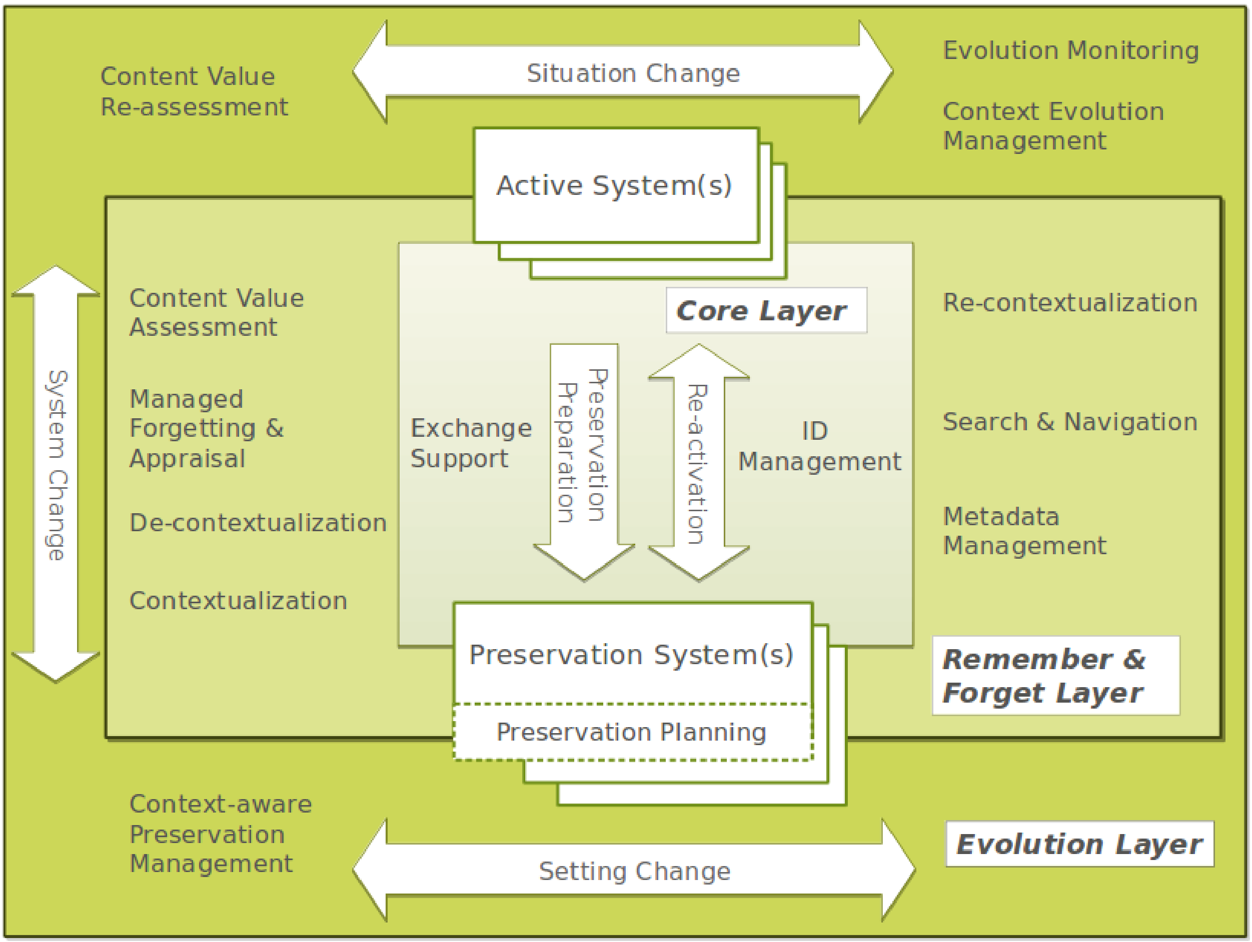

Figure 2: High Level Functional View of the PoF Reference Model

An overview of the PoF Functional Model components (layers, workflows, functional entities) is depicted in Figure 2): within each layer box the relevant entities and workflows are shown. In the following Sections we provide a more detailed representation of each workflow, with the steps associated to each process and the involved entities. It is worth noting that Figure 2 already makes some assumptions about the functionalities implemented in the Active System and, especially, the DPS: those functionalities, which are parts of one of the respective systems, are not explicitly listed in the PoF Reference Model. For our purpose, we assume a OAIS compliant DPS implementing functionalities such as *Ingest*, *Data Management*, *Preservation Planning*, *Archival Storage* and *Access* of preserved content (see [CCSDS, 2012]).

The three layers are used in the following to describe the functional view of the PoF Reference Model in more detail.

Model Layers	Workflows	Functional Entities
Core Layer	Preservation Preparation (*basic*) Re-activation (*basic*)	ID Management Exchange Support
Remember & Forget Layer	Preservation Preparation Re-activation	Content Value Assessment Managed Forgetting & Appraisal De-contextualization Contextualization Re-contextualization Search & Navigation Metadata Management
Evolution Layer	Situation Change Setting Change System Change	Evolution Monitoring Context Evolution Management Content Value Re-assessment Context-aware Preservation Management

Table 1: PoF Reference Model Components: Layers, Workflows and Functional Entities.

3.1.1 Core Layer

The Core Layer embraces basic forms of two workflows connecting the Active System and the DPS, the **Preservation Preparation** workflow and the **Re-activation** workflow, and includes two functionalities, **ID Management** and **Exchange Support** (see Figure 3). The workflows and the functional entities are described in the following.

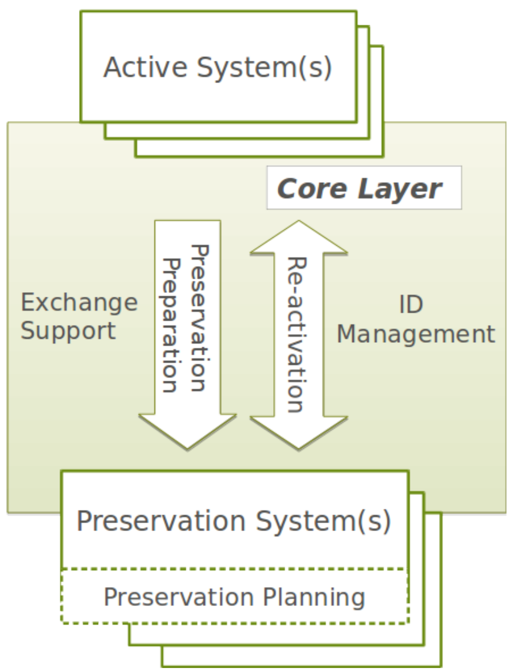

Figure 3: Core Layer of the PoF Reference Model

The Preservation Preparation workflow takes care of transferring content to be preserved between the Active System and the DPS. On an abstract level, this workflow - in its basic form - consists of five steps (see also Figure 4) (1) **select** the content to be archived, (2)

provide the content to the archival process (3) **enrich** the content with context for preservation (4) **package** the content according to the expectations of the DPS (5) **transfer** the content into the DPS. In terms of preservation terminology, the preservation preparation workflow can be considered a pre-ingest workflow, which leads into the ingest functionality of the DPS. This interaction is discussed in more detail in Section 6. Furthermore, it is worth noticing that the enrich functionality is available on the Core Layer only in its basic form (e.g. to add technical information such as file format to the content to be archived). More advanced enrich functionality is discussed for the Remember & Forget Layer.

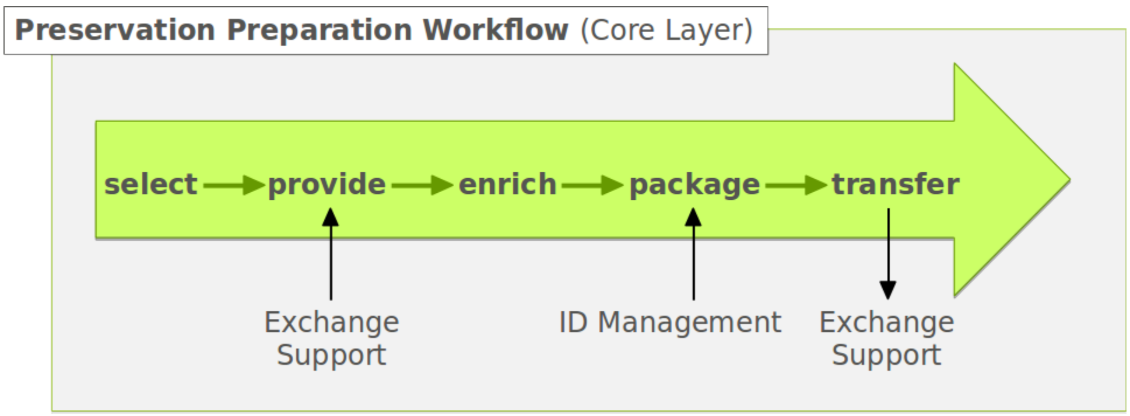

Figure 4: Preservation Preparation Workflow (Basic) in the Core Layer

The *Re-activation* workflow takes care of enabling the Active System to retrieve and re-activate content, which has been transferred to the DPS. Again on an abstract level, this workflow - in its basic form - consists of five steps (see also Figure 5): (1) **request** the content to be retrieved from the PDS (here via its identifier), (2) **search** requested content thus translating the request into archival ID(s) (in the basic workflow just using the ID Manager) (3) **fetch** the respective content from the DPS (4) **prepare** the content for delivery to the Active System, (4) **transfer** content to the Active System. The link to the DPS is for the Core Layer mainly in the fetch activity, which interacts by the access functionality offered by the DPS (see also Section 6).

As in the case of the Preservation Preparation workflow, some of the steps in the Re-activation workflow are only included in a very basic form on the Core Layer and are extended with more advanced functionalities on the Remember & Forget Layer.

In support of these two core workflows in their basic form the Core Layer includes two types of further functionality, namely ID Management and Exchange Support:

ID Management The ID Management is mainly responsible for mapping between the IDs of the resources in the Active System and IDs that are used in the DPS for identifying (and locating) the respective content resources. Since several versions of the same resource can be put in the DPS, the ID Management also has to take care of resource versions and their mappings to archive IDs.

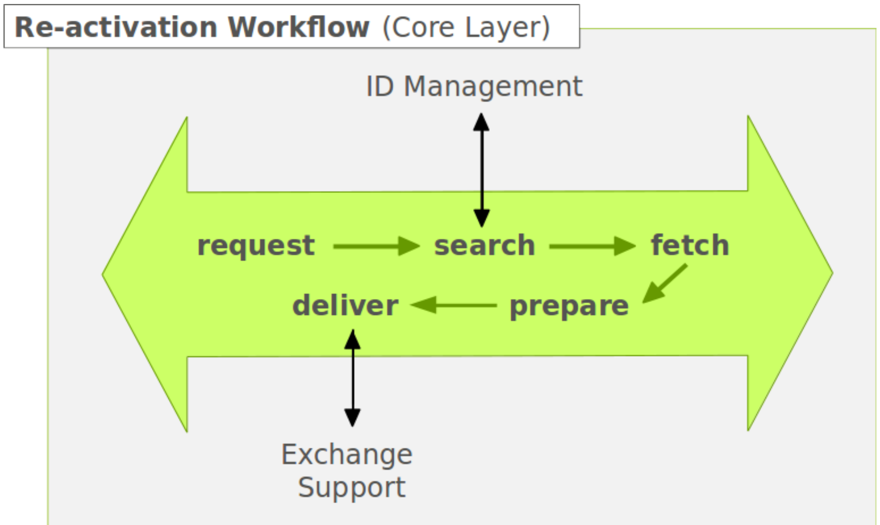

Figure 5: Re-activation Workflow (Basic) in the Core Layer

Exchange Support Exchange support is responsible for enabling the exchange of content and metadata between the Active System and the DPS. It adapts and maintains protocols for this purpose. The Exchange Support handles both outgoing information and incoming packages that should be put back into active use. The Exchange Support can be considered as a client-side communication adapter and can, for example, be implemented in the form of a repository.

3.1.2 Remember & Forget Layer

The Remember & Forget Layer introduces brain-inspired functionality into the PoF Reference Model, which targets the concepts of managed forgetting and contextualized remembering. For this purpose, the Remember & Forget Layer extends the two workflows **Preservation Preparation** and **Re-activation** from the Core Layer with further, more advanced functionalities: **Content Value Assessment**, **Managed Forgetting & Appraisal**, **De-contextualization**, **Contextualization**, **Re-contextualization** and **Search & Navigation**, which are all described in the following. All of those listed functionalities create additional metadata, which have to be managed in a systematic way. Therefore, the Remember & Forget Layer also contains a functional entity for **Metadata Management**.

In more detail, the Preservation Preparation workflow, which still consists of the five steps *select*, *provide*, *enrich*, *package* and *transfer* (see Figure 6), now uses the additional functionalities of Content Value Assessment and Managed Forgetting in the phase of selecting content for preservation:

Content Value Assessment Understanding the value of content is in the core of content appraisal for preservation and managed forgetting. Content value assessment

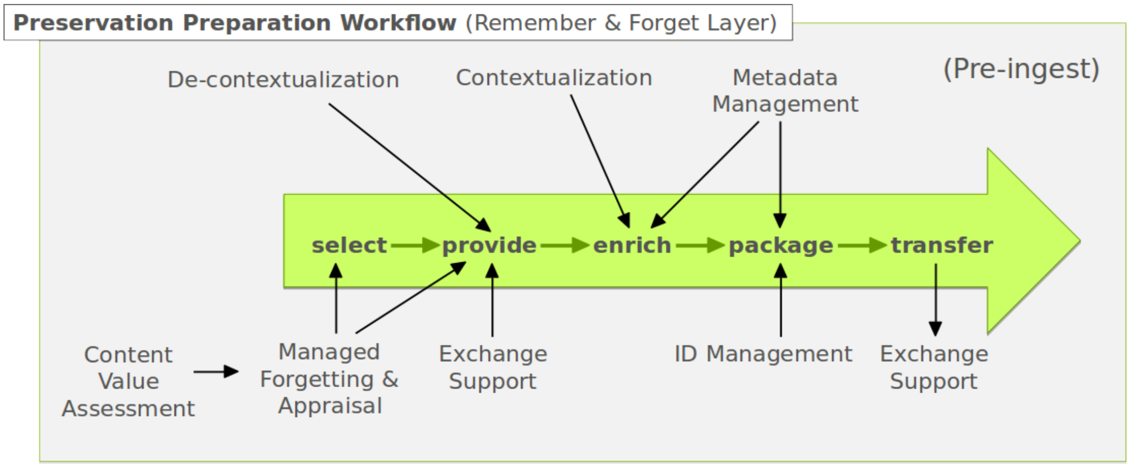

Figure 6: Preservation Preparation Workflow in the Remember & Forget Layer

aims to determine the value of a resource. This value may change over time and there are different value dimensions, which reflect the value considering different purposes or perspectives and which may influence each other. There is, for example a value dimension reflecting current importance, e.g. Memory Buoyancy (MB), and a dimension reflecting the long term importance or relevance of a resource, the Preservation Value (PV). For assessing content value, the content value assessment component takes evidences from the Active System, e.g. about information use, content creation, and further knowledge about the role of resources in the Active System. Content value can be used as a basis for making preservation decisions, e.g. if a resource should be preserved or how much should be invested in the preservation of a resource. Content value can also be used in the Active System, e.g. for especially highlighting resources with high content value.

Managed Forgetting & Appraisal With the dramatic growth of the amount of content, nowadays it becomes more and more important to make conscious decisions about preservation. Clear decisions on what to put into the DPS and explicit content appraisal have always been part of the processes of an archive [Harvey, 2006], although not always as much in personal archiving [Marshall, 2011]. The component for appraisal and managed forgetting aims to help in automating such decisions, a need that has been identified earlier [Harvey and Thompson, 2010], for both personal archiving as well as organizational settings. This is encapsulated in the concept of managed forgetting, which uses the results of content value assessment for deciding about preservation and forgetting actions. The effects of managed forgetting functionality is not restricted to the preservation functionality. It can also be used in the Active System for improved information access.

Furthermore, the workflow steps provide and enrich are extended with De-Contextualization and Contextualization functionality, respectively:

De-Contextualization De-Contextualization refers to the extraction of an object from its

Active System context in preparation of packaging it for archiving. Decoupling the object under preservation from its Active System context is non-trivial, since it has to be decided how much of its current context has to be taken for its future contextualization and where a cutting can and should be made. De-contextualization and contextualization (see below) are conceptually closely related.

Contextualization Contextualization consists in providing sufficient additional information for the content to be preserved, in order to allow archived items to be fully and correctly interpreted at some undefined future date. This entity is responsible for defining and assigning the appropriate context to content to be archived. Contextualization can leverage other processes (similarity analysis, concept detection) to explicate context. Contextualization provides the basis for the management of context evolution over time (see Evolution Layer in Section 3.1.3) and Re-contextualization (see Re-activation workflow in Figure 7).

The Preservation Preparation workflow is linked to the **Pre-Ingest** functionality as it is described for Preservation Systems, e.g. in the OAIS model. In order to facilitate easy (seamless) ingest into the DPS and make sure that the packages contain metadata needed for both the DPS as well as for access, pre-ingest aids the Active Systems as well as the DPS systems adhering to standard protocols and metadata. This also means that the Pre-ingest function puts up some requirements on the Active Systems to follow certain protocols (which can/should be domain specific). Our Preservation Preparation workflow, from the perspective of the preservation system serves as a rich pre-ingest function.

In general, before the DPS process of transferring digital collections from Active System to DPS, a submission agreement has to be established between the participating organizations, following a standard approach, as described in the Producer-Archive Interface Methodology Abstract Standards (PAIMAS) [CCSDS, 2004].

The agreement should contain accurate information about package content, structure, and metadata. It should also include requirements for security and privacy mechanisms at transfer and storage. There should be stated in the agreement if there is a need for migration at ingest. Other examples is a specification to what extent metadata should be obtained and generated during the pre-ingest process and if there is specific demand on storage. In the case of the PoF, this agreement is mainly reduced to an agreement on interfaces between the Active System and the Remember & Forget Layer. In addition, there are however, still needs for agreements e.g. about security and privacy requirements.

For the Re-activation workflow, two types of additional functionality are used, with the respect to the Core Layer, namely Re-contextualization and Search & Navigation:

Re-contextualization The purpose of the Re-contextualization functional entity is to support the interpretation of a content object at the time of access (which might be a considerable time after archival). Re-contextualization occurs when a document is retrieved from the DPS at some future date. Once retrieved from the DPS and

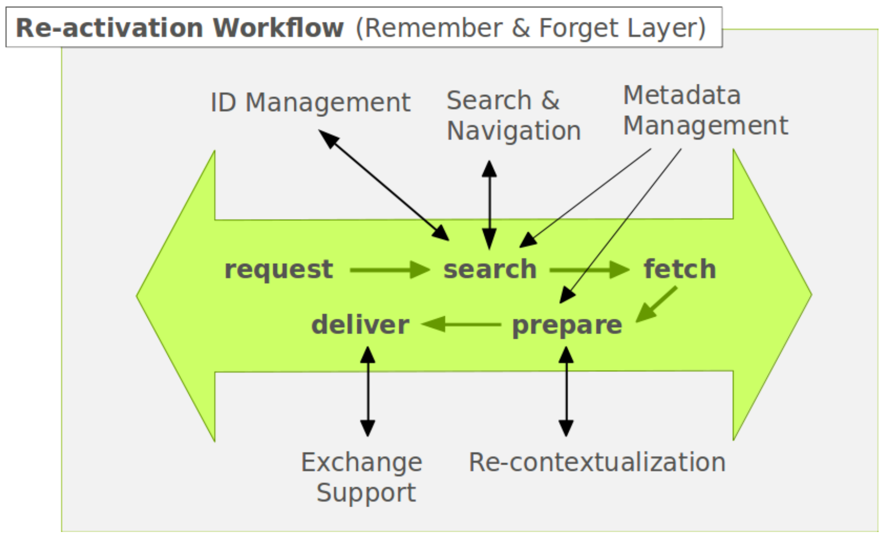

Figure 7: Re-activation Workflow in the Remember & Forget Layer

before it is put back into active use, the context information, which has been provided by the contextualization functionality, stored together with the content object and possibly updated or extended over time is retrieved. This context information is used for Re-contextualization, i.e. to relate the content object to the current usage context. Re-contextualization can also include the re-construction or extension of context information for content archived with no or not sufficient original context.

Search & Navigation The Search & Navigation functional entity is responsible to enable finding things that have been preserved. Various types of search and navigation will be supported here. This includes search in the metadata, full-text search in the content (or more general content-based search also including non-textual content), search in the context information and in other types of annotation, exploratory search for understanding the archive content, etc. What is crucial for the integrative and forgetful approach we are following here, is (a) to manage the interaction between the search in the Active System and the search in the DPS and (b) to understand how the forgetful approach and search support interact. Since we are following an integrative approach, it makes sense for (a) to consider the information in the Active System and the DPS as a type of a *joint virtual information space*, which are both considered for search in the Active System. However, it might still make sense to differentiate the two types of content taking into account the cost that might be attached to accessing content from the DPS. Archive content might, for example, only be considered on demand or if nothing can be found in the Active System. Furthermore, content stemming from the preservation store might be marked in result lists. For aspect (b), the influence of the forgetful approach, the results of content value assessment, namely MB and PV, can be considered in result ranking (or even indexing): this would prefer results with higher content value balancing content value and relevance as it is for example done in diversification approaches. Furthermore, this includes adequate filtering and ranking approaches for handling versioned archive content.

Finally, in both workflows metadata are generated and used for different purposes. This metadata is taken care of by the functional entity for Metadata Management:

Metadata Management The Matadata Management functional entity is responsible for the different kinds of metadata that are created and exchanged by the aforementioned functional entities. This includes a variety of different types of metadata such as current and past values for memory buoyancy and preservation value, information on the context of a resource, information extracted from resources for further processing as well as indexing information for search and navigation. Some of the metadata, which is collected here just remains in this middle layer for supporting its operation. Other parts of the metadata such as the preservation value and the context information will be stored as part of the archived object in the preservation system. The Metadata Management of the Remember & Forget Layer clearly interacts with the respective components of the preservation system by (a) providing input for enriching the metadata in the preservation system for improved preservation management, (b) incorporating information coming from the preservation system (e.g. for the joint indexing) and (c) by - as mentioned before - storing some of the metadata as part of the resources to be preserved.

3.1.3 Evolution Layer

The Evolution Layer contains three new workflows: the **Situation Change** workflow is responsible for monitoring semantic changes in the Active System and to propagate them into the DPS, in order to keep the preserved content understandable; the **Setting Change** workflow deals with changes in practices, formats and technology in the environment of the preservation (setting); finally, the **System Change** workflow is responsible for situations, where one of the involved systems changes. The first two workflows are described in more detail below, whereas the last workflow is left to the revised version of the PoF Reference Model which will be reported in deliverable D8.5 [ForgetIT, 2016]. In support of the aforementioned workflows, the Evolution Layer includes additional functionalities: **Evolution Monitoring**, **Context Evolution Management**, **Content Value Re-assessment** and **Context-aware Preservation Management**.

The Situation Change workflow consists of four steps (as depicted in Figure 8): (1) **change monitoring**, (2) **change assessment** (assessment of detected changes), (3) **change notification** (notification of involved components as well as the DPS on relevant changes) and (4) **change propagation**, which performs different types of actions depending on the observed change and the chosen change propagation strategy.

For monitoring, assessing the changes and deciding about the consequences, in support to the Situation Change workflow, the functionalities Evolution Monitoring (mainly part of Active System), Context Evolution Management and Content Value Re-assessment have been introduced in the Evolution Layer:

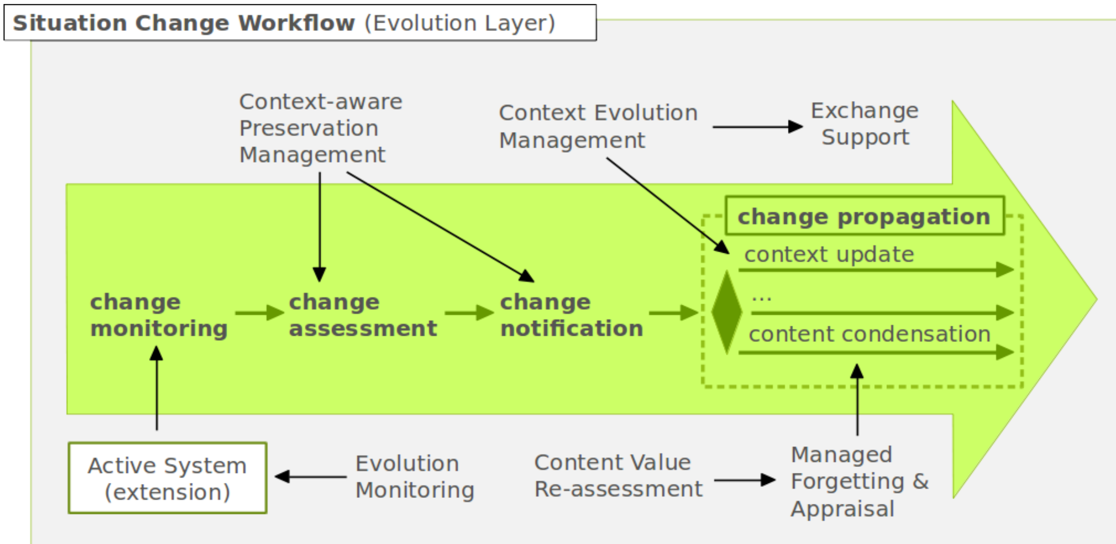

Figure 8: Situation Change Workflow in the Evolution Layer

Evolution Monitoring The Evolution Monitoring functionality is required for the change monitoring step in the Situation Change workflow. This is mainly performed in the Active System (extension), because this is the main place, where evolution in the conceptual model of the application become visible. Such changes might, for example, be changes in the ontology (implicitly) underlying the organization of the information triggered e.g. by a major re-organization. Evolution Monitoring has to observe changes in the explicit representations of the conceptual model (such as taxonomies, information structures) in the Active System as well as more implicit signals of context evolution (e.g. newly upcoming topics, tags getting out of use, department sites no longer updated).

Context Evolution Management The Context Evolution Management is responsible for keeping up-to-date the context information which has been stored along with the archived content. This may include the storage of further context information, in case of larger changes in the Active System (e.g. major restructuring of an organization). This also includes keeping track of such larger changes, which have happened in the Active System on a conceptual level. Such a change history can be used in Re-contextualization for making the content understandable in the changed new context.

Content Value Re-assessment The functionality Content Value Re-assessment serves a very similar purpose as the functionality Content Value Assessment described for the Remember & Forget layer. It re-visits the value assessments originally provided by the Content Value Assessment based on observed evolution in context. It is considered separately in our model, since the resources that it works on are now already preserved, which has implications on the computation of the value (e.g. the role of usage data) as well as on where this functionality is performed. One option, which is followed in the ForgetIT project is to map it to in storage computation.

The Setting Change workflow consists of four different phases with two different starting points (as depicted in Figure 9): the **activity monitoring** which (1.1) logs the bi-directional communication between the Active System and DPS and (1.2) receives a change request from a DPS, (2) the **change assessment** that detects and triggers change requests, (3) the **change estimation** of suitable change recommendations based on content value, purpose of use, and use statistics, (4) the **change recommendation**, namely the notification of recommended actions to DPS, which could be of different types, such as change of content or change of physical and logical content structure.

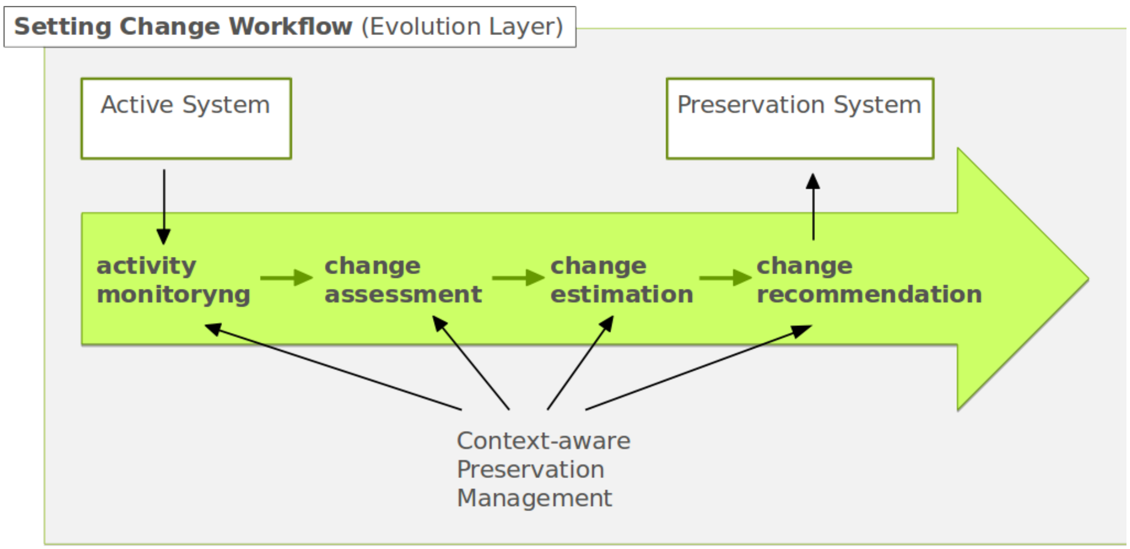

Figure 9: Setting Change Workflow in the Evolution Layer

The Evolution Layer includes the functional entity Context-aware Preservation Management, supporting the workflow described above:

Context-aware Preservation Management The Context-aware Preservation Management functional entity extends OAIS Preservation Planning and OAIS Administration by keeping track of Active Systems as well as the (several) DPS that are involved. The main idea here is that preserved information should be easy to seamlessly put back into active use in either the same system as it originates from, but even more importantly other information systems (of the same type). Even the same system might have evolved to newer versions with different standards, and maybe even a different information structure or ontology. This must also be tracked by the Context-aware Preservation Management.

The above functionality is closely linked to typical OAIS functionalities of the underlying Preservation System, such as Preservation Planning and Administration. **Preservation Planning** is an OAIS functional entity residing in the DPS, that is responsible for developing preservation policies and requirements by monitoring technological changes and needs and requirements from the user community. Further information is available in Section 6, where the mapping of the PoF Reference Model is discussed.

The functional entity (Preservation) **Administration** provides the "spider in the net" of the DPS, handling contracts regarding e.g. submission agreements, and taking decisions on new policies and standards for the DPS. It is also in charge of managing software and hardware configurations and the overall DPS operations on a daily basis. The Administration functional entity receives input from all other functional entities in the OAIS model. Part of this functionality needs to be externalized since the PoF Middleware (see Section 4) needs to keep track of several systems and adapt functionality for the seamless interaction/integration between Active Systems and DPS, and also to give input to several DPS. Finally, another OAIS functionality related to Context-aware Preservation Management is **Data Management**, which manages OAIS internal data and provides search metadata to the Access function. This function also manages internal integrity in the DPS. Part of this functionality needs to be external to the DPS in order to facilitate PoF functionality, including support for using several different DPS.

3.2 Information Model: Core Elements

In this Section, we provide some preliminary ideas for the definition of an information model as part of the PoF Reference Model, following the approach used in OAIS. The information model again focuses on the needs of bridging the gap between the Active System and the DPS. In order to meet the requirements of a variety of systems on the site of the Active System as well as on the site of the DPS, the information model has been kept quite generic in this reference model.

The core elements of the information model are represented by the UML diagram depicted in Figure 10, where we included the main types used to represent the objects of preservation.

The core element of the information model is an *Archival Object* (see Figure 10), which is made up of a `Content Object` and a `Context Object`. `Content Object` and `Context Object` are linked to each other by a `gives_context` link.

Both `Content Object` and `Context Object` are of type `Archival Resource`, which is described by a `Metadata` type and is represented by an `Information Object` type. An `Information Object` can be a *Single Object* such as a photo, a file, or a concept or it can be a `Complex Object`. An important type of `Complex Object` is a `Collection` of objects that are to be archived together. A second type of `Complex Object` is an `Object Graph` of interlinked `Information Object` types.

Context is represented by either a `Local Context` or a `World Context` type. The `Local Context` has to be stored in the PDS, because it can, in future, not be easily found elsewhere. `World Context` is context information that is expected that also in future can be found somewhere else and, thus, it might be sufficient to store only some

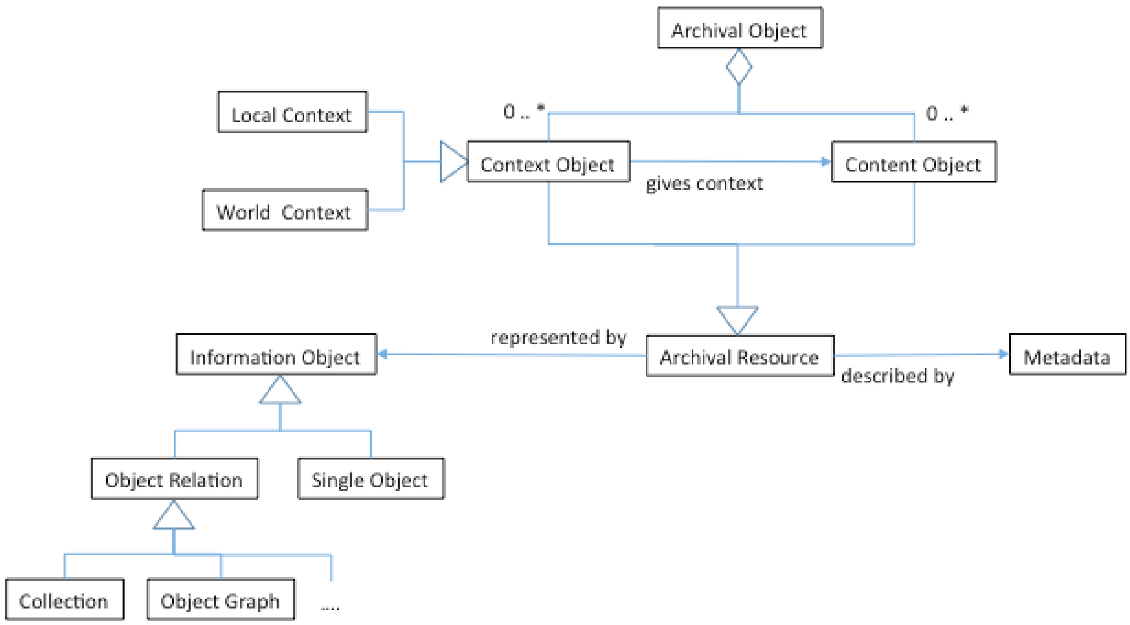

Figure 10: Abstract Information Model of the PoF Reference Model

references to keep this context information. See deliverable D6.1 [ForgetIT, 2013c] for a more detailed discussion of this issue.

The information model drafted here will be further refined and described with all details in the next version of the PoF Reference Model, in deliverable D8.5 [ForgetIT, 2016].

4 Mapping to ForgetIT Architecture

In the following, we describe how the PoF Reference Model can be mapped to the architecture of the PoF Framework, described in deliverables D8.1 [ForgetIT, 2014c] and D8.3 [ForgetIT, 2014d]. Please note that some organizational aspects (such as PoF Middleware ownership) are not discussed here, they will be included in the final version of the PoF Reference Model, in deliverable D8.5 [ForgetIT, 2016].

4.1 PoF Framework Architecture

The architecture of the PoF Framework, described in detail in deliverable D8.1, is made up of three layers: *Active Systems*, *Preserve-or-Forget (PoF) Middleware* and *Digital Preservation System (DPS)*, as depicted in Figure 11. The Active Systems represent user applications, while the PoF Middleware is intended to enable seamless transition from Active Systems to the DPS (and vice versa) for the synergetic preservation, and to provide the necessary functionality for supporting managed forgetting and contextualized remembering. The PoF Middleware provides also the integration framework for all components developed in WP3-WP6. The DPS is composed by two sub-systems (Digital Repository and Preservation-aware Storage, including a Cloud Storage Service) and provides both content management and typical archive functionalities required for the synergetic preservation.

The deliverable D8.1 also contained a preliminary discussion about the role of the OAIS Reference Model in the overall architecture: according to the project proposal, since OAIS nowadays is the most recognizable conceptualization of a digital preservation system, it was considered as one of the building blocks of ForgetIT approach. However, the model described in Section 3 complements and supersedes this initial approach towards the OAIS model. We further discuss the relationship with OAIS in Section 6.

4.2 Relationship between PoF Reference Model and PoF Architecture

We provide the mapping between each functional entity in the PoF Reference Model and the architectural components of the PoF Middleware in Table 2. The list of components is taken from deliverable D8.1. It is worth noting that for some components the mapping with model entities is not one-to-one, because more than one component can participate in the implementation of a given functional entity of the model.

It is worth noting that the `Scheduler` component is not explicitly mentioned in Table 2, because it mainly provides process management functionalities supporting the different workflows across the three layers described before.

Functional Entity	Model Layers	PoF Middleware Components
ID Management	*Core*, *Remember & Forget*	`ID Manager`
Exchange Support	*Core*, *Remember & Forget*	`Collector`, `Archiver`, `Metadata Repository`
Content Value Assessment	*Remember & Forget*	`Forgettor`
Managed Forgetting & Appraisal	*Remember & Forget*	`Forgettor`
Contextualization	*Remember & Forget*	`Contextualizer`, `Extractor`, `Condensator`
De-contextualization	*Remember & Forget*	`Contextualizer`
Re-contextualization	*Remember & Forget*	`Contextualizer`, `Archiver`
Search & Navigation	*Remember & Forget*	`Navigator`
Context Evolution Management	*Evolution*	`Context-aware Preservation Manager`, `Contextualizer`
Context-aware Preservation Management	*Evolution*	`Context-aware Preservation Manager`
Preservation Planning	*Evolution*	`Context-aware Preservation Manager`
Administration	*Evolution*	`Context-aware Preservation Manager`
Pre-ingest	*Evolution*	`Archiver`

Table 2: Mapping between PoF Reference Model Functional Entities and the PoF Middleware Components.

The PoF Functional Model described in Section 3.1 includes several workflows spanning across the three model layers:

- the Preservation Preparation workflow (Figure 4 and Figure 6) for the Core Layer and for the Remember & Forget Layer;
- the Re-activation workflow (Figure 5 and Figure 7) for the Core Layer and for the Remember & Forget Layer;
- the Situation Change workflow (Figure 8) for the Evolution Layer;
- the Setting Change workflow (Figure 9) for the Evolution layer.

The different steps of such workflows involve different PoF Framework components, mainly for what concerns the PoF Middleware. We provide a representation of the activation of the different components in each workflow in Figure 12, Figure 13, Figure 14 and Figure 15, respectively.

It is worth noting that some components are involved only in one of the layers, e.g. the Remember & Forget Layer or the Evolution Layer (see Table 2) and that Figure 15 about the Setting Change workflow includes also the Active System and Preservation System components, while the others involve only the components within the PoF Middleware.

4.3 Model Implementation in PoF Architecture

The main challenges in implementing the PoF Reference Model are associated to (a) the integration of the components within the PoF Middleware, (b) the integration of the Active Systems and of the Preservation Systems with the PoF Middleware, and (c) the implementation of some specific components which are crucial for the new ForgetIT approach to digital preservation described in this model. Examples of such components are the `Forgettor`, the `Contextualizer` and the `Context-aware Preservation Manager`, just to name a few.

The integration of the components in the PoF Middleware leverages the outcomes of WP5, detailed in deliverables D5.1 [ForgetIT, 2013b] and D5.2 [ForgetIT, 2014b], which described the foundations of synergetic preservation and a workflow model and prototype for the transition between Active System and DPS. The communication among the components and the business logic for the different workflows are implemented in the PoF Middleware using a Message Oriented Middleware (MOM) approach [Chappell, 2004], powered by a rule-based engine which activates the different components according to specific Enterprise Integration Patterns (EIP) [Hohpe and Woolf, 2003], based on best practices in the field of Enterprise Application Integration (EAI). The choice of the most suitable solution is performed according to ForgetIT requirements and also taking into account the future adoption of the PoF Framework (see deliverable D8.3). The adoption of any middleware technology requires some effort by software architects and engineers before it can be used in a effective way. The integration of the Active Systems and the Preservation Systems should be based on standard technologies and robust APIs, to enable the integration with different preservation solutions and user applications.

In the previous Section we described how the different components of the PoF Framework are related to the PoF Reference Model functional entities. The final release of the PoF Framework will be fully compliant to the model described here. The first release of the prototype described in deliverable D8.3 already integrates many components and implements two priority workflows for basic synergetic preservation and managed forgetting support, as described in Section 4.2 of deliverable D8.1. The second release of the PoF Framework, described in deliverable D8.4 [ForgetIT, 2015c], will be further improved according to the model.

The main challenge in implementing PoF Framework components is related to the novelty of the approach and to the lack of similar technologies or applications. Examples of such components are the `Context-aware Preservation Manager` (synergetic preservation), the `Forgettor` (managed forgetting) and the `Contextualizer` (contextualized remembering), just to name a few which are closely related to the core ForgetIT principles. The development of such components is based on a iterative approach, with new requirements identified within the project as a result of the analysis and discussion among all partners, resulting in the identification of the main processes and the relevant scenarios to be implemented. Moreover, the development of the Active Systems for the personal and organizational scenarios has to provide a proof of the ForgetIT approach validity through the implementation of application pilots. Other components in the PoF Middleware which have not been mentioned above are crucial for the implementation of the relevant workflows, providing the required input for preserve-or-forget decisions, dealing with the core preservation objects in the information model, managing metadata, logging processes, etc.

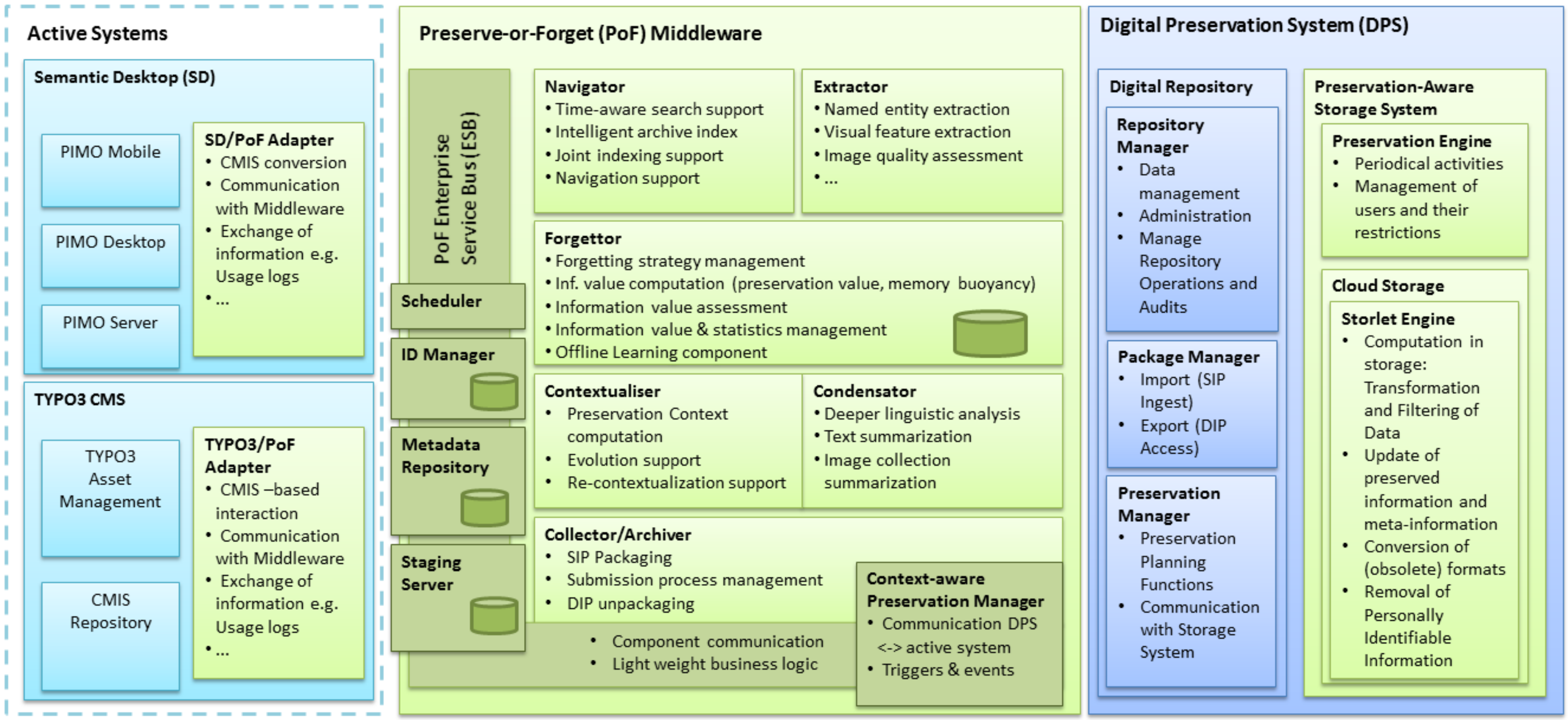

Figure 11: Architecture Diagram of the Preserve-or-Forget (PoF) Framework.

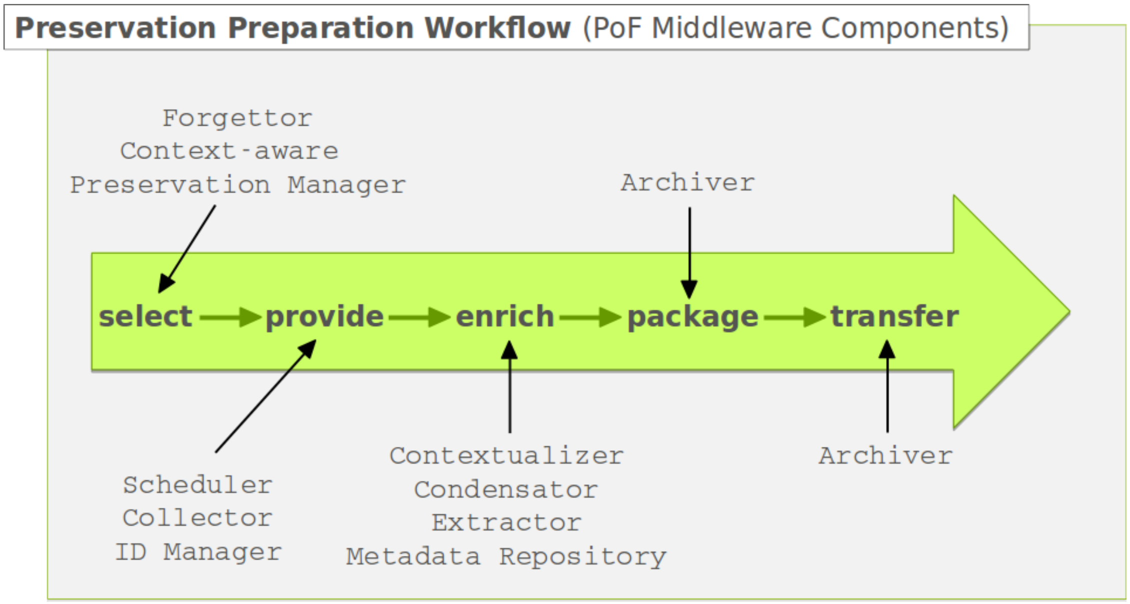

Figure 12: Mapping between the PoF Middleware Components and the Preservation Preparation Workflow.

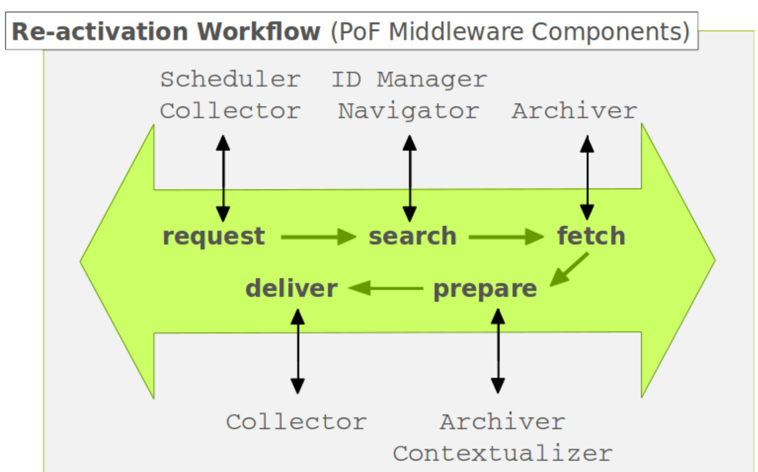

Figure 13: Mapping between the PoF Middleware Components and the Re-activation Workflow.

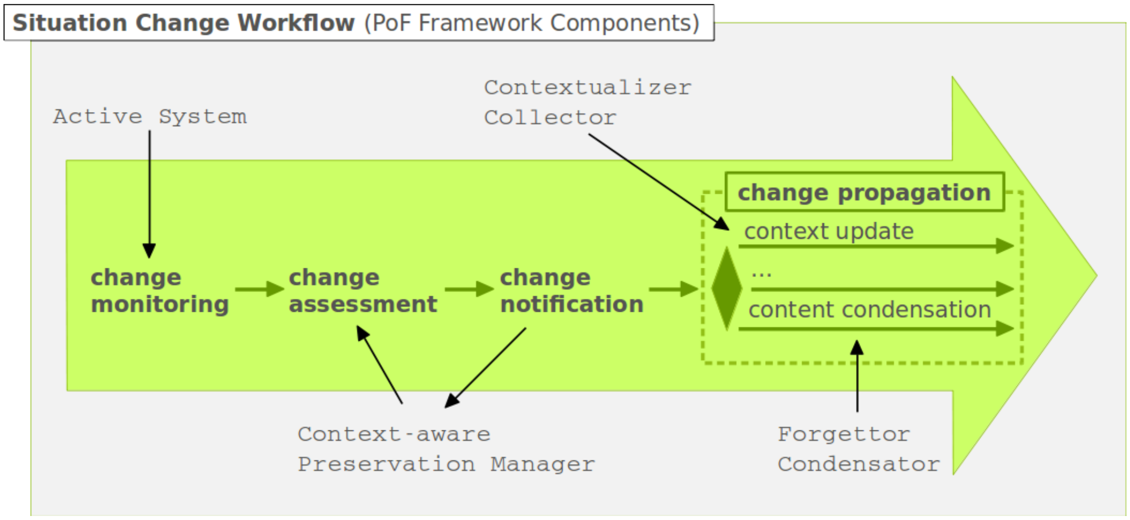

Figure 14: Mapping between the PoF Framework Components and the Situation Change Workflow.

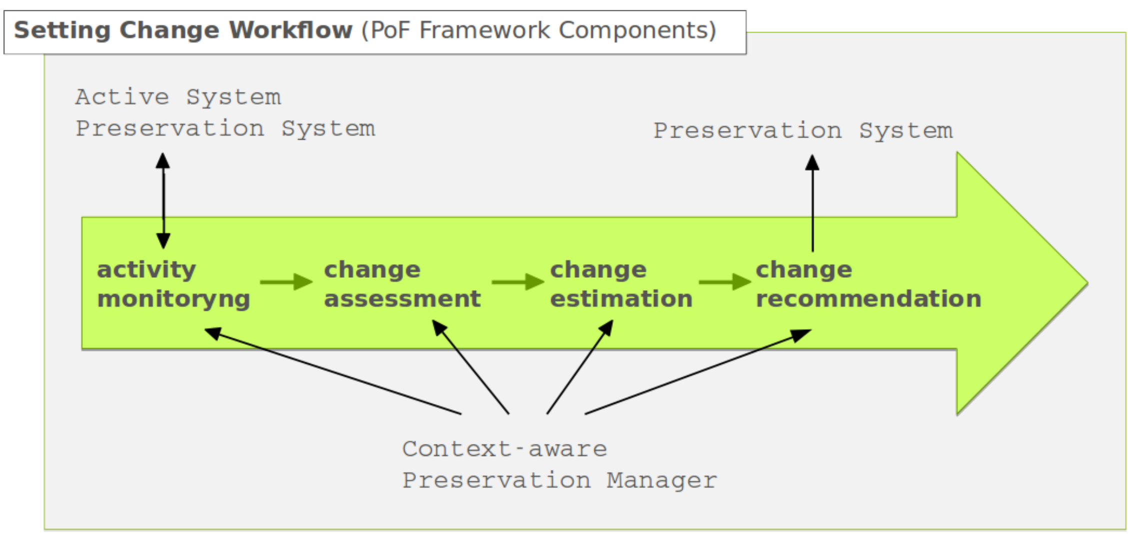

Figure 15: Mapping between the PoF Framework Components and the Setting Change Workflow.

5 Information Management Systems Extensions

The goal of the ForgetIT approach is to keep the impact of introducing preservation into the information management workflow as small as possible. Besides the long-term preservation of selected content - which is the aim - the approach also has the potential to introduce other more immediate benefits into active information management.

Both for the basic functionality of supporting preservation as well as for leveraging the benefit enabled by the approach, some extensions are required in the Active System. These are discussed in more detail in Section 5.1, while Section 5.2 elaborates on the benefit that can be created in the Active System. Finally, Section 5.3 focuses on preservation strategies, discussing different interactions of the Active System with the DPS.

5.1 Extensions of the Active System

Extensions to the Active System are required, where it has to interact with a DPS (possibly via a middleware as in the case of ForgetIT architecture) and where information has to be provided for the targeted intelligent preservation processes. This includes the collection of evidences for information value assessment and the collection of information in support of contextualization.

5.1.1 Supporting Information Exchanges

A core functionality, which needs to be enabled for synergetic preservation is *information exchange* between the Active System and the DPS. Information to be exchanged includes the content to be preserved as well as metadata and context information describing this content. Furthermore, it has to be possible to bring content from the DPS back into the Active System (see Re-activation in Section 3.1.1). Thus, bi-directional information exchange has to be enabled. This can be enabled for example by a repository used by both sides for making content available to the respective other system (plus possibly a notification channel). This approach is investigated in WP5 deliverables (see for example deliverable D5.1 [ForgetIT, 2013b] and D5.2 [ForgetIT, 2014b]). The approach adopted in ForgetIT makes use of a standard-based repository leveraging the content exchange standard CMIS [OASIS, 2013], which enables the interaction with different Active Systems. Besides these asynchronous channels, more synchronized forms of information exchange are also possible, such as direct service call.

5.1.2 Information Value Evidences

The Managed Forgetting & Appraisal function described in Section 3.1.2 heavily depends on the idea of content value assessment, which is discussed mostly in WP3 deliverables (see for example deliverable D3.3 [ForgetIT, 2015b]). This is true for assessing short term importance as it is done for computing Memory Buoyancy (MB) as well as for assessing the long term value, named Preservation Value (PV).

For substantial content value assessment, evidences have to be collected from the Active System. For the short-term importance these are for example information about the usage pattern of a resource as well as information about the relationship between resources. In order to provide such evidences, specific interfaces as well as protocols are required, which define which evidences are provided, in which format and in which frequency. Furthermore, methods for collecting such evidences have to be implemented in the Active System, if not yet available. In the opposite direction, the Active System might also profit from the computed content value information and use it for advanced functionalities and generate short-term benefits (see Section 5.2).

5.1.3 Supporting Contextualization

Context information added to the preserved content is meant to ease the interpretation of such content in case of re-activation. The information contextualization is investigated in WP6 (see for example deliverables D6.1 [ForgetIT, 2013c]). Context information can be gained in different ways, since (a) it can be provided by the Active System at the time content is sent to the DPS, (b) the PoF Middleware can automatically extract information from the provided content and other sources such as e.g. a domain specific ontology or external knowledge sources (e.g. Wikipedia) and (c) it can be a mix of the previous two approaches.

If this is possible in the considered Active System, harvesting context information which is already explicated in the Active System (option (a) above) looks more promising. In this way a richer and more quality-controlled form of context can be provided, with respect to what can be automatically extracted in the preservation process. For example in the Semantic Desktop (see WP9) content is already annotated using a ontology, i.e. the Personal Information MOdel (PIMO). This annotation, obviously, is a good source of context information for preservation.

However, option (a) also puts higher requirements on the Active System: (1) explicated context has to be available (or it has to be explicated for this purpose) and (2) the Active System has to be extended with a functionality that is able to attach context information to the content information sent for preservation.

5.2 Creating Benefit in the Active System

Investment in preservation is typically paying back on the long run only. In order to foster the adoption of preservation technology, it is one of the goal of the ForgetIT approach to also create short term benefit in the Active Systems - as a kind of positive side-effect of introducing preservation technology. This Section summarizes some ideas on how such side-effects can be created in the Active System based on the ForgetIT approach.

5.2.1 Forgetful Information Presentation

The functionality for Managed Forgetting & Appraisal and the related Content Value Assessment (see Section 3.1.2) can be used to distinguish between content, which is of current importance from other content (see discussion above for MB). The values of MB can be used in the Active System to bring the content currently important ”closer to the user”, e.g. by showing it on the desktop or in special lists, having it on mobile devices, or by preferring search results with high MB, the so-called *forgetful search*. The core idea is to ease access to the things that are currently important: this is related to one of the five characteristics of the PoF Reference Model (*brain-inspired*), as discussed in Section 2.3 where we describe the digital working memory. This would be very similar to the human working memory, thus helping the user to focus on current activities. Examples for implementations of such forgetful information presentation can be found in the deliverables of WP9, most notably in deliverable D9.3 [ForgetIT, 2014e].

5.2.2 Forgetful and Archive-aware Search

Search is one of the core content access methods. Search & Navigation functionality (see Section 3.1.2) in the Active System can be affected in two ways by the introduction of preservation technology, as described in the following. The most obvious way is to smoothly integrate the archived content into the Active System search functionality, i.e. archive-aware search. This idea has already been discussed in more detail in Section 3.1.2, within the Active System extensions in support of archive-aware search have to be implemented. As a second way of modifying search in order to benefit from the ForgetIT approach, forgetful search (see above) can be introduced. The idea here is to take MB into account in the ranking function, thus preferring resources of relevance for the current task in the search result list.

5.2.3 Creating Awareness for Content Value

On a more conceptual level, the idea of content value assessment as it is used for computing MB and PV in managed forgetting can also be used to raise the awareness for the value of content assets. An improved understanding of the value of content, which is based on a variety of factors such as investment, usage, popularity, etc. can become an important building block for next generation Content Management System (CMS) applications. This idea is discussed in more detail in the deliverables of WP10.

5.3 Preservation Strategies

When preservation is introduced into the content management life-cycle of an Active System, a variety of decisions have to be taken defining the preservation strategy to be used. This includes decisions about when to preserve and about the granularity of preservation. Furthermore, the interaction between resource versioning and the preservation of a resource has to be defined.

Preservation actions can be triggered in different ways. They can for example be activated by the content management life-cycle: resources might be considered for preservation, when they go out of active use (low MB) or already upon creation or import into the system (e.g. for very valuable resources).

Furthermore, preservation can be scheduled, for example, by queuing all resources above a predefined PV threshold for preservation on a regular basis. Finally, it is of course also possible to manually trigger preservation actions for individual resources or resource collections.

It depends on the type of resources considered as well as on the level of control the user wants (or needs) over the preservation process, which of the options are best suited for an Active System under consideration. The options chosen influence the way the preservation process is integrated into the Active System beyond enabling the transfer of content to be archived.

Besides deciding *when* to preserve, it is also necessary to decide *what* to preserve. This can be considered along two related dimensions. First, it is possible to either preserve individual resources or entire collections of resources (or other types of complex objects such as sets of related resources) as one unit of archival. Second, resources can be preserved in isolation or together with context, which describes them. This second point is closely related to the work on contextualization in WP6 and the results affect the definition of the archival objects, the basic units of the PoF Information Model (see Section 3.2).

The choices with respect to granularity of preservation has consequences for the transfer protocols between Active System and DPS. In addition, it might require methods for selecting (extracting/collecting) relevant context information for a resource to be preserved.

An interesting further aspect of the preservation strategy is to think about the model of co-existence between the copy of the resource in the archive and the resource in the Active System (if the strategy allows for such a co-existence). This has some implications, when the resource is changed in the Active System, after a copy has been archived. Typically, newer versions will not overwrite the archived version. Rather, the changed version will be archived as a new version. There are, however, decisions to be made about, if and when the updated version is put (automatically) into the archive. As already stated above, this will provide input to the definition of the PoF Information Model in deliverable D8.5 [ForgetIT, 2016].

6 Mapping to OAIS and Preservation Management

6.1 The OAIS Reference Model

The Reference Model for an Open Archival Information System (OAIS) is a specification of the Consultative Committee for Space Data Systems (CCSDS), describing *an archive, consisting of an organization of people and systems, that has accepted the responsibility to preserve information and make it available for a Designated Community* [CCSDS, 2012]. OAIS specification includes a functional model and an information model. The OAIS functional model is depicted in Figure 16 . OAIS functional entities include *Ingest*, *Access*, *Data Management*, *Administration*, *Preservation Planning* and *Archival Storage*.

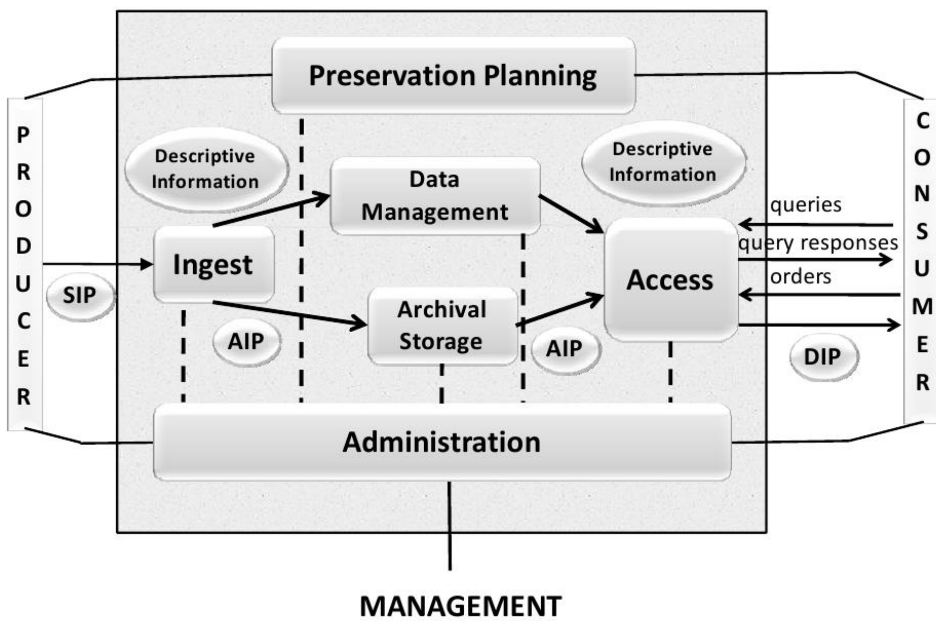

Figure 16: OAIS Functional Entities [CCSDS, 2012].

A brief description of the purpose of these functional entities follows, for a full description please refer to the CCSDS Recommended Practice 650.0-M-2 [CCSDS, 2012]. Starting from the information producer side in Figure 16, the *Producer* sends in a Submission Information Package (SIP) to the *Ingest* function. This function is responsible for receiving packages and re-packaging them into an Archival Information Package (AIP). The AIPs have to be complete in order to facilitate preservation, while the SIP can lack some information that already is in the archive, or that only the archive can add. The AIP is then sent to *Archival Storage*, and descriptive information is sent to *Data Management*. The *Archival Storage* function handles permanent storage of AIPs including error checking and media replacement procedures. The *Data Management* function is responsible for maintaining both descriptive information about the archive holdings, as well as system related data and other administrative data needed for managing the archive. The *Access* function then uses the Data Management function to provide query results to the *Consumer*, and request the actual AIP(s) from the Archival Storage function in order to prepare a Dissemination Information Package (DIP) in accordance with what the *Consumer* requested. In order to be prepared for changes in requirements from the environment, the *Preservation Planning* function monitors the relevant user communities and technology changes and recommends different actions to be taken. The recipient of these recommendations, and main responsible function for establishing policies and standards in the archive is the *Administration* function. This function also handles negotiations with information producers and consumers, and configuration management of the system. In the next Section, relationships between these entities and the PoF entities will be described.

6.2 Relationship with OAIS Reference Model

In Table 3, we describe the relationship between the functional entities of the two models, the PoF Reference Model and the OAIS Reference Model. For OAIS we refer to the last approved OAIS specification [CCSDS, 2012].

Functional Entity	**OAIS Functional Entity**	**Relation**
Exchange support, Contextualization, Re-contextualization	Ingest (p.4-5), Access (p.4-16)	Supports smooth bi-directional transfer of archival objects from Active System to DPS and vice versa. Creates submission information packages according to DPS submission agreements. The packages are adjusted to metadata and structure of information packages requirements. The Contextualization entity adds use context metadata to submission packages. At re-activation of archival objects the Re-contextualization entity adds information if there is a lack of sufficient context information. Exchange support is also responsible of restructuring of archival objects received from DPS on advice from Context-aware Preservation Management.
Context-aware Preservation Management	Administration (p.4-11)	Verifies that the content in submissions is according to DPS agreement. Supports the Audit Submission function that verifies that archival object packages meet the specifications of submission agreements as part of Administration functional entity.

Context-aware Preservation Management	Preservation planning (p.4-14), Administration (p.4-11)	Activity monitoring; collects information related to exchange of archival objects between Active System and DPS. Information related to use of content such as file format, system information, physical and logical content structure. Information submitted in the archival object transferred to the DPS intended as input to Preservation Planning and Administration entities.
Context-aware Preservation Management	Preservation planning (p.4-14), Administration (p.4-11)	Statistical computation of collected information from bi-directional exchange of archival objects between Active System and DPS. Information such as use of file format(s), system information, physical and logical content structure. The monitor of many-to-many relation between Active Systems and DPSs provides a federated Technology watch function supporting the Preservation planning and Administration functional entities.
Context-aware Preservation Management	Administration (p.4-11)	Supporting preservation decisions on how much should be invested in the preservation of a resource. This is done by matchmaking of digital objects content value and purpose of use to configuration of preservation services. Provides input to Manage System Configuration function in the Administration functional entity.
Search & Navigation	Access (p.4-16), Federation of archives (p.6-4), Distributed Access Aid (p.6-5)	The PoF framework acts as a federation of DPSs in providing search capabilities for an Active System. The search service provides the ability to search in a combination of content from both the Active System and DPS(s), as a joint virtual information space. The ranking of search results from the DPS(s) is based on the definition of Content Value for each archival object.
Managed Forgetting	Administration (p.4-11)	The results from content value assessments are input to the Administration entity in the DPS. This information is basis for update actions in the DPS.

Table 3: Mapping between PoF Reference Model and OAIS Functional Entities in the OAIS Specification (CCSDS Magenta Book) [CCSDS, 2012].

6.3 Relationship with PAIMAS Reference Model

The Producer-Archive Interface Methodology Abstract Standard (PAIMAS) [CCSDS, 2004] provides method description supporting the interaction between Producer and Archive. PAIMAS is divided in four different phases: `Preliminary Phase`, `Formal Definition Phase`, `Transfer Phase`, and `Validation Phase` as depicted in Figure 17.

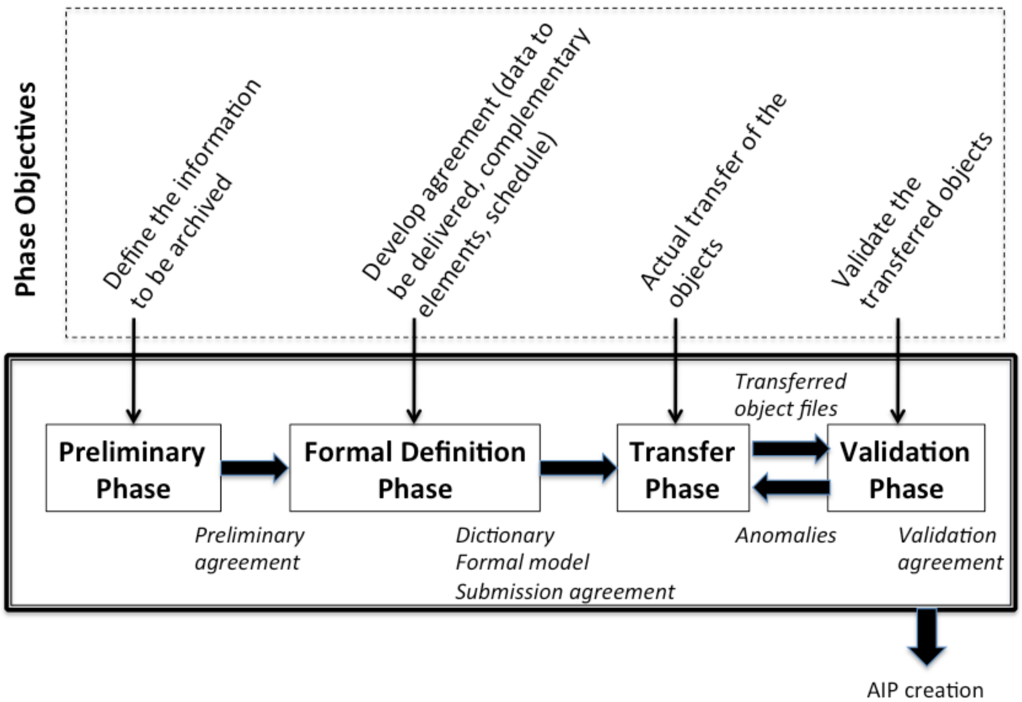

Figure 17: PAIMAS Main Phases [CCSDS, 2004].

The PoF Reference Model is responsible for defining a submission agreement containing the digital information objects and metadata to be submitted from the information Producer to an Archive and for effectively packaging the objects in the form of Submission Information Packages (SIPs). There can be several ”submission agreements” between a Producer and an Archive, and these different agreements cover different and independent sets of information. Previous agreements can be used to guide the completion of a new submission agreement. The PoF method is influenced by the approach defined in the recommended standard for Producer-Archive Interface Specification (PAIS) [CCSDS, 2014]. PAIS is a standard method for formally defining the digital information objects to be transferred by an information Producer to an Archive and applies specifically to the implementation of the main part of the Formal Definition Phase and the Transfer Phase in PAIMAS [CCSDS, 2004]. PAIS provides an abstract syntax for describing how these data will be aggregated into packages for transmission and one concrete implementation for the packages based on the XML Formatted Data Unit (XFDU) [CCSDS, 2008] standard.

The Context-aware Preservation Management entity supports the `Preliminary Phase` by identification of suitable DPS for preserving the significant properties of Producers information objects within acceptable cost limits associated with the storage needs. This is the basis for deciding the feasibility of a PoF solution according to the Producer archival needs.

In the Formal Definition Phase, the PoF components support the following activities:

- Agree upon the content and structure of the collection to be transferred from AS to PoF.
- Define expected file formats, expected instances, migration needs, and maximum volume of each transferred collection.
- Agree on descriptive metadata and its structure.
- Define level of provenance support by PoF.
- Define if and to what extent metadata should be added by Active System and PoF components.
- Adapt the SIP content, structure and use of metadata to receiving DPS.
- Define the communication operations as the use of interoperability services and security conditions.
- Agree on procedures for triggering a transfer, rejection, re-transfer, and transfer collection acceptance.
- Determine storage service configuration based on identified conditions as location restrictions, security, performance, and cost.

The output from this phase ends in a submission agreement stated in a Manifest used as template for configuration of the PoF process.

6.4 Evolution and Value creation in the Preservation Store

Storage plays a key role in any preservation system. Nowadays, much of the worlds data resides in storage clouds, both public and private. Storage clouds are highly distributed and scalable, and they offer high availability, high performance, and huge capacities at low costs. However, storage services tend to focus on storage and data services, while being agnostic to the content of the data they service. To increase the value of stored information over time, there is a need for the storage services to have some understanding of the content of the data and perform automatic processes on it, some of them periodically. For instance, in order to maintain the logical preservation of the stored data, the storage service could periodically check whether it needs to perform transformations on the data from obsolete formats to updated formats.

Data analytics is an umbrella term covering a variety of techniques and approaches in different domains for inspecting, transforming, and modeling data in order to discover hidden value, such as information which is not easily or directly accessible. Data analytics continues to gain momentum and several open source platforms are available. The approach described above, where storage services perform processes close to data, enables analysis of big data in the cloud. Cloud analytics is offered by all big data players, including IBM, Google and Amazon, just to name a few.

The Preservation Store in ForgetIT architecture is based on cloud, as discussed in deliverables D8.1 [ForgetIT, 2014c] and D7.1 [ForgetIT, 2013d]. The DPS is made up of a Digital Repository and a Cloud-based Preservation-aware Storage System, as described in Section 4. In ForgetIT, a new mechanism based on *storlets* [Rabinovici-Cohen et al., 2014] will be investigated, to implement long term digital preservation in the cloud within the DPS [Rabinovici-Cohen et al., 2013, Rabinovici-Cohen et al., 2008]. Storlets can help increasing the value we get out of storage and the speed at which we can access what we need, in a nutshell: getting more value from data, much faster. A storlet is code that runs in the storage cloud. The main idea is to run the code close to where the data is stored, so as to minimize costs of data transfer when operating in the cloud. Storlets are typically used to transform the data, filter the data, or analyze the data, all in the storage cloud. Storlets are an ideal way to introduce new services. Storlets could also be used for automated creation of data-object metadata. This metadata could be used to facilitate the retrieval of preserved information back into active use, even if a long time has passed since it was archived. Additional information can be found in WP7 deliverables.

7 Summary and Future Work

In this deliverable we described the first version of the PoF Reference Model, based on the ForgetIT novel approach to digital preservation inspired by human forgetting and remembering. The model described here aims to provide the basic terminology and concepts inspired by core ForgetIT principles: synergetic preservation, managed forgetting and contextualized remembering. The model also supports the implementation of a forgetful approach to preservation in the PoF Framework.

The foundations of the PoF Reference Model (Section 2) are identified by five characteristics: integrative, value-driven, brain-inspired, forgetful and evolution-aware. Each characteristic has been discussed and the relationship to the relevant work package has been also clarified. Following the OAIS specification approach [CCSDS, 2012], we have first defined a Functional Model (Section 3.1) , with several functional entities and workflows. Some preliminary ideas for the ForgetIT Information Model (Section 3.2) are also included, although further analysis is required, as discussed in the following.

The functional part of the PoF Reference Model is made up of three layers: Core Layer, Remember & Forget Layer and Evolution Layer. For each layer we described the main features and the identified functional entities and for each entity we described the role within that layer. Different workflows have been also defined, for preservation preparation, re-activation, situation change and setting change. For each workflow we described the involved model entities. Such workflows will be further evaluated and refined in the final version of the model.

The ForgetIT model does not intend to replace the OAIS model, but rather to go beyond to support the novel approach of the project. In the document we explained how the PoF Reference Model can interact with OAIS compatible approaches and then described the relationship between the functional entities of the two models (Section 6), the PoF Reference Model and the OAIS Reference Model.

The PoF Framework integrates the results of the different WPs and the first integrated prototype described in deliverable D8.3 [ForgetIT, 2014d] is strongly inspired by the concepts of the PoF Reference Model. The new releases of the framework will further implement the model and will be used for validating it. In the document we provided a mapping between the functional entities and the PoF Framework architecture (Section 4), based on the components described in D8.1 [ForgetIT, 2014c] . Moreover, for each aforementioned workflow we described the role of the framework components, mainly for what concerns the PoF Middleware, which includes the most relevant part of the new ForgetIT approach. In this way we have shown that the ForgetIT architecture is compliant to the PoF Reference Model.

Finally, we included some preliminary considerations related to the extension of the current information management systems and to the possible impact of introducing the preservation into the information management workflows (Section 5). This is related to one of the core ForgetIT principles, namely the synergetic preservation: enabling a smooth transition between active use and preservation by making intelligent preservation processes an integral part of the content life-cycle in information management and by developing solutions for smooth bi-directional transitions. We described possible extensions of the Active System to enable interaction with the DPS, describing information exchange, evidences for information value and contextualization support. This is also related to creating immediate benefit from the preservation, not only on the long run, and can drive the preservation strategies in terms of choosing the preservation objects and managing with the co-existence of the resources in the DPS and in the Active System.

In the following Sections we briefly discuss the assessment of the results presented here according to the WP8 performance indicators and then describe the plan for future work.

7.1 Assessment of Performance Indicators

The expected WP8 outcomes, reported in the project proposal, are:

- the Preserve-or-Forget (PoF) Reference Model
- the PoF Framework

The results described here for the first framework prototype refer to outcome 1, for which the following success/progress indicators have been identified in the project proposal:

1. *availability of reference models and degree of adoption in the different contexts covered by each work package*
2. *availability of adequate "formats" for content and metadata*

The model described here represents the first release of the PoF Reference Model. The results achieved so far are compatible with the expected progress and success indicators for WP8, although further investigation is requested within the project towards the final model release, as described below.

Indicator 1: Reference Model

The PoF Reference Model has been created from scratch collecting the inputs from all work packages, taking into account the recommendations after the first project review, mainly for what concerns the role of OAIS in the PoF Framework and the need to go beyond such model to support the novel ForgetIT approach to preservation. The model reflects the common knowledge and the new ideas of all partners to represent the ForgetIT approach in a coherent framework based on a reference model. The reference model includes a functional model and an information model. The functional model has been described in detail and can be considered almost complete. Further details could be still added in the future, but the three proposed layers cover all aspects of ForgetIT approach to digital preservation. The information model is still under definition and requires further investigation within the project to propose a complete and coherent definition of the preservation objects, their interaction and evolution within the PoF Framework, although the building blocks of the information objects have already been identified and suitable technologies have been identified for the first framework prototype in D8.3. The PoF Framework architecture described in D8.1 is compliant to the model, although some modifications are still possible when the information model is available.

Indicator 2: Content and Metadata Formats

The information model is still under definition. When ready, it will be also used to define requirements on content and metadata formats required to represent the preservation objects discussed in this document. Nevertheless, the project has already released the first integrated prototype, described in D8.3, and has identified several content and metadata formats and the associated technologies to properly represent the core concepts of the information model described here. For example the adoption of the CMIS standard enables the content and metadata exchange between the user applications and the framework, while the standards used for content packaging and archival (such as METS, DublinCore, PREMIS and others) are suitable to support the representation of contents, their exchange between user applications and preservation systems, their evolution over time (see deliverables D8.1 and D8.3). The definition of the archival units and the most suitable data model in the PoF Framework requires further investigation.

7.2 Future Work

For the initial version of the PoF Reference Model, we focused mainly on the functional model. The definition of a ForgetIT information model is also planned. In this new information model we will provide a representation of the archival entities in the context of ForgetIT, identifying the most appropriate structure and describing the interaction and evolution of the information objects with the PoF Framework. The information model requires further investigation within the project, in parallel with the development of the reference platform and the availability of the prototype deliverables from technical WPs.

The final release of the PoF Reference Model is expected at M36, and will be reported in deliverable D8.5 [ForgetIT, 2016]. The second release of the PoF Framework, described in D8.4 [ForgetIT, 2015c], will integrate further project components and will provide additional features based on the model described here, mainly for what concerns the development of the PoF Middleware, the integration with the Active Systems and the DPS and the support for all the workflows described here.

References

[Afrasiabi Rad et al., 2014] Afrasiabi Rad, P., Nilsson, J., and Päivärinta, T. (2014). *Administration of Digital Preservation Services in the Cloud Over Time: Design Issues and Challenges for Organizations*. Proceedings of the international conference on cloud security management. Academic Conferences and Publishing International Ltd.

[Andersson et al., 2014] Andersson, I., Lindqvist, G., and Randers, F. (2014). Preservation services planning: A decision support framework.

[Bote et al., 2012] Bote, J., Fernandez-Feijoo, B., and Ruiz, S. (2012). The cost of digital preservation: A methodological analysis. *Procedia Technology*, 5:103–111.

[CCSDS, 2004] CCSDS (2004). Producer-Archive Interface Methodology Abstract Standard (PAIMAS) - Recommended Practice, CCSDS 651.0-M-1 (Magenta Book) Issue 1. Equivalent to ISO 20652:2006. `http://public.ccsds.org/publications/archive/651x0m1.pdf`. Retrieved 30 November 2014.

[CCSDS, 2008] CCSDS (2008). XML Formatted Data Unit (XFDU) - Recommended Standard, CCSDS 661.0-B-1 (Blue Book) Issue 1. Equivalent to ISO 13527:2010. `http://public.ccsds.org/publications/archive/661x0b1.pdf`. Retrieved 30 November 2014.

[CCSDS, 2012] CCSDS (2012). Reference Model for an Open Archival Information System (OAIS) - Recommended Practice, CCSDS 650.0-M-2 (Magenta Book) Issue 2. Equivalent to ISO 14721:2012. `http://public.ccsds.org/publications/archive/650x0m2.pdf`. Retrieved 30 November 2014.

[CCSDS, 2014] CCSDS (2014). Producer-Archive Interface Specification (PAIS) - Recommended Standard, CCSDS 651.1-B-1 (Blue Book) Issue 1. Currently ISO/DIS 20104. `http://public.ccsds.org/publications/archive/651x0m1.pdf`. Retrieved 30 November 2014.

[Chappell, 2004] Chappell, D. (2004). *Enterprise Service Bus*. O'Reilly Media, Inc.

[ForgetIT, 2013a] ForgetIT (2013a). Deliverable D3.1: Report on Foundations of Managed Forgetting.

[ForgetIT, 2013b] ForgetIT (2013b). Deliverable D5.1: Foundations of Synergetic Preservation.

[ForgetIT, 2013c] ForgetIT (2013c). Deliverable D6.1: State of the Art and Approach for Contextualization.

[ForgetIT, 2013d] ForgetIT (2013d). Deliverable D7.1: Foundations of Computational Storage Services.

[ForgetIT, 2014a] ForgetIT (2014a). Deliverable D2.2: Foundations of Forgetting and Remembering – Preliminary Report.

[ForgetIT, 2014b] ForgetIT (2014b). Deliverable D5.2: Workflow Model and Prototype for Transition between Active System and AIS.

[ForgetIT, 2014c] ForgetIT (2014c). Deliverable D8.1: Integration Plan and Architectural Approach (v2).

[ForgetIT, 2014d] ForgetIT (2014d). Deliverable D8.3: The Preserve-or-Forget Framework – First Release.

[ForgetIT, 2014e] ForgetIT (2014e). Deliverable D9.3: Personal Preservation Pilot I: Concise Preserving Personal Desktop.

[ForgetIT, 2015a] ForgetIT (2015a). Deliverable D2.3: Foundations of Forgetting and Remembering - Intermediary Report.

[ForgetIT, 2015b] ForgetIT (2015b). Deliverable D3.3: Strategies and Components for Managed Forgetting - Second Release.

[ForgetIT, 2015c] ForgetIT (2015c). Deliverable D8.4: Preserve-or-Forget Framework – Second Release.

[ForgetIT, 2016] ForgetIT (2016). Deliverable D8.5: The Preserve-or-Forget Reference Model – Final Model.

[Harvey, 2006] Harvey, R. (2006). Appraisal and selection. In Ross, S. and Day, M., editors, *DCC Digital Curation Manual*.

[Harvey and Thompson, 2010] Harvey, R. and Thompson, D. (2010). Automating the appraisal of digital materials. *Library Hi Tech*, 28(2):313–322.

[Hohpe and Woolf, 2003] Hohpe, G. and Woolf, B. (2003). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Kejser et al., 2011] Kejser, U. B., Nielsen, A. B., and Thirifays, A. (2011). Cost model for digital preservation: Cost of digital migration. *IJDC*, 6(1):255–267.

[Marshall, 2011] Marshall, C. C. (2011). Challenges and opportunities for personal digital archiving. *Digital: Personal Collections in the Digital Era*, pages 90–114.

[OASIS, 2013] OASIS (2013). Content Management Interoperability Services (CMIS) Version 1.1. OASIS Standard. `http://docs.oasis-open.org/cmis/CMIS/v1.1/CMIS-v1.1.html`. Retrieved 30 November 2014.

[Rabinovici-Cohen et al., 2008] Rabinovici-Cohen, S., Factor, M. E., Naor, D., Ramati, L., Reshef, P., Ronen, S., Satran, J., and Giaretta, D. L. (2008). Preservation DataStores: New Storage Paradigm for Preservation Environments. *IBM Journal of Research and Development*, 52(4):389–399.

[Rabinovici-Cohen et al., 2014] Rabinovici-Cohen, S., Henis, E., Marberg, J., and Nagin, K. (2014). Storlet Engine for Executing Biomedical Processes within the Storage System. In *Proceedings of the 7th International Workshop on Process-oriented Information Systems in Healthcare*, ProHealth '14.

[Rabinovici-Cohen et al., 2013] Rabinovici-Cohen, S., Marberg, J., Nagin, K., and Pease, D. (2013). PDS Cloud: Long Term Digital Preservation in the Cloud. In *Proceedings of the 2013 IEEE International Conference on Cloud Engineering*, IC2E '13, pages 38–45, Washington, DC, USA. IEEE Computer Society.

[Rauch et al., 2013] Rauch, J., Braud, M., Edelstein, O., Rabinovici-Cohen, S., Nagin, K., Marberg, J., Voets, D., Sanya, I., Badawy, M., Shebab, E., Randers, F., Droppert, J., and Klecha, M. (2013). Ensure: Long term digital preservation of health care, clinical trial and financial data. In Borbinha, J., Nelson, M., and Knight, S., editors, *iPRES 2013: Proceedings of the 10th International Conference on Preservation of Digital Objects*.

Glossary

AIP Archival Information Package. 41, 42

CCSDS Consultative Committee for Space Data Systems. 41

CMS Content Management System. 39

DIP Dissemination Information Package. 42

DPS Digital Preservation System. 6–11, 16–25, 27, 28, 30, 37, 38, 40, 42, 43, 46, 49

EAI Enterprise Application Integration. 32

EIP Enterprise Integration Patterns. 32

MB Memory Buoyancy. 22, 24, 38–40

MOM Message Oriented Middleware. 32

OAIS Open Archival Information System. 41

PAIMAS Producer-Archive Interface Methodology Abstract Standards. 23, 44

PAIS Producer-Archive Interface Specification. 44

PIMO Personal Information MOdel. 38

PoF Preserve-or-Forget. 6, 7, 17, 30, 31, 34, 45, 47, 48

PV Preservation Value. 22, 24, 38–40

SIP Submission Information Package. 41

UML Unified Modeling Language. 8

XFDU XML Formatted Data Unit. 44