www.forgetit-project.eu

ForgetIT

Concise Preservation by Combining Managed Forgetting and Contextualization Remembering

Grant Agreement No. 600826

Deliverable D5.3

Work-package	WP5: Joint Information and Preservation Management
Deliverable	D5.3: Workflow model and prototype for transition between active system and AIS – Second release
Deliverable Leader	Jörgen Nilsson
Quality Assessor	--
Estimation of PM spent	18 PM
Dissemination level	PU
Delivery date in Annex I	M24
Actual delivery date	17/04/2015
Revisions	7
Status	Final
Keywords:	Preservation, ingest, re-activation

Disclaimer

This document contains material, which is under copyright of individual or several ForgetIT consortium parties, and no copying or distributing, in any form or by any means, is allowed without the prior written agreement of the owner of the property rights.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the ForgetIT consortium as a whole, nor individual parties of the ForgetIT consortium warrant that the information contained in this document is suitable for use, nor that the use of the information is free from risk, and accepts no liability for loss or damage suffered by any person using this information.

This document reflects only the authors' view. The European Community is not liable for any use that may be made of the information contained herein.

Revision History

Version	Major changes	Authors
0.1	Overall insertion of material in all sections	JN, IA
0.2	Major revision of text and some alteration of structure	JN
0.3	Addition of content in all sections	IA, PAR, JN
0.4	Expanded description of workflows + updated figures	IA
0.5	Finalised "Big picture"	JN
0.6	Completion of Implementation section	GL, IA, JN
0.7	Harmonisation and clean up	CN

List of Authors

Partner Acronym	Authors
LTU	Ingemar Andersson, Jörgen Nilsson, Göran Lindqvist, Parvaneh Westerlund

Table of Contents

Acronyms

AIP Archival Information Package

AIS Archival Information System

API Application Programming Interface

AS Active System

CMIS Content Management Interoperability Services

CaPM Context-aware Preservation Manager

DIP Dissemination Information Package

DO Digital Object

DoW Description of Work

DPS Digital Preservation System

DROID Digital Record Object Identification

ESB Enterprise Service Bus

FTR Format Transformation Rule

IS Information System

METS Metadata Encoding and Transmission Standard

MODS Metadata Object Description Standard

OAIS Open Archival Information System

OAI-PMH Open Archives Initiative – Protocol for Metadata Harvesting

PASS Preservation-Aware Storage System

PIMO Personal Information MOdel

PoF Preserve-or-Forget

REST Representational State Transfer

SD Semantic Desktop

SIP Submission Information Package

SP Storage Provider

WP Work Package

Executive summary

This deliverable summarizes the current status of the components for enabling a smooth transition between the active system and the preservation system in the ForgetIT project. The core components considered for this purpose are the Collector/Archiver and the context-aware Preservation Manager (CaPM). The Collector/Archiver is directly involved in the transfer and exchange activities, whereas the CaPM monitors and manages such processes on a meta-level.

Work described in this deliverables builds upon the approach and workflows defined in D5.1 as well as on the conceptual processes defined in D8.2 for the PoF Reference Model.

The further development of the component presented in this deliverable, especially of the CaPM will be reported in deliverable D5.4.

1 Introduction

The ForgetIT project aims at helping people and organisations with decisions on what to preserve and where it should be kept, through ideas from psychology (on human memory) and with assistance of automated processes. In order to make this as transparent as possible to the users, there is a need of smooth transition of objects between the active systems (the systems that are in use by the users) and the preservation systems (the system[s] where material that should be preserved is kept). This work package (WP) describes these workflows and implements some of the functionality needed for the automated processes.

In previous WP5 deliverables, we identified a gap between content management systems and preservation systems especially regarding support for ingest of objects (D5.1) [ForgetIT, 2013], and we also modelled the first iteration of two workflows, one for pre-ingest and ingest, and one for re-contextualization and access (D5.2) [ForgetIT, 2014b]. D5.2 also included discussion, documentation and reasoning on the first prototype versions of the software components and the message oriented middleware approach.

During the work in the project, especially around the reference model (D8.2) [ForgetIT, 2015a], there have been some changes in terminology. In this report it will mainly be visible in the labelling of the updated versions of the workflows. A quick legend follows:

Pre-ingest and ingest -> Preservation Preparation

Access and re-contextualization -> Re-activation

Tapping into these workflows is the Context-aware Preservation Manager (software component) that acts upon agreements between the users and the service providers, monitoring what goes in and out of the systems, upholds agreements, and suggest actions to be taken e.g. in re-activation of material.

It has to be noted that this report is just part of deliverable D5.3. The rest of the deliverable are the implemented components described in this report, which have been integrated into the PoF Framework.

1.1 Structure of the Report

After the Introduction, a "Big Picture" section describing circumstances related to High-level Workflows and Integration Considerations follows. The report then continues with section 3 describing current workflow descriptions for Preservation preparation and Re-activation and what has changed since the previous release. Section 4 describes in some detail the implementation done so far on components and middleware related to WP5, followed by Summary and Future Work. Appendix A and B contains class descriptions of the components from section 4.

1.2 Target Audience

The Introduction section, Big Picture section, and the Summary should be of interest to practitioners and researchers in the area of preservation and information management, who are interested in the ForgetIT approach and solution as it provides a brief overview of the structure of the framework built in the project and where the components described in this report fit in. The Workflow section and Implementation section are intended for a more technical readership with interest in more of the details surrounding the work.

2 The Big Picture

To describe where the work reported in this report fit in to the rest of the project, an overview of the role of the components and the workflows is provided here.

This work package is responsible for two components in the Preserve-or-Forget (PoF) middleware, as well as the PoF Adapters in the Active Systems. The overall architecture can be seen in Figure 1 with the relevant component highlighted.

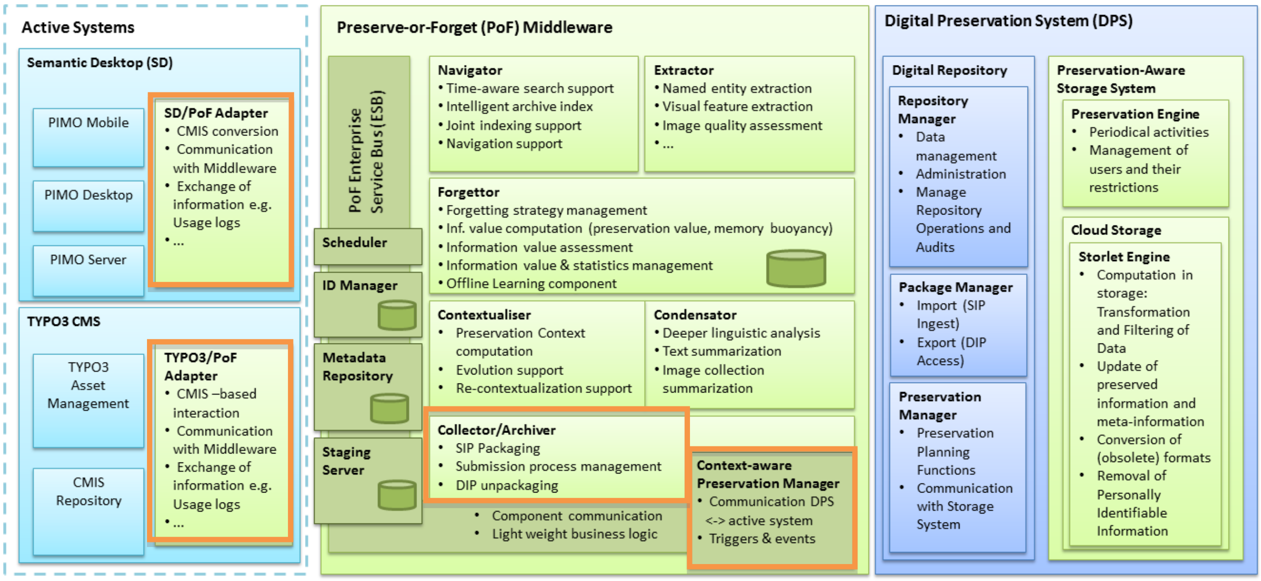
Active Systems
Semantic Desktop (SD)
PIMO Mobile
PIMO Desktop
PIMO Server
SD/PoF Adapter
• CMIS conversion
• Communication with Middleware
• Exchange of information e.g. Usage logs
• ...
TYPO3 CMS
TYPO3 Asset Management
CMIS Repository
TYPO3/PoF Adapter
• CMIS –based interaction
• Communication with Middleware
• Exchange of information e.g. Usage logs
• ...
Preserve-or-Forget (PoF) Middleware
PoF Enterprise Service Bus (ESB)
Scheduler
ID Manager
Metadata Repository
Staging Server
Navigator
• Time-aware search support
• Intelligent archive index
• Joint indexing support
• Navigation support
Extractor
• Named entity extraction
• Visual feature extraction
• Image quality assessment
• ...
Forgettor
• Forgetting strategy management
• Inf. value computation (preservation value, memory buoyancy)
• Information value assessment
• Information value & statistics management
• Offline Learning component
Contextualiser
• Preservation Context computation
• Evolution support
• Re-contextualization support
Condensator
• Deeper linguistic analysis
• Text summarization
• Image collection summarization
Collector/Archiver
• SIP Packaging
• Submission process management
• DIP unpackaging
• Component communication
• Light weight business logic
Context-aware Preservation Manager
• Communication DPS <-> active system
• Triggers & events
Digital Preservation System (DPS)
Digital Repository
Repository Manager
• Data management
• Administration
• Manage Repository Operations and Audits
Package Manager
• Import (SIP Ingest)
• Export (DIP Access)
Preservation Manager
• Preservation Planning Functions
• Communication with Storage System
Preservation-Aware Storage System
Preservation Engine
• Periodical activities
• Management of users and their restrictions
Cloud Storage
Storlet Engine
• Computation in storage: Transformation and Filtering of Data
• Update of preserved information and meta-information
• Conversion of (obsolete) formats
• Removal of Personally Identifiable Information

Figure 1: The Collector/Archiver, the Context-aware Preservation Manager, and the Active System Adapters in the ForgetIT architecture

While Figure 2 depict an overall interaction between an Active System and a Digital Preservation System, including the need for exchange of administrative information, earlier deliverables, in particular *D8.3 The Preserve-or-Forget Framework* [ForgetIT 2014a], describe the communication between the Active System adapters and the middleware and its components. The communication adopts a REST[1] approach and the exchange of Digital Objects (DO) is handled using CMIS[2]. The communication between the Digital Preservation System (DPS) and the middleware also adopts REST, but not CMIS since that currently is not a common option in DPS solutions. One design issue worth noting is that the ForgetIT project works under the assumption that a customer (Active System owner) might use several digital preservation service providers, perhaps at the same time, but most certainly over time, and that the systems involved will change.

[1] Representational State Transfer

[2] Content Management Interoperability Services – http://docs.oasis-open.org/cmis/CMIS/v1.1/os/CMIS-v1.1-os.html

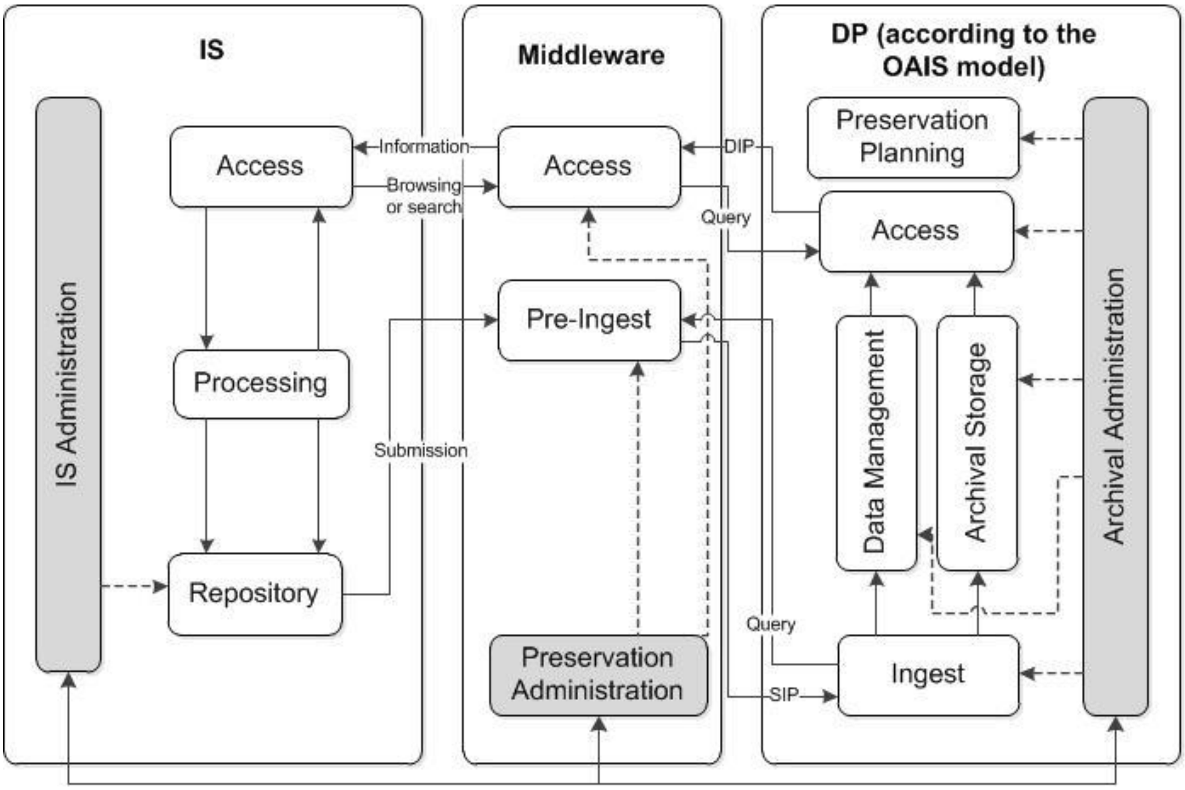

Figure 2: Model for interaction between Active System (IS) and Digital Preservation System (DP) [Afrasiabi, Nilsson, Päivärinta, 2014]

2.1 Collector/Archiver software component

The Collector/Archiver can be seen as the component that manages the basic flow of digital objects between the active systems and the preservation system(s). In general it can be seen as two parts, where in a typical preservation preparation scenario, the Collector is responsible for fetching/gathering objects that should be preserved from the Active System and the Archiver is responsible for packaging them in a suitable way, thereafter transferring them to the DPS. In a re-activation scenario, the roles would be similar, but the Collector would then work with the DPS and the Archiver would be responsible for extraction of the Dissemination Information Package (DIP) and making the objects available to the Active System.

2.2 Context-aware Preservation Manager component

The Context-aware Preservation Manager (CaPM) is responsible for management of administrative information from the PoF middleware process. This information is used in the preservation and re-activation process to enhance the usability of digital objects when brought back from a DPS to active use in an Information System (IS). This component is responsible for monitoring of changes in semantic ontologies in the IS, monitoring of use and change of physical and logical structures in IS, logging of changes in practices; capture the use of file formats and technologies. The CaPM will use this data in computation of change recommendations based on content value and use statistics and to propagate this information to the DPS that is responsible for keeping preserved objects usable. This component is also responsible for the establishment of a submission agreement, in hold of information that defines the terms and conditions of routes for different types of content acting as a semantic matchmaker between identified needs and provided digital preservation (DP) services.

2.3 Active System adapter

The active system adapter act as a bridgehead between the system that has need of preservation services and the PoF Middleware. The idea is that the middleware provides a number of standard-based interfaces for exchange of objects and that the adapter implements that interface on the Active System side. As already mentioned, the ForgetIT project employs the CMIS standard for exchange of objects, and the Active System adapters in the two different systems implement this in two slightly different ways. The TYPO3 case use a full-fledged Alfresco[3] server acting as a CMIS repository, and the Sematic Desktop (SD) case employ an Apache Chemistry[4] implementation. Both these implementations are then (loosely) integrated with the respective systems.

[3] Alfresco CMIS – http://www.alfresco.com/cmis

[4] Apache Chemistry – OpenCMIS http://chemistry.apache.org/java/opencmis.html

3 Workflow Descriptions

This section contains descriptions of the two main workflows related to seamless transition between active systems and preservation systems. These workflows are for *Preservation* Preparation (previously known as Pre-ingest and Ingest workflow), and *Re-activation* (previously known as the Access and Re-contextualization workflow).

3.1 Problem Statement

In this work package, one of the objectives, as stated in the Description of Work, is to achieve "Seamless transition between active and archival system – based on forgetting methods". This includes defining workflows that describe this transition with relevant steps and involvement of components. Two major workflows are described in this section, namely the Preservation Preparation workflow, and the Re-activation workflow.

3.2 Preservation Preparation Workflow

In deliverable D8.2, *The Preserve-or-Forget Reference Model* [ForgetIT, 2015a], the ideas and concepts behind the ForgetIT approach are gathered to an implementable model. There are different layers in the model, but in Figure 3 we see the Preservation Preparation Workflow in the *Remember & Forget Layer* since this provides ample detail for this section.

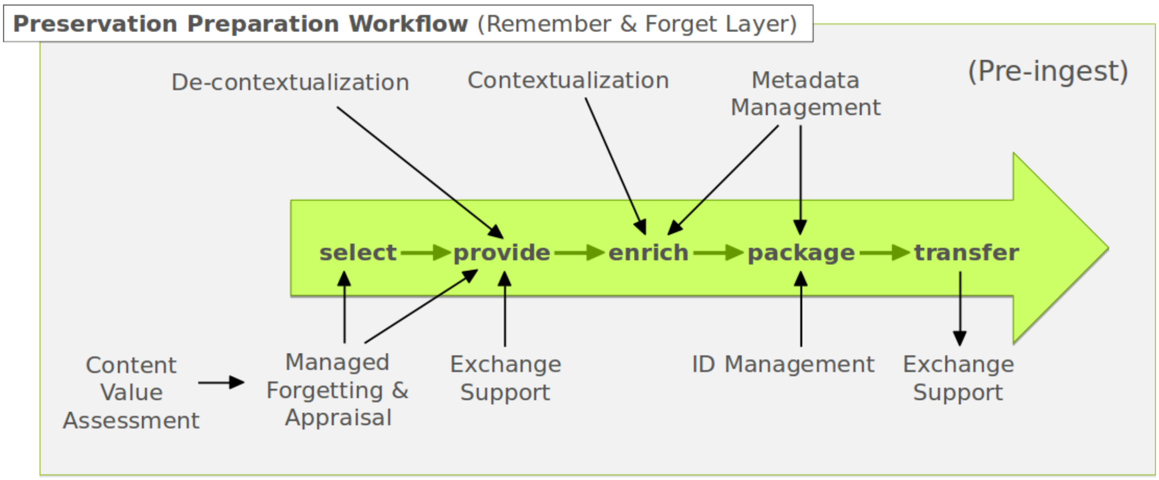

Figure 3: Preservation Preparation Workflow, reference model [ForgetIT, 2015a, p. 22]

The workflow has five basic steps, *select, provide, enrich, package,* and *transfer*. Although these steps are described in more detail in D8.2 [ForgetIT, 2015a], a brief description is in its place here. The *select* phase is about deciding what should be archived, and in this layer that is supported by *Managed Forgetting & Appraisal* that in turn is assisted by a *Content Value Assessment*. These two functionalities are thoroughly described in deliverable D3.3 *Strategies and Concepts for Managed Forgetting* [ForgetIT, 2015c].

The *provide* step is supported by the *Exchange Support* which essentially is the CMIS client and the Collector, described in more detail later in this report. This step is also supported by De-contextualization, which aims at getting enough context from the active system.

In the next step, *enrich*, the selected object is enriched by the *Contextualization* functionality, which is described in detail in deliverable D6.3 *Contextualisation Tools - Second Release: Updates to the Context Modelling* [ForgetIT, 2015b]. This provides semantic information that should improve both the discovery and reuse of the object. This information will be treated as metadata, and therefore there is also need for a *Metadata Management* functionality, which today mainly is used for the next step.

The *package* step is responsible for creating a suitable Submission Information Package (SIP) based on agreements between the active systems owner and the preservation system provider. The package step is supported by *Metadata Management* and *ID Management* for holding metadata that is needed for the creation of the package, and management of both local identifiers as well as identifiers received from e.g. the preservation system. The package step is largely covered by the *Package Module* described in section 4.2.

The functionality mentioned in Figure 3 is implemented in several different components in the PoF Middleware. Following is a short functional description of the PoF Middleware components that is interacting in the preservation preparation (pre-ingest) workflow depicted in Figure 4. The *Forgettor* is the component that assists in the appraisal process by assessments of short- and longterm value of information resources (digital objects) based on its actuality, usage frequency, age of object, related objects etc. The *Forgettor* is responsible for automatically triggering the start of the preservation preparation workflow. The *Collector/Archiver* component is responsible for upon request (a trigger), automatically fetch the digital items (content) and additional metadata as agreed upon from the active systems, in Figure 4 referred as the Information System (IS). The protocol in use for fetching data from the IS in the use case is the Content Management Interoperability Services (CMIS).

The *Collector/Archiver* is also responsible for assembling items and its metadata to a submission information package (SIP) according to a submission agreement with receiving digital preservation system (DPS). The data transfer protocol in use for transferring the SIP to the Ingest module in DPS is HTTP using the Representational State Transfer (REST) architecture style. The *Extractor* component has the ability to automatically extract data from digital items as text and images as named entity extraction from text, concept detection in images, and image quality assessments. This component is useful to automatically generate metadata and detecting images of poor quality. The *Condensator* is a component that executes advanced linguistic text and image analysis as text summarization, face detection and clustering. The *Contextualizer* component is responsible for enriching the digital items in the SIP with context metadata based on the output from the extractor and condensator components. If necessary it also fetch context metadata from external web resources.

In the preservation preparation workflow the *Context-aware Preservation Manager* (CaPM) component is responsible for verifying that the items to be fetched from the active system (IS) is not out of scope according to each submission agreement. CaPM also keeps track if physical and/or logical structure should be fetched, according to the agreement, and later used for adaptation of items in the re-activation process. This commitment also extends to keep track of change of ontologies. The CaPM component also monitor items passing through the PoF Middleware, gathering information about them which then can be compiled and used as statistics on the use of file formats. The *Metadata Repository* and the *Staging Server* components is a relational database management system and a file repository that temporarily stores metadata and keep track of all digital items in the PoF Middleware process.

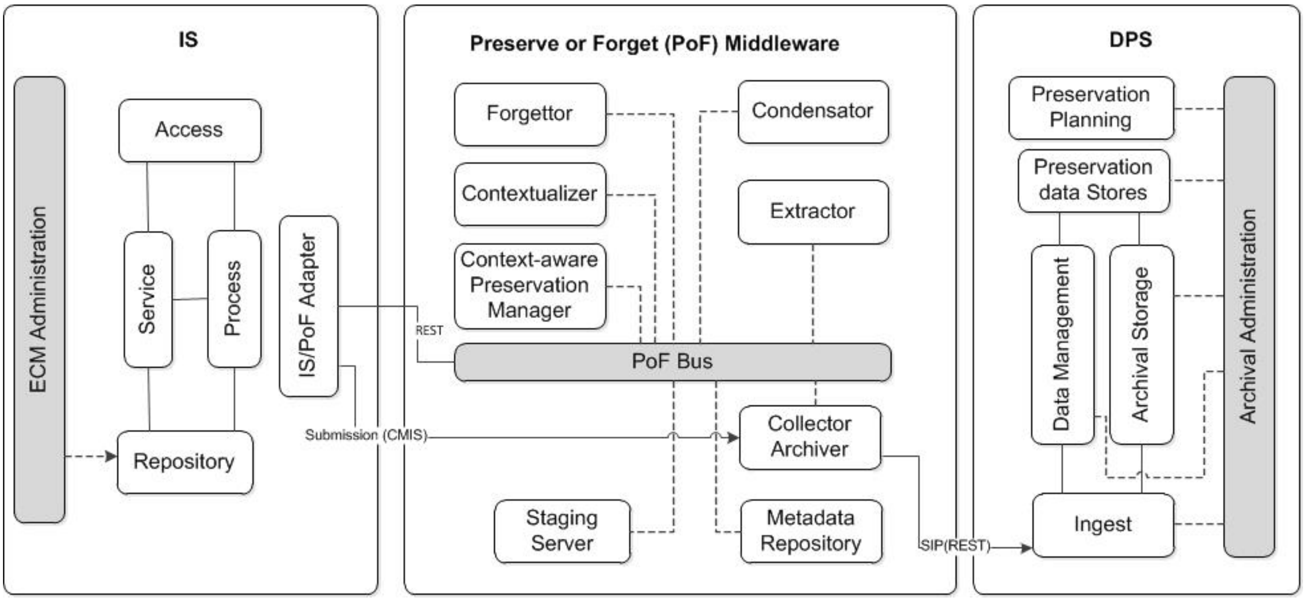

Figure 4: Preservation Preparation component architecture

3.2.1 Detailed description of Preservation Preparation workflow

This section contains a detailed description of the interaction between the major components of the preservation preparation workflow illustrated in Figure 5.

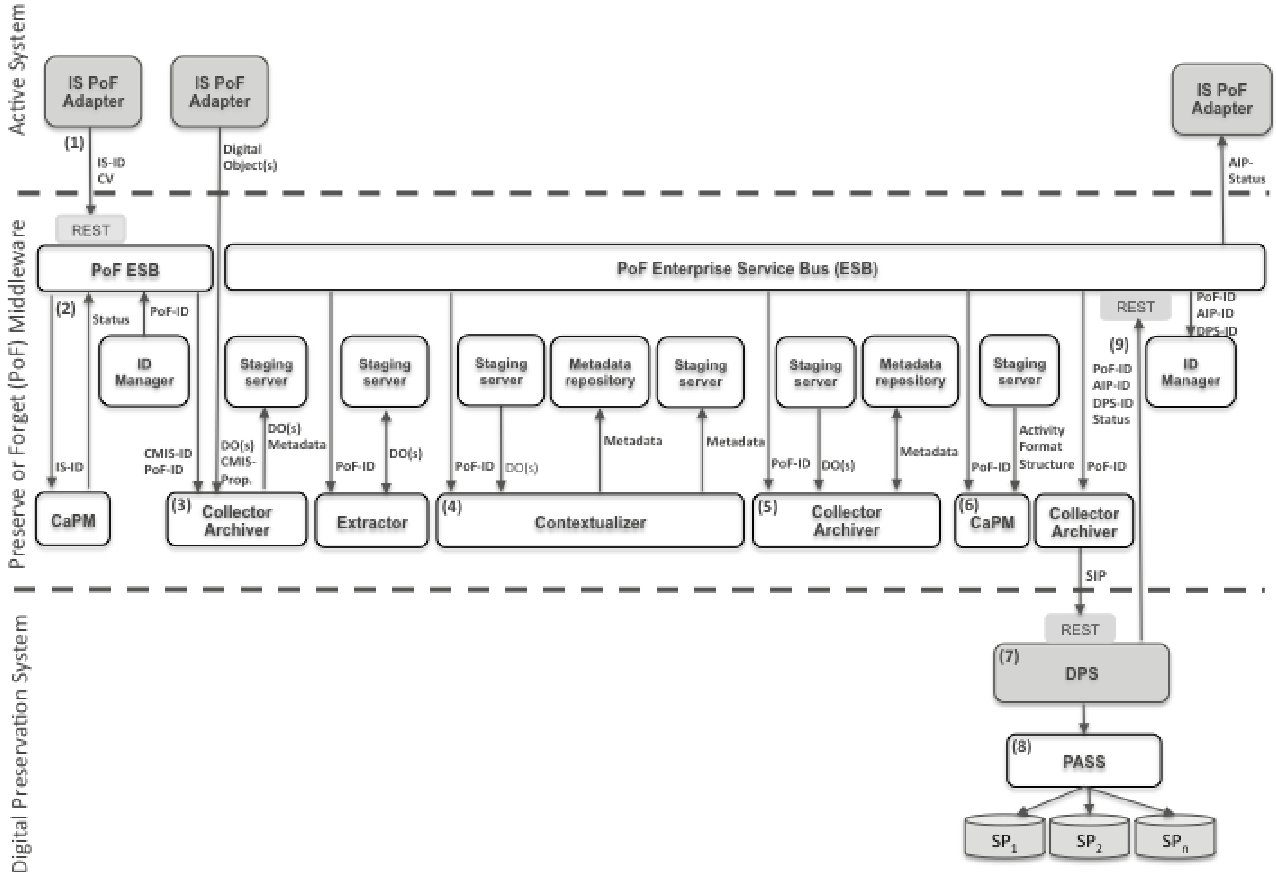

Figure 5: Preservation Preparation Workflow

[1] An IS triggers the pre-ingest process by calling the PoF enterprise service bus (ESB) submitting information that identifies the IS and the localization of content represented by the IS-ID. The content value (CV) is computed before the start of the PoF workflow and represents the contents long- and short-term relevancy represented by preservation value (PV) and memory buoyancy (MB), used as a trigger and affects how the content will be managed.

[2] The PoF enterprise service bus (ESB) receives the preservation request and sends the IS-ID forward to the context-aware preservation manager (CaPM). CaPM checks (assisted by the collector) if the content is within the scope of expected submission, before transfer of content to the PoF middleware. If there is a mismatch, the PoF request is rejected. The id-manager creates an internal globally unique identifier (PoF-ID), which keeps track of the content managed during PoF middleware process.

[3] The collector component receives a trigger from PoF ESB and start fetching content and information about the folder structure using CMIS as transfer protocol exposed by the IS PoF Adapter. The content is stored on the staging server in a folder structure identified by the PoF-ID.

[4] The contextualizer is responsible for the process of adding context information to an archival unit. This process is using the output from the extractor component, which automatically extracts information from the content and external sources such as DBpedia[5] and Wikidata[6]. The output from this process is saved in the metadata repository and staging server.

[5] The archiver starts the activities included in the process of creating a SIP using metadata standard as METS[7] and MODS[8]. The process includes fetching content from the staging server, file identification (DROID[9]), compute fixity checksums (MD5[10]), fetch metadata from the repository, ending with the creation of a SIP based on the eARD[11] specification.

[6] The Context-aware Preservation Manager (CaPM) saves persistent information from the activities in the PoF workflow. Information that contains data related to the workflow process as the information system id, date time, fixity checksums, content values, file format(s), ontologies, and logical/physical structures etc. Information that will ensure the usability of content when brought back from DPS to active use in IS. When CaPM has finished the logging of PoF-data, the archiver triggers the transfer of SIP using a REST service provided by the DPS.

[7] The digital preservation system (DPS) receives the SIP and starts the ingest process that creates an archival information package (AIP). The DPS in use in the ForgtIT project is DSpace[12] open source repository application.

[8] When DPS has created an AIP it is transferred to the preservation-aware storage service (PASS) that manages the content storage. PASS generates an AIP-ID, computes fixity and

[5] http://dbpedia.org/
[6] http://www.wikidata.org/
[7] http://www.loc.gov/standards/mets/
[8] http://www.loc.gov/standards/mods/
[9] http://www.dcc.ac.uk/resources/external/droid
[10] http://en.wikipedia.org/wiki/MD5
[11] http://riksarkivet.se/Media/pdf-filer/Projekt/eARD_informationstext_eng.pdf
[12] http://www.dspace.org/

generates an “ingest” provenance event record stored as preservation descriptive information together with the AIP. PASS supports the use of different cloud storage providers (SP). The cloud storage in use in the PoF demonstration is OpenStack Swift Open Source framework[13].

[9] When the Ingest process has been finished in the DPS the PoF-ID, AIP-ID, DPS-ID, and a status are sent to the PoF-ESB that triggers the id-manager to record this information. The status either confirms that the process went according to expectations or request for a restart of different PoF workflows. These alternative workflows could be a request for a restart of the workflow by a re-fetch of objects from active system, or repackage and retransmission of SIP. There are different causes that could lead to an alternative workflow: detection of a difference between a computed file checksum compared to what is specified in the metadata, identification of a corrupted file, expected file is missing, or if necessary metadata is missing. If the status from DPS is ok the workflow will end by the sending a notification back to the active system that requested the start of the PoF workflow.

3.3 Re-activation Workflow

As mentioned earlier, deliverable D8.2 describes the ForgetIT approach and an implementable model of this approach. In Figure 6 we see the Re-activation Workflow in the *Remember & Forget Layer*. The workflow will be described in more detail later in this section (see 3.3.1).

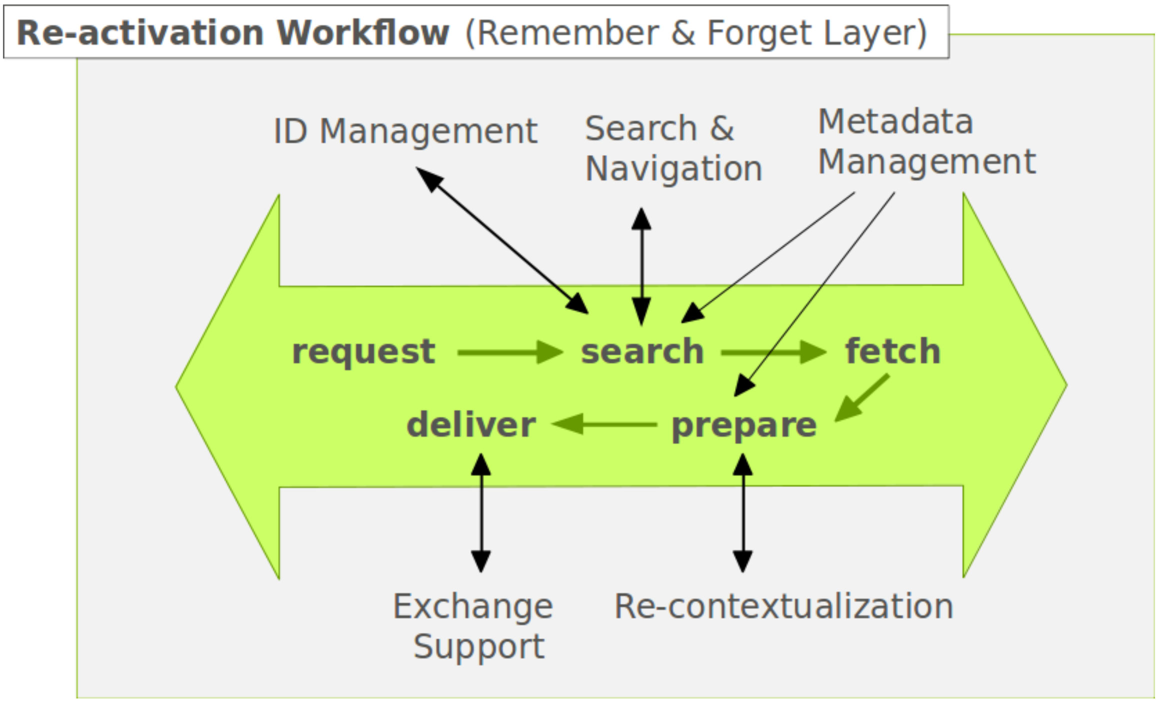

Figure 6: Re-activation Workflow, reference model [ForgetIT, 2015a, p. 24]

The workflow has five basic steps, *request, search, fetch, prepare,* and *deliver*. Although these steps are described in more detail in D8.2 [ForgetIT, 2015a], a brief description is in its place here.

[13] http://docs.openstack.org/developer/swift/

The *request* step simply initiates the process stating that we want something from the preservation system. The next step, *search*, is then the process of locating the object(s) in the preservation system, utilising functionality from *ID Management*, *Search & Navigation*, and *Metadata Management*. The *fetch* step is handled by exchange support, tailored to the DPS in question.

The next step is to *prepare* the fetched object(s) so that the re-activation in the Active System is as smooth as possible. This includes support from the *Re-contextualization* functionality (WP6) to potentially add semantic information, and the *Collector/Archiver* in order to prepare the package for delivery. The last step is *deliver*, which is cared for by *Exchange support*, putting the package up for fetching on a CMIS server that the Active System(s) then can access.

3.3.1 Detailed description of Re-activation workflow

This section contains a detailed description of the interaction between the major components of the PoF re-activation workflow illustrated in Figure 7.

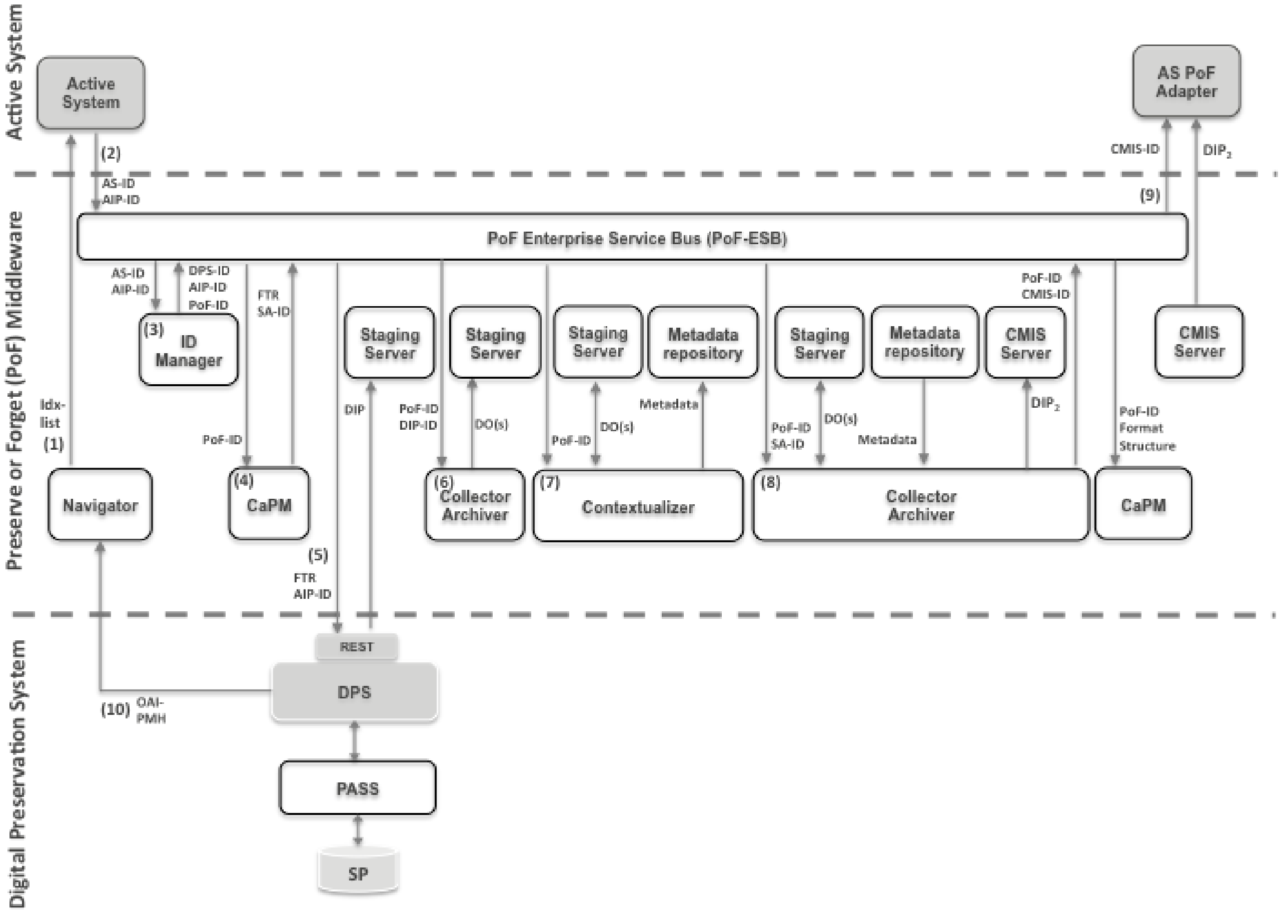

Figure 7: Re-activation Workflow

(1) The navigator component sends, on request from an active system (AS), a list of search index results (REST-URIs) that contains information that identifies an item or a collection of items in the digital preservation system (DPS).

(2) A request that contains information about the active system (AS-ID) and identification of requested item(s) (AIP-ID) is sent to the PoF middleware.

(3) The PoF-ESB redirects the item request information to the ID-manager that keep track of information identifying the digital preservation system (DPS-ID) in hold of the requested

resource(s). The id-manager creates an internal workflow identifier (PoF-ID) and forward the request to the CaPM component.

(4) The context-aware preservation manager (CaPM) receives a request (PoF-ID) from PoF-ESB that triggers a check for the existence and return of any format transformation rules (FTR) containing information about migration constraints or actions. CaPM uses the PoF-ID to request information from the id-manager that is used to identify a submission agreement that contains information about established rules between the DPS and AS. A submission agreement id (SA-ID) is sent forward to the PoF-ESB.

(5) The PoF-ESB sends a request (REST) for item(s) in the DPS identified by the AIP-ID and submitting a FTR if exists. The DPS returns a dissemination information package (DIP) as response to the request.

(6) The collector/archiver component receives a request containing information that identifies the DIP and unpacks the content on the staging server in a folder identified by the PoF-ID.

(7) The contextualizer component receives a trigger (PoF-ID) and starts the re-contextualization process. The result is stored at staging server and the metadata repository.

(8) A trigger from the PoF-ESB to the collector/archiver component starts the process of creating an adjusted dissemination information package (DIP_2). A package that is adapted to the receiving active system according to a submission agreement identified by the SA-ID. When the package procedure has finished, the DIP_2 is uploaded to an internal CMIS server. The CMIS-ID and PoF-ID is sent back to the PoF-ESB. The CaPM logs information as use of file format and physical/logical structure from the re-activation process.

(9) The PoF-ESB sends the CMIS-ID to the AS which triggers the process of retrieving DIP_2.

(10) The open archives initiative protocol for metadata harvesting (OAI-PMH) is an alternative way to expose content in a DPS that is not previous captured by the navigator.

3.4 Influence on DoW Success/Progress Indicators

As described in the DoW of the project, the performance indicator related to workflow states: "Seamless transition between active and archival system – based on forgetting methods". This seamless transition is not entirely related to components in WP5, especially not with regard to the second part, forgetting methods. The workflows described do however involve many components outside of WP5, an intended, and by employing an Enterprise Service Bus together with the active system adapters, the exchange support, as well as preparation of information packages for both preservation preparation and re-activation (ingest and access), the transition is, albeit not fully seamless, nearly seamless. This to some extent also depends on how close the integration between Active System Adapters and the Active Systems is.

There is also a performance indicator for "types of major change that can be communicated to the preservation system" where the initial work on the Context-aware Preservation Manager helps in keeping track of e.g. types and formats of material passing through the middleware as well as the physical and logical structure of this information. Not all of this information is relevant for the Preservation System, at least not until a dissemination request is made at which time the CaPM can assist with information about the current situation.

4 Implementation Details

This section describe in some detail the components implementation, dependencies and "behind the scene" interaction. It starts with a brief problem statement and then describes the preservation preparation components, the re-activation components, and the preservation management components. The section is mainly intended for readers interested in implementation details and consideration.

4.1 Problem Statement

The overarching objective for WP5 is to facilitate "smooth bi-directional transition between information management and preservation". This requires a fair amount of automation and normalization of communication and exchange of digital objects. The components implemented in WP5 mainly focus on exchange of digital objects, and communication of changes in the environment, mainly to the digital preservation system.

4.2 Preservation Preparation

The simplified component diagrams in this section depict the WP5 components and the most relevant relations to other components or modules, which also might include external components (i.e. components not built in the project). Each figure is briefly described to explain typical process flow.

4.2.1 Pre-ingest

The pre-ingest component in Figure 8 is the component that is called by the PoF Enterprise Service Bus when digital objects should be collected from the active systems. The Collector initiates the process with calling the CMIS Client (described in 4.2.2) through StartCMISClient. When the objects have been fetched, other components in the middleware process the objects. After the processing is finished, the Collector initiates packaging (see 4.2.3) of the objects by calling StartPackage. When packaging is finished, a submission to the DPS is done, in this case through a REST call.

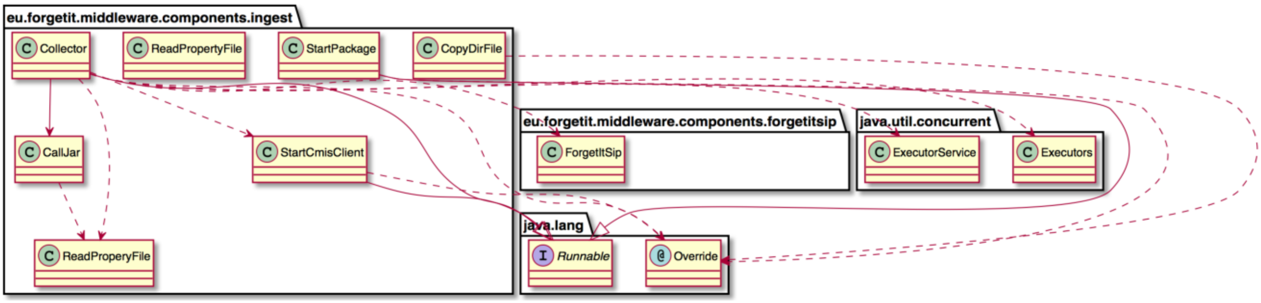

Figure 8: (Pre-)ingest module

4.2.2 CMIS Client

The CMIS Client (Figure 9) is responsible for connecting to the Active System Adapter(s) in order to retrieve digital objects. In order to do this, it utilises the OpenCMIS client API for the CMIS functionality. Based on which system the request is coming from (specified by parameter), it uses connection parameters available in a property file, download the file(s) and put them in a dedicated space, labelled with the unique PoF-ID, on the staging server.

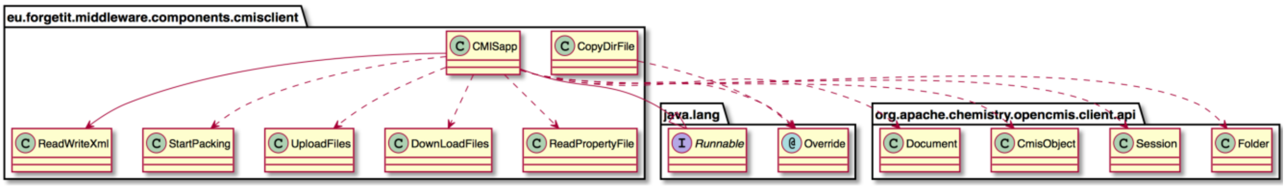

Figure 9: CMIS Client

4.2.3 Packager

This component, labelled *forgetitsip* (Figure 10), is responsible for the creation of a Submission Information Package (SIP) that adheres to the requirements of the DPS. It does so by running the DROID file format identification tool[14] to extract technical metadata about files about to be included in the package. Then it calculates fixity values for the files, which together with any metadata extracted in other PoF processes (e.g. Extractor and Contextualizer) serves as input to the MetsCreator. The MetsCreator utilises a METS API[15] from Australian National University, slightly modified, to create a METS document, which in the ForgetIT project also embed a MODS. The content files, any metadata files, and the METS document is then packaged into a tar-package or a zip-file, depending on what is expected from the DPS. The package has three subfolders, one for content, one for metadata, and one for system related information (essentially execution related information).

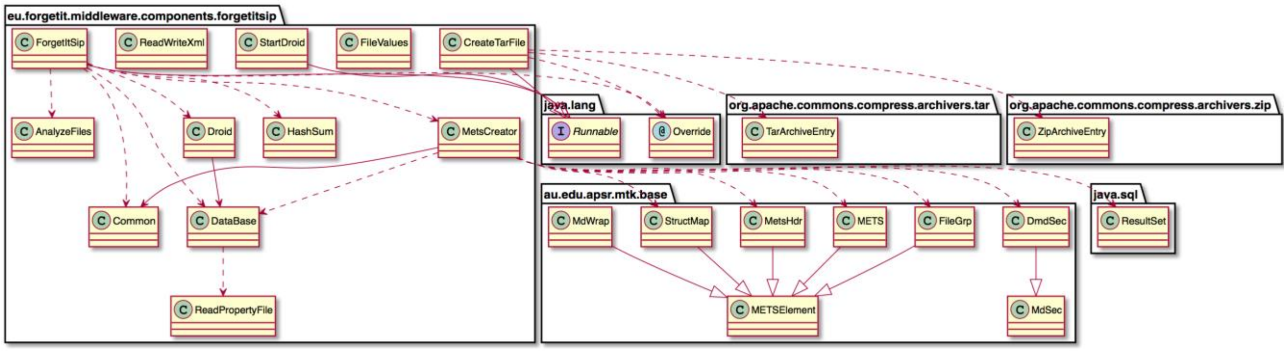

Figure 10: Package module

4.2.4 Influence on DoW Success/Progress Indicators

The progress indicators for outcome 2 (of WP5) in the DoW are aiming for: increase in *degree of automation of SIP generation*; *automated quality control of SIP*; *ease of integration of packaging generator into information management systems*. To start with the last one, that indicator is at this stage quite irrelevant since the project instead adopted a middleware approach, where the packaging generator instead resides within the middleware. Another way of looking at this is to conclude that it is very easy to integrate the packaging generator, since no integration at all is needed. You do however need to build an PoF adapter, which at the moment would provide a CMIS endpoint. The *automation of SIP generation* have been strengthened by packaging of metadata and generation of MODS section within the METS, as well as by preparation for following submission agreements that would state requirements from different preservation systems. This

[14] Digital Record Object Identification - http://www.nationalarchives.gov.uk/information-management/manage-information/policy-process/digital-continuity/file-profiling-tool-droid/
[15] http://sourceforge.net/projects/mets-api/

also ties into the *automated quality control of SIP* regarding format and metadata, which also is supported by the inclusion of technical metadata extraction through DROID.

4.3 Re-activation

The *access* component (Figure 11) start with unpacking a DIP as retrieved from the Digital Preservation System. This takes place on the *Staging Server* where the StartUpload creates a folder structure on the server where the main folder gets a PoF-ID. After this, *MovingFiles* moves the files into this structure. There is also an option to unpack the objects into a folder structure defined by an XML file (created at pre-ingest to reflect the active system structure). After this, other middleware components can process the objects if needed. The resulting folder is then shared through a CMIS server (which is outside of this component), enabling access for the Active System.

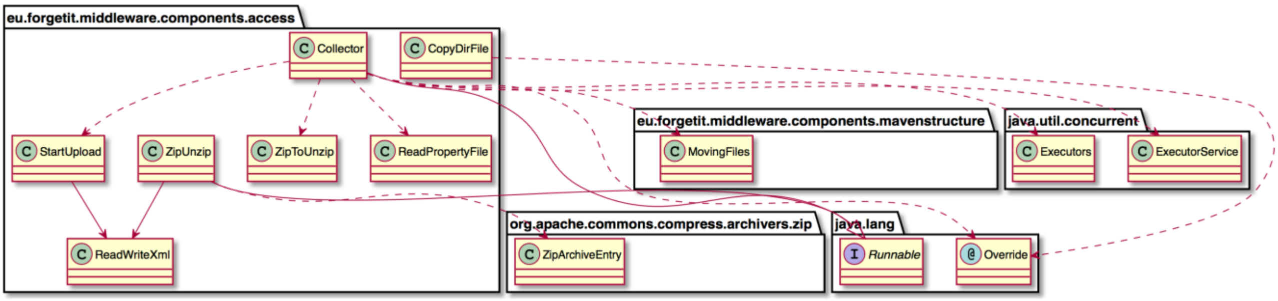

Figure 11: Access module

4.3.1 Influence on DoW Success/Progress Indicators

The progress indicators for outcome 3 (of WP5) in the DoW are aiming for: increase in *degree of automation of the transfer from archival system into active system; number of different types of content supported; flexibility with respect to different context settings*. The degree of automation could be considered fair at this stage, with good support for a simple transfer. A more complex transfer, involving e.g. several DIPs combining them into one package is not fully supported, while we on the other hand have initial support for structuring the delivered package according to requirements from the active system. The latter certainly improving the *flexibility with respect to different context settings*. The number of different types of content supported is not limited by the work conducted in WP5, but instead by the possibility to employ *extraction*, *contextualization* and other PoF middleware functionality such as *forgetting* on the different types of content.

5 Summary and Future Work

As indicated by the success/progress indicators discussion in respective sections, the transition of objects between active systems and preservation systems are in good shape, albeit there is room for refinement in especially the re-activation workflow. This includes restructuring packages according to requirements from new active systems or other changes on the consumer side.

The preservation planning workflow and related administrative tasks have been modelled in a first iteration and some basic functionality has been designed for the Context-aware Preservation Manager component. This component will be the focus for WP5 during the final year of the project and although not everything that has been identified as relevant tasks for the component will be implemented in the scope of the project, we aim at implementing advisory functionality for e.g. which actions need to be taken in order to re-activate content into a particular system.

References

[Afrasiabi Rad, Nilsson, Päivärinta, 2014] Afrasiabi Rad, P. (2014). Administration of Digital Preservation Services in the Cloud Over Time : Design Issues and Challenges for Organizations. *The Proceedings of the 2nd International Conference on Cloud Security Management*, The Proceedings of the 2nd International Conference on Cloud Security Management / edited by Barbara Endicott-Popovsky.

[ForgetIT, 2013] Päivärinta, T. et al. (2013). *D5.1: Concise preservation by combining managed forgetting and contextualization remembering: Foundations of synergetic preservation*. ForgetIT.

[ForgetIT, 2014a] Gallo, F. et al. (2014). *D8.3: The Preserve-or-Forget Framework – First release*. ForgetIT

[ForgetIT, 2014b] Nilsson, J. et al. (2014). *D5.2*: *Workflow model and prototype for transition between active system and AIS - first release*. ForgetIT

[ForgetIT, 2015a] Gallo, F. et al. (2015). *D8.2: The Preserve-or-Forget Reference Model Initial Model*. ForgetIT

[ForgetIT, 2015b] Greenwood, M.A. et al. (2015). *D6.3 Contextualisation Tools - Second Release: Updates to the Context Modelling Framework and Module.* ForgetIT

[ForgetIT, 2015c] Kanhabua, N. et al. (2015). *D3.3: Strategies and Components for Managed Forgetting – Second Release*. ForgetIT