# FP7-ICT-2007-3-231161


## Deliverable 5.2.2
## First Prototype of Open PrestoPRIME Reference Implementation


F. Gallo (EURIX)

2011-12-23

# Document Administrative Table

| | |
|---|---|
| Document Identifier | PP_WP5_D5.2.2_FirstPrototype_R0 Release 0 |
| Filename | PP_WP5_D5.2.2_FirstPrototype_R0_v1.00.pdf |
| Workpackage and | WP5 - Integration and validation |
| Task(s) | WP5T2 - Integration of archives, libraries and UGC |
| Author (company) | Francesco Gallo (EURIX) |
| Contributors (company) | Francesco Rosso (EURIX), Philip Kahle (University of Innsbruck), Matthew Addis, Stephen Phillips (IT Innovation) |
| Internal Reviewers (company) | Laurent Boch (RAI), Adil Hasan (University of Liverpool) |
| Date | 2011-12-23 |
| Status | Release |
| Version | 1.00 |
| Type | Deliverable |
| Deliverable Nature | Report |
| Dissemination Level | Public |
| Planned Deliv. Date | M36 - 31/12/2011 |
| This IsPartOf | |
| This HasPart | |
| Abstract | First Prototype of Open PrestoPRIME Reference Implementation |

# DOCUMENT HISTORY

| Version | Date | Reason of change | Status | Distribution |
|---|---|---|---|---|
| v0.01 | 2011-11-15 | Created index and sections | Outline | Confidential |
| v0.02 | 2011-11-30 | Added new sections and text | Outline | Confidential |
| v0.03 | 2011-12-06 | Completed sections 1,2,3,4,6,7 | Working Draft | Confidential |
| v0.04 | 2011-12-12 | Updated missing sections | Working Draft | Confidential |
| v0.05 | 2011-12-16 | Completed section 5, added external tools | Working Draft | Confidential |
| v0.06 | 2011-12-22 | Completed missing sections, included reviewers' comments | Final Draft | Confidential |
| v1.00 | 2011-12-23 | Delivered | Release | Public |

# Executive Summary

The present document describes the first integrated prototype of the open PrestoPRIME Preservation Platform (referred to as P4 in the following).

**Architecture** The implemented prototype is based on the reference architecture defined in deliverable D5.2.1. The prototype implements an OAIS-compliant preservation system integrating several tools and services provided by project partners. We designed the architecture using the following approach: we first defined the needed abstract components using Object Oriented Design principles and design patterns; then we derived the architecture as a result of such building blocks. Using such Model Driven Architecture design, we mapped the six OAIS functional entities mentioned above to software components making use of available design patterns. Several design patterns were identified for each component and were brought together to build up the software component diagrams of the functional entities.

**Data model** The implemented prototype and and is fully compliant to the final AV data model specification described in D2.1.3. In our representation the three OAIS Information Packages are represented by XML files containing a METS schema. The METS standard was identified as a convenient wrapper and we selected several cases for the use of METS in PrestoPRIME scenarios, taking into account the number of audiovisual files referenced in the METS and the way they are used to represent an editorial entity. PrestoPRIME AV data model contains the description of all metadata formats used in the project to describe content and how such representations were integrated in the overall data model. The prototype implementation described in this document is fully compliant to the data model specification in D2.1.3.

**Tools and technologies** Several tools and technologies were used for the prototype development, whenever possible we have chosen free open source technologies supported by an active community.

**P4 implementation** The P4 protype is made up of three components: core modules (OAIS compliant), REST web server and user interface; in addition to these components, we integrated several tools and services provided by project partners, in order to setup a demo for the last test-bed event. In the core components we implemented the main functional entities of the OAIS model: ingest, access, administration, data management, storage and preservation planning. Ingest, access and administration can be accessed through the web server interfaces. These three components make use of data management, storage and preservation planning. The preservation platform can be accessed through a web server. The web server provides interfaces for ingest, access and administration. It can be accessed remotely through the user interface or from a user application. It has been implemented as a CRUD web server exposing REST APIs. We implemented a web interface for interacting with the platform. The interface provides functionalities for ingest, access and administration. Basically the user can make use of the web interface to ingest SIP files in to the platform, to get information about the status of the submitted jobs and of the whole system, to search for AIP available in the archive and to get access

to the DIP.

**License** The software components of the P4 implementation (core modules, web server and user interface) are released under open source license (GPL), while for external tools the choice of the license if left to the owner.

**Documentation** The software documentation and installation guides are provided, with a set of minimal requirements; the installation process will be improved for the final version of P4.

**Demonstration** The current release of the prototype has been tested during the last evaluation phase which was held in RAI in November 2011 (see deliverable D5.4.3).

**Future work** All the feedbacks collected at the test-bed as well as several improvements identified within the project concerning P4 components and external tools will be evaluated in the final year of activity and the final release will be described in D5.2.3.

# Table of contents

# 1 Introduction

The present document describes the first implementation of the PrestoPRIME Preservation Platform (P4). The platform has been implemented according to the architecture designed in deliverable D5.2.1 [1] and is fully compliant with the final data model specification described in deliverable D2.1.3 [2].

The prototype implements an OAIS-compliant preservation system integrating several tools provided by project partners. The current implementation was used during the test-bed event which was held in RAI in November 2011, as reported in D5.4.3 [3].

The current prototype will be improved based on the feedback from the test-bed and a final prototype of the preservation platform will be delivered at M45 and will be described in D5.2.3 [4].

## 1.1 Architecture Design

Deliverable D5.2.1 - *Definition and Design of a PrestoPRIME Reference Architecture for the Integration Framework* [1] was delivered at M12 and contained the design of the preservation platform based on the scenarios reported in deliverable D5.1.1 [5]. In this document we analysed the results of other projects and initiatives in the field of digital preservation, focusing on the system architectures and software solutions produced at that time (see D5.2.1 Section 2). The conclusion of such analysis was that, compared to other projects, PrestoPRIME was strongly focused on digital preservation of AV content, with the opportunity to release the first implementation of a preservation platform based on the OAIS model for AV content. Additionally in D5.2.1 we evaluated the approach used by other projects and institutions for the definition of SIP (see D5.2.1 Section 3), AIP and DIP and we defined the basic structure of SIP.

In our representation the three OAIS Information Packages are represented by XML files containing a METS[1] schema. The METS standard was identified as a convenient wrapper (see D5.2.1 Section 4) and we selected several cases for the use of METS in PrestoPRIME scenarios, taking into account the number of audiovisual files referenced in the METS and the way they are used to represent an editorial entity. For example we identified the one-to-one case where one editorial entity is represented by exactly one file or the one-sequence case where one editorial entity is represented by several files in sequence. For each case we studied examples provided by project partners and defined the structure of the SIP. We also studied in detail the features of the METS standard and identified specific requirements for PrestoPRIME scenarios. In Figure 1 the structure of the METS wrapper and the different types of metadata used in the SIP are depicted. Examples of SIP and AIP can be found in Appendix B and C.

For further details please refer to D5.2.1 Section 4. The different metadata types and

[1] http://www.loc.gov/standards/mets/

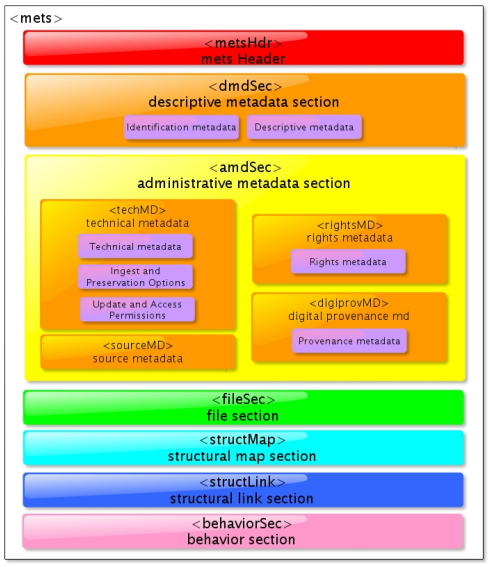


Figure 1: *PrestoPRIME SIP: METS wrapper and metadata sections*

formats to be used in the SIP were studied and included in the data model specification, as described in the next Section.

The other major result of D5.2.1 was the design of the reference architecture, with the representation of the main OAIS components (Ingest, Data Management, Preservation Planning, Storage, Access and Administration) using UML 2.0 diagrams and design patterns. A crucial requirement for the architecture design was the possibility to integrate external tools and services provided by project partners or available from outside to implement specific features. We designed the architecture using the following approach: we first defined the needed abstract components using Object Oriented Design principles and design patterns; then we derived the architecture as a result of such building blocks. In Figure 2 the component diagram of the preservation platform is depicted. Using such Model Driven Architecture design, we mapped the six OAIS functional entities mentioned above to software components making use of available design patterns. Several design patterns were identified for each component and were brought together to build up the software component diagrams of the functional entities.

Finally we decided to release the core components of the preservation platform with an open license. This issue is discussed in detail in Section 10.

## 1.2 AV Data Model

Deliverable D2.1.3 - *AV Data Model: Final Specification* [2] was delivered at M12 and contained the description of all metadata formats used in PrestoPRIME to describe content and how such representations were integrated in the overall data model. The prototype

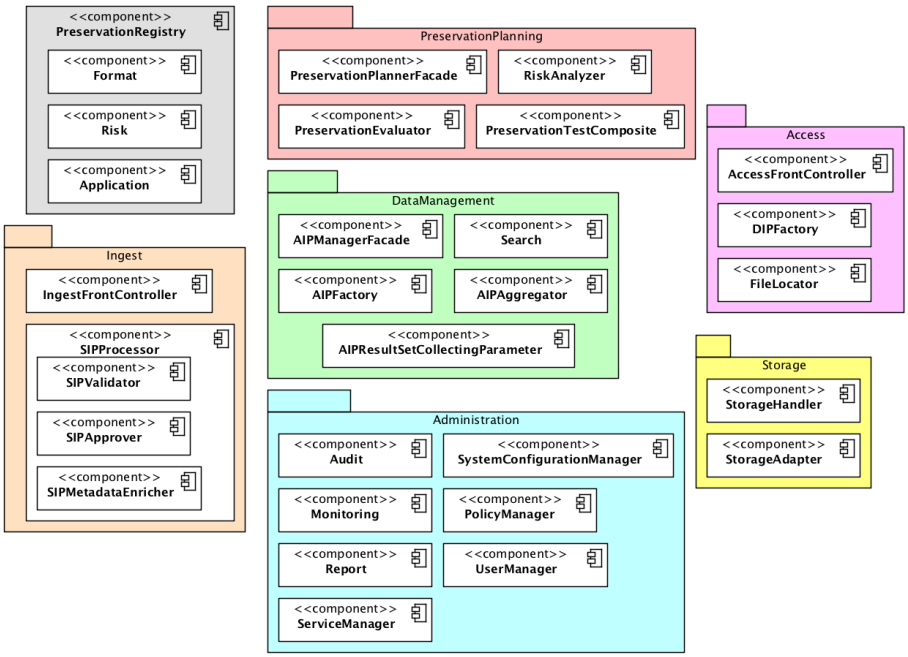


Figure 2: *PrestoPRIME Preservation Platform Component Diagram*

implementation described in this document is fully compliant to the data model specification in D2.1.3. We made some assumptions (see D2.1.3 Section 2): OAIS is the reference model for the preservation system and METS is the standard of choice for representing the wrapper in the PrestoPRIME SIP. We identified several benefits in the use of METS, such as the wide popularity of the standard, the modularity of the schema, the possibility to accomodate virtually any metadata format chosen by the user, an efficient way to relate different parts in the schema and the association between AV files and their metadata representation, just to mention a few. An other advantage of METS is that it was chosen also for Rosetta. The AV data model (see D2.1.3 Section 3) covers the representation of the editorial entity as well as different kinds of metadata involved: descriptive, technical and rights. In the definition of the AV data model, we used extensively the DNX format, a simple and unified XML schema which is used in Rosetta to represent different metadata types (see D2.1.3 Section 4). Finally we analyzed different metadata types in detail, proposing several standard formats for the reference implementation (see D2.1.3 Sections 5 and 6). It is worthwhile noticing that the choice of metadata formats takes into account the different requirements from all partners for what concerns the preparation of the SIP and the use of such formats in different tools and services. The list of XML formats and related namespaces used in the data model can be found in Appendix A.

Both the open preservation platform described here and Rosetta, the commercial solution from ExLibris, are compliant to OAIS model and implement the same data model, while some of the results have been integrated in both systems (e.g. tools for MXF files developed in PrestoPRIME).

# 2 Technologies

In this chapter we will discuss the technologies used in our implementation of P4, the reasons why we chose them and the features exploited to improve our product.

## 2.1 Development

The most important technologies used for the implementation, starting with the programming language and ending with help libraries.

### 2.1.1 Java

Java[2] is a cross-platform, high-level object oriented programming language that allows the execution of the same code on different machines with different OSs; the only requirement is the Java Virtual Machine.
In details, we used two different Java specifications: J2SE and J2EE. The aim of the first one is to provide tools that are useful to create standard applications that compute some operations with integers, strings and objects, while the second one is an extension and provides some useful tools to create dynamic web applications. Naturally, our application core is based on J2SE, while both the web server and the web user interface (details in the following chapters) are using J2EE and Java Server Pages plus other libraries.
The DataModel Java Object representation is automatically generated from the XML schema with the help of JAXB, a tool with the scope of binding XML into Java Objects.

### 2.1.2 HTML5

We tested new features provided by the young W3C[3] standard HTML5[4]; in particular we avoid the need of a streaming server introducing the simple tag <video> in our HTML code. Also the pages style was extremely bounded into the CSS stylesheet, to be compliant to the suggestions of W3C.

### 2.1.3 ECMAScript

The old JavaScript, now renamed to ECMAScript[5], lets us compute some code on the client side, especially introducing AJAX heavily, making pages more light-weight as heavy

[2] http://java.com
[3] http://www.w3.org
[4] http://www.w3.org/TR/html5/
[5] http://www.ecmascript.org

contents are loaded dynamically after the page has been displayed. To simplify the development we used the popular jQuery library, that adds clever wrappers for commonly used Javascript functions while maintaining cross-browser compatibility of the dynamic pages. Besides that, jQuery's event handler model vastly eases the separation of page content and behaviour, a paradigm known as unobtrusive Javascript, which allows us to write much cleaner code.

### 2.1.4 Eclipse IDE

Eclipse[6] is an integrated development environment (IDE) that is written in and focused on Java. It is highly extensible via plugins to support various programming languages and development tools, fitting virtually any needs. Another major advantage of this software is that it analyzes written code in realtime, looks up and suggests possible operations and warns about imminent problems and errors. As many of the herein presented technologies can be integrated into Eclipse it can be thought of as a central point in development.

### 2.1.5 Subversion

Subversion[7] is an open source project, useful for managing the team development and permanently keeping track of changes. This tool works as a central service that gives the possibility to some authenticated users to download the latest version of the code and upload only the changes without overwriting previous versions of the files. With some very simple commands, it is possible to recover an old version, monitor the differences between two versions and restore, if needed, an old running version. Subversion can be published on the web simply via Apache HTTP server and can manage, as mentioned, users and roles: every developer has an account and all the changes are related to a user; in this way it is also possible to track the work of a single user.

Subclipse[8] integrates subversion with Eclipse and allows a developer to access the subversion repository from within Eclipse. With a couple of simple commands, you can commit new code to the server or restore the old one from the network. SubClipse uses a low-level library, JavaHL, to deal with subversion on a specific platform and OS.

### 2.1.6 Jersey

Jersey[9] is the JAX-RS reference implementation released under open source license. Its aim is to provide tools to simplify the creation of REST interfaces using the Java Servlet

[6] http://www.eclipse.org
[7] http://subversion.apache.org
[8] http://subclipse.tigris.org
[9] http://jersey.java.net

API. Jersey is a library integrated in the project that provides Java annotations and a single master Servlet for completing this task. As described in Section 5, each method of the P4 web server is published via HTTP as an interface; by simply calling this interface a user can request the execution of this method remotely. Jersey supports all HTTP methods, like GET, POST, PUT and DELETE, and all incoming and outcoming MIME types. Moreover, Jersey allows to publish the complete interface as WADL file (Web Application Description Language) in a machine readable manner, simplifying the development of applications that shall access the REST service; the current P4WS interfaces descriptor is reported in the Appendix D.

### 2.1.7 Apache Lucene and Solr

Apache Lucene[10] is a Java library for text indexing and search. In contrast to a database it does not necessarily store data in its original form but builds an inverted file index from all terms occurring in the data. Furthermore, Lucene supplies several configurable filter components that reduce the amount of text for instance by removing stop words or lowercasing all characters. This heavily reduces the occupied disk space and access time allowing high-performance searching even on very large amounts of data. Apache Solr[11] is a server application built on top of Lucene that maintains the search index independently from the applications that access it which perfectly fits the modular architecture of P4. One of the main advantages of using Solr over Lucene is that it allows to write a schema for the index in XML (somewhat comparable to the schema of a relational database table) which is a very convenient method for structuring the text data. For each field on the schema one can accurately define filter chains, that process the text before indexing or a query before matching it to the indexed data.
P4's search component completely operates Solr via its Java API SolrJ. This way all data may be gathered, indexed and queried conveniently using the same language. Also this interface speeds up communication between Solr and P4 as SolrJ streams data in javabin format without the detour HTTP.

### 2.1.8 Apache Ant

Ant[12] gives us the possibility to define targets, like procedures, useful for instance to compile your code from command-line or to clean your project. As there is a plugin for Eclipse Ant targets may be run directly from within the IDE.

Ivy[13] complements Ant, providing a simple library to download the latest version of each library in the project. Using this tool, you don't have to replace each library every time it is updated; you only have to publish it on the web and, during the execution of a special

[10] http://lucene.apache.org/
[11] http://lucene.apache.org/solr/
[12] http://ant.apache.org
[13] http://ant.apache.org/ivy

Ant target, Ivy will automatically download the last one and put it into the project lib folder. With Ivy it is also possible to manage dependencies of a library, telling Ivy that it has also to download other external tools useful to run the first one.

### 2.1.9 XSLT

XSL transformations (XSLT) are a crucial tool in all applications that make heavy use of XML data. It can be used to transform a given XML such that it obeys a foreign schema or even into another format, e.g. plain text or HTML. A XSL transformation is defined using the eXtensible Stylesheet Language (XSL), which is a Turing-complete programming language designed specifically to process XML data, making XSLT outclass any object-oriented programming language, such as Java, when it comes to these tasks.

## 2.2 Databases

We adopted two types of databases for storing data and metadata. For information packages we used a solution based on a native-XML database because SIP, AIP and DIP are represented by XML files, as discussed before. Then we need to store data related to the preservation system itself, such as information used for jobs monitoring, users management, logging and other temporary information (e.g. tokens with expiry dates used in the OAI-PMH interface). For this kind of data we use a typical DBMS supporting SQL queries.

### 2.2.1 Exist

Exist[14] is a native-XML database that supports the most prominent XML related technologies, such as XPath and XSLT. Because our system has to store .XML files (the SIPs) in the first line, a relational database is not the best choice as it is very difficult to model the data in an efficient way: a little change to the XML structure involves a big change to the database tables; a simple query involves a lot of joins, a simple XML structure involves a lot of different tables. An XML based database, like Exist, removes all these issues and abandons SQL in favor of XQuery which is much more suitable.

### 2.2.2 Apache Derby

Derby[15] is an embedded relational database entirely written in Java with a very small footprint, provided by the Apache Software Foundation with an open source license. In

[14] http://exist.sourceforge.net
[15] http://db.apache.org/derby

contrast to Exist it relies on classic SQL. The biggest added value of Derby is that you can start, query and stop the database directly within the code; further Derby stores all its files into a folder on your file system and you can migrate it onto another server while keeping all the database contents simply by copying this folder to the right position.

## 2.3 Deployment

To get a running product, there are two requirements: a servlet container, able to run Java web applications, and a Web Server, to publish files via HTTP.

### 2.3.1 Apache Tomcat

Tomcat[16] is a popular servlet container useful to deploy J2EE applications. It is very simple, without any sophisticated tools, but with all the utilities needed to keep the execution of the Java code under control and find errors, bugs and malfunctions. Tomcat offers also the possibility to work on a Secure Socket Layer (SSL) and keep the authentication under control with a basic user management. We use Tomcat to deploy both the P4WS and the P4GUI, but also our customized version of Exist and Solr.

### 2.3.2 Apache Web Server

Apache Web Server[17] is the most widespread web server. We use it to publish audiovisual materials in both low quality and master quality, thumbnail images, keyframes and so on. Apache Web Server allows us to publish all the things needed to compose the user interface on the network.

## 2.4 AV formats

Several AV formats are supported, please refer to previous project deliverables for additional information. For master quality videos we basically make use of MXF or high quality AVI files. For browsing quality, we make use of free open source formats such as OGV[18] and WebM[19]. These formats are supported also by HTML5, which is used in the P4 interface for video streaming and preview. The tools used for transcoding AV material are described in the next Sections.

[16] http://tomcat.apache.org
[17] http://httpd.apache.org
[18] http://theora.org/
[19] http://www.webmproject.org/

# 3 P4 Implementation

The implementation of the first prototype of the preservation platform can be split in three parts: core components (implementing the OAIS functional entities), REST web server and user interface. We briefly describe each part below, then we provide further details for each of them in the next Sections.

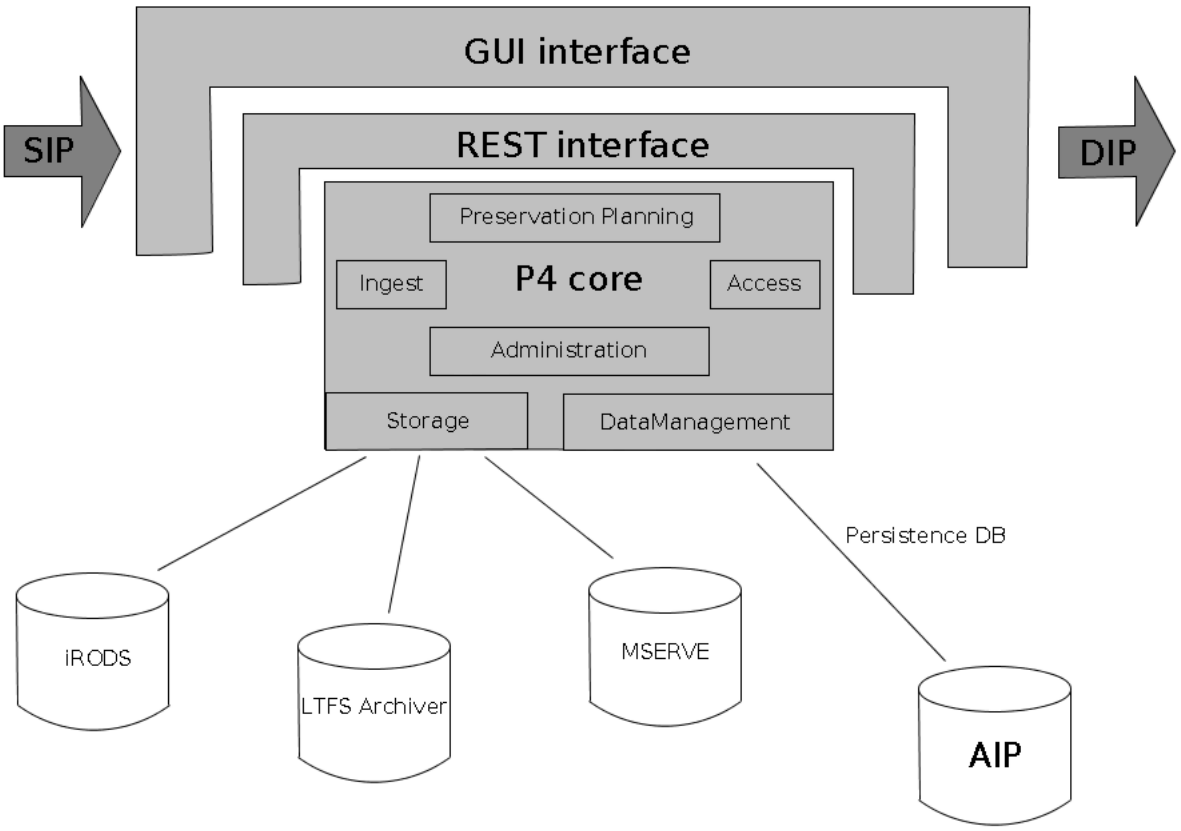


Figure 3: *P4 overview: core components, REST web server and web interface are shown*

The overview of the implemented prototype is depicted in Figure 3, which shows the three parts mentioned above: ingesting SIP, storing AIP and accessing DIP. Concerning storage, three different solutions taken from the test-bed setup are shown in Figure 3. For data management we show the persistence database (used to store AIP). For other components the details are omitted.

## 3.1 P4 core components

We implemented the main functional entities of the OAIS model: ingest, access, administration, data management, storage and preservation planning. Ingest, access and

administration can be accessed through the web server interfaces. These three components make use of data management, storage and preservation planning. Details of the P4 core components can be found in Section 4.

## 3.2 P4 web server

The preservation platform can be accessed through a web server. The web server provides interfaces for ingest, access and administration. It can be accessed remotely through the user interface or from a user application. It has been implemented as a CRUD web server exposing REST APIs. Details of the P4 web server and the available interfaces can be found in Section 5.

## 3.3 P4 web interface

We implemented a web interface for interacting with the platform. The interface provides functionalities for ingest, access and administration. Basically the user can make use of the web interface to ingest SIP files in to the platform, to get information about the status of the submitted jobs and of the whole system, to search for AIP available in the archive and to get access to the DIP. Details of the P4 web interface can be found in Section 6.

# 4 P4 Core Components

In the following we describe the P4 components which have been released in the first version of the prototype. We first describe OAIS-compliant modules and then we describe additional components which have been introduced for specific requirements. The development of some of the components is still in progress and will be performed in the final year.

## 4.1 OAIS components

In this Section we briefly describe the core components of the platform which implement the OAIS functional entities. For each component we report the Java package name, the main classes and a short description. Further details can be found in the software documentation (see Section 9).

In the current implementation we provide a REST web server with ingest and access interfaces. Such interfaces are described in Section 5. Regarding all activities related to SIP ingestion and DIP access, the P4 web server interfaces make use of two classes: `IngestManager` (ingest component) and `AccessManager` (access compoment) which in turn make use of the class `DataManager` (data management component).

### 4.1.1 Ingest

***Package names***: `eu.prestoprime.ingest, eu.prestoprime.ingest.modules`

***Main classes***: `IngestManager, IngestConfiguration, FixityChecker, FramesExtractor, IndexingManager, MDEnrichment, RightsManager SIPParser, SIPValidator, VideoTranscoder`

***Description***: For the first prototype we identified an ingestion workflow made up of several steps: SIP validation, metadata enrichment, creation of a low quality copy, production of key frame sequences, permanent storage of the AV file, fixity checks, creation and persistent storage of the AIP, metadata indexing for search and AV files indexing (DRACMA). When a new SIP is submitted for ingestion through the web server interface, a new process is created and the SIP is passed to the IngestManager which is in charge of executing the workflow described above. For each step in the workfow we defined a separate class in charge of a specific task. Such classes are contained in the package `eu.prestoprime.modules`, including P4 tools and wrappers interfacing with external tools. For the functionalities of each class in `eu.prestoprime.modules` please refer to the software documentation. The ingest component makes use of the data model implementation in order to validate SIP; it also uses the data manager for all operations related to AIP. For each step in the workflow the `IngestManager` saves the intermediate XML with the result of

processing. The ingest component also depends on the storage component in order to handle different scenarios, as described below. The `IndexingManager` class is in charge of indexing metadata used for the search and the rights information. The `IndexingManager` is used by the P4 web interface.

***Notes***: Currently we manage multiple ingestion by running separate tasks, a possible improvement for this component would be the implementation of an advanced workflow manager based on message queues. We will evaluate the integration with available tools, such as MServe, which provides a workflow engine.

### 4.1.2 Data Management

***Package names***: `eu.prestoprime.datamanagement, eu.prestoprime.model, eu.prestoprime.model.oais`

***Main classes***: `DataManager, DatabaseManager, IdentityManager, OaiPmhManager, PersistenceManager, DataModelTools, SIP, AIP, DIP`

***Description***: Data management component contains classes for the data model and for AIP persistence. The package `eu.prestoprime.datamodel` contains classes implementing the final data model specification (see deliverable D2.1.3 [2]). For each metadata format there are automatically generated classes which constitute the object representation of the XML schema binded with JAXB. The data management components provide CRUD methods for the persistence database, which is implemented as a native-XML database. Class `DataManager` is used by IngestManager and makes use of classes for AIP persistence (`PersistenceManager`) and for other support databases (`DatabaseManager`). For OAIS information packages three classes have been defined: SIP, AIP, DIP. They are used by other components and contain properties for assigning identifiers and metadata representations.

***Notes***: The data management components could be improved in order to support additional metadata formats.

### 4.1.3 Access

***Package names***: `eu.prestoprime.access, eu.prestoprime.search`

***Main classes***: `AccessManager, Searcher, P4Indexer, AbstractIndexer AbstractDocumentIndexer, AbstractPOJOIndexer`

***Description***: The access component contains classes implementing access functionalities. The `AccessManager` makes use of the data manager and data model components in order to access AIP from the persistence database. The access component includes also the search classes which can perform queries on the persistence database or on the metadata index. The `AccessManager` is used by the P4 web

interface.

***Notes***: The development and improvement of the access component will be mainly related to the `AccessManager` class, adding missing functionalities for the integration with the data management component.

### 4.1.4 Storage

***Package names***: `eu.prestoprime.storage`

***Main classes***: `IRODSManager, LTOManager, LTOPerformer, LTORequest, LTOResponse`

***Description***: The storage component provides utility classes for different storage scenarios. In the first prototype implementation we identified three different solutions: (1) first copy on disk (NFS) and second copy on LTO tape (2) copy to iRODS (3) copy to MServe. The storage classes implement an interface mechanism with the storage device. For example, in case of copy to LTO, the `LTOManager` interacts with the LTO service developed by RAI evaluating response and critical situations, taking decisions and finally returning the reference to the tape to be stored in the AIP. The `LTOManager` can also handle requests for restore of corrupted files from LTO (see also Section 6).

***Notes***: Further development of the storage component will focus on a better integration with iRODS through the Java APIs and with MServe. An other feature which could be investigated is the development of a simple and flexible local storage management.

### 4.1.5 Administration and Preservation Planning

These components are still under development. Some functionalities related to preservation and archive administration are already available in other components, developed in the other work packages. For example WP2 developed preservation planning tools (see Section 7.1) which were shown at the test-bed event and are currently available as standalone tools. The integration of these external tools and the development of new features will be carried out in the next year.

A feature related to these components which has already been implemented is the execution of automatic tasks for persistence database backup and integrity check, as well as a feature for AIP versioning. A configurable task is executed, which performs periodic backups of the whole database containing AIP XML files. A number of incremental backups is regularly executed, then a full backup with integrity checks of all XML files and the database index is performed and a report is created for the administrator. In addition to this, each time the XML of a given AIP is modified, a new version is created and the whole AIP history is kept. It is possible to retrieve previous versions of the same AIP and restore them. Examples of this feature are provided in Section 6.

Currently the P4 web server provides three interfaces for computing fixity checks on AIPs in the archive, for restoring a corrupted file from second copy on LTO and for monitoring the status of ingest queue (see Section 5.3). A possible development for the next year could be a generalisation of the restore functionality to recover from any second copy registered in the system (unless an independent recovery service is not already provided by the storage service itself).

## 4.2 Other components

In the following we list other modules which are used by OAIS components.

### 4.2.1 Tools

***Package names***: `it.eurix.tool, it.eurix.tool.api, it.eurix.tool.xjc`

***Main classes***: `ToolManager, ToolProcessor, ToolFactory, GenericTool, Executable, Tool, Dynlib, FFmbc, FFmpeg, MXFTechMDExtractor, D10SumChecker, XSLTProc, MessageDigestExtractor`

***Description***: This component provides a flexible mechanism to integrate and execute tools which are intended for specific tasks. The integration of these tools can be twofold according to their implementation: some tools have been implemented in Java and the source code or the JAR library is available, while other tools are available as binary tools written in C/C++ or other languages and are dependant on operating system and architecture. Examples of Java-based tools integrated in P4 are the MXFTechMDExtractor or D10SumChecker tools for metadata enrichment (see Section 7.6). Examples of binary tools are FFmbc and XSLTProc. The integration of Java-based tools is straightforward, for binary tools we implemented wrappers for runtime execution. This component provides several APIs: `GenericTool` and `Tool` are interfaces implemented by all concrete classes for each tool; `ToolFactory` returns instances of implemented tools or new ones to be configured, `ToolProcess` is in charge of the execution of tools as separate processes and provides methods to retrieve all output streams; `ToolManager` loads the configuration and provides methods for executing all processes. We make use of XML configuration files for each tool, for binary tools the configuration includes also information about executables and dynamic libraries. For further details please refer to the software documentation (see Section 9).

***Notes***: The development of this component in the next year will focus on the integration of new tools required for running specific tasks which have not been implemented yet.

### 4.2.2 Workflow

This component is still under development. We investigated several technologies for the implementation, such as the use of BPMN for modelling business processes and jPDL as a machine processable language. We were able to run some test jobs and to map ingest workflow components to business processes in jPDL. We also investigated other technologies, such as Java Message Service (JMS), in order to implement a message queue for asynchronous behaviour. We defined some test queues with specific topics and implemented some test classes as listeners. The benefit of having a message broker is mainly due to the fact that the implementation of each process is decoupled from the overall workflow. We tested Apache ActiveMQ as a possible implementation. We also started investigating the features of MServe and its workflow engine, which includes a message broker based on RabbitMQ.

Finally, WP2 developed several tools, including a tool for the simulation of digitisation and migration workflows (see Section 7.1), which is currently available as a stand-alone tool and offers several functionalities for managing preservation workflows.

The possibility to integrate P4 with such workflow components for task execution will be investigated in the next year and the final implementation will be described in D5.2.3.

### 4.2.3 Registry

This component is still under development, it will be included in the second prototype and reported in deliverable D5.2.3 which will be released at M45. This component will benefit from the activities in the MPEG digital preservation initiative and from other preliminary studies carried out in the project.

# 5 P4 Web Server

This section is dedicated to the CRUD Web Server exposing REST APIs, created with the help of Jersey (see section 2.1.6). With simple GET or POST HTTP requests it is possible to get some informations, act on the platform and manage the AIPs.

A complete list with all available details is published by Jersey itself and it is reachable at the URL `<context>/application.wadl`. The current interfaces descriptor is reported in the Appendix D.

## 5.1 Access

- `GET /access/checkCredentials`
  Returns a welcome message in plain text if the credentials used for the HTTPS connection are accepted by the server.
- `GET /access/dip/list`
  Lists all the DIPs IDs in plain text, useful to ask for details on a precise DIP.
- `GET /access/dip/<id>`
  Provides the XML representation of the DIP.
- `GET /access/dip/preview/<id>`
  Lists in plain text all the low quality files available for browsing the DIP.
- `GET /access/dip/info/container<id>`
  Checks if the container of the master quality file is MXF and if the mapping is D10. Returns plain text.
- `GET /access/dip/info/dracma/<id>`
  Returns (if any) the UMID to be used to retrieve selected segments of the master quality file from DRACMA, in plain text format.
- `GET /access/dip/info/history/<id>`
  Returns the plain text history (revisions, modifications, etc.) for a given AIP.
- `GET /access/dip/<id>/rev/<rev>`
  Returns a specific revision of a given AIP, obviously in XML format.
- `GET /access/dip/thumb/<id>`
  Returns the URL of the DIP thumb.
- `GET /access/dip/multivalent/<id>`
  Compones a Java Web Start launcher for the Multivalent tool with the chosen DIP.

- `GET /access/dip/video/<id>`
  Creates an HTML page on the fly, with a video player embedded, and starts playing the chosen DIP.

- `GET /access/dip/frames/<id>`
  Lists all the frames extracted during the ingestion phase returning in plain text; useful to quickly surf the browsing quality and select a portion to be extracted by DRACMA.

- `GET /access/dip/rights/<id>`
  Returns the OWL URL related to that the DIP.

- `GET /access/dip/graph/<id>`
  Differently from the previous interface, it returns the URL pointing to the graphical representation of the rights.

- `GET /access/dip/dc/<id>`
  Extracts and gives only the Dublin Core XML sections of the DIP.

- `GET /access/dip/qa/<id>`
  Extracts from AIP and returns the quality analysis results in MPEG-7 format.

- `GET /access/dip/usermd/<id>`
  Extracts from AIP and returns the user metadata annotation from tagging game in MPEG-7 format.

- `POST /dip/rights/query-by-sample` Takes in input an OWL file containing a query instance and returns an XML response with a list of results, if any. Each row in the result set contains the information about the editorial entity identifier (typically a DC identifier) as well as a thumbnail of the video and a graph representing the rights information in the OWL. This response is used by the RightsDraw tool (see Section 7.5).

### 5.1.1 Search

- `GET /access/searchDCRecords`
  A quick and easy way to search on Dublin Core records contained in all AIPs in the database. It returns the list in plain text of the DIPs the satisfies the search parameters (“title”, “description”, “format”, “identifier”).

- `GET /oai`
  With a request with parameters set as described in the OAI-PMH standard protocol, this interface provides a standard response in XML format.

- `GET /search/quick`
  Exposes the search functionality of the Solr-based search component and accepts a search term compliant to the query language of Solr. Returns an XML that contains all search results and the given query parameters.

- `GET /search/reindex`
  Triggers a complete rebuild of the Solr index on the basis of the currently stored AIPs.

## 5.2 Ingest

- `POST /ingest`
  This interface requires a multipart POST request, with the SIP to be processed (”sipfile” param) and the storage method to be used (”store” param). The server will try to parse the SIP and run the workflow for the ingestion, returning a JobId useful for get work progress informations.

- `GET /ingest/getStatus`
  With the JobId returned by the previous interface and sent as query parameter (“jobId”), this interface returns the current status of the job and work progress in plain text.

- `GET /ingest/getResult`
  In the same way as the previous interface, once the ingestion workflow is ended, it is possible to get information about the result in plain text.

- `GET /ingest/checkIdentifier`
  The Dublin Core identifier must be unique, so with this interface it is possible to check if the desired identifier, sent as query parameter (“id”), is already assigned to an already stored AIP.

### 5.2.1 Update

All these interfaces have the aim to add the possibility to add or replace a particular part of the AIP; they require multipart parameters as described below.

- `POST /ingest/update/rights`
  This is a simple interface useful to add or replace the OWL path. The new SIP with a link to the new OWL has to be submitted with the key “sipfile”.

- `POST /ingest/update/qa`
  This interface perform the same operation as the previous one, but acts on the quality analysis reference and requires both the qa report (“file” param) and the AIP id (“id” param).

- `POST /ingest/update/usermd`
  Same as above, with as “file” the contents to be added or updated and “id” the AIP id.

## 5.3 Admin

- `GET /admin/fixitycheck/<id>`
  Performs a check matching the original checksum with the current one, returning the result in plain text.

- `POST /admin/restore`
  In case of corruption of the current file, if a second copy is stored on a LTO-5 and it is available the URI representing the tape and the position on it, administrators can ask for a restore from the URI (“from” param) to a location of file system (“to” param).

- `GET /admin/jobs/<status>`
  Queries the database and create a list of all available jobs with the chosen status. Admitted status are “ALL”, “RUNNING”, “COMPLETED”, “EXITED”; the response is formatted in plain text.

# 6 P4 User Interface

The first visible element to the final user, but the last developed, is the User Interface. As mentioned we wrote it with the support of huge technologies like J2EE, HTML5, ECMAScript, AJAX and jQuery, but in this section we want to explain the effective integration of all of them.

First of all, the User Interface (nickname P4GUI) is a new web application and it is completely separated from the P4WS; in this way, for each single installation, it is possible to decide if a Graphical Interface should be added or not, leaving only the possibility to call directly the Web Server through the REST interfaces.

Our implementation is divided in a few sections, but we want to underline only the main three:

- Access, to search into the AIPs and see the low quality videos, metadata, locations of resources and extract segments of the master quality;
- Ingest, provides a simple wizard to write metadata and ask the system to process your video;
- Admin, for authorized users it is possible to compute some operations like fixity checks, database surfing, user management and jobs monitoring.

## 6.1 Access

As mentioned, the Access section is intended for search of a resource already ingested and processed, in an easy way and with convenient graphic tools to get access to all external integrated features quickly.

### 6.1.1 Search

Search module is divided into:

- Quick Search: A simple search engine performing queries on the XML persistence DB on the Dublin Core fields “Title”, “Description”, “Format” and “Identifier”. Queries are expressed with XPath and XQuery, using data management component.
- OAI-PMH: Exposes an interface for harvesting metadata according to the OAI Protocol for Metadata Harvesting (OAI-PMH)[20].
- Advanced Search: A search interface that accesses the Solr index and offers a quicksearch as well as an expert search tool.

[20] http://www.openarchives.org/pmh/

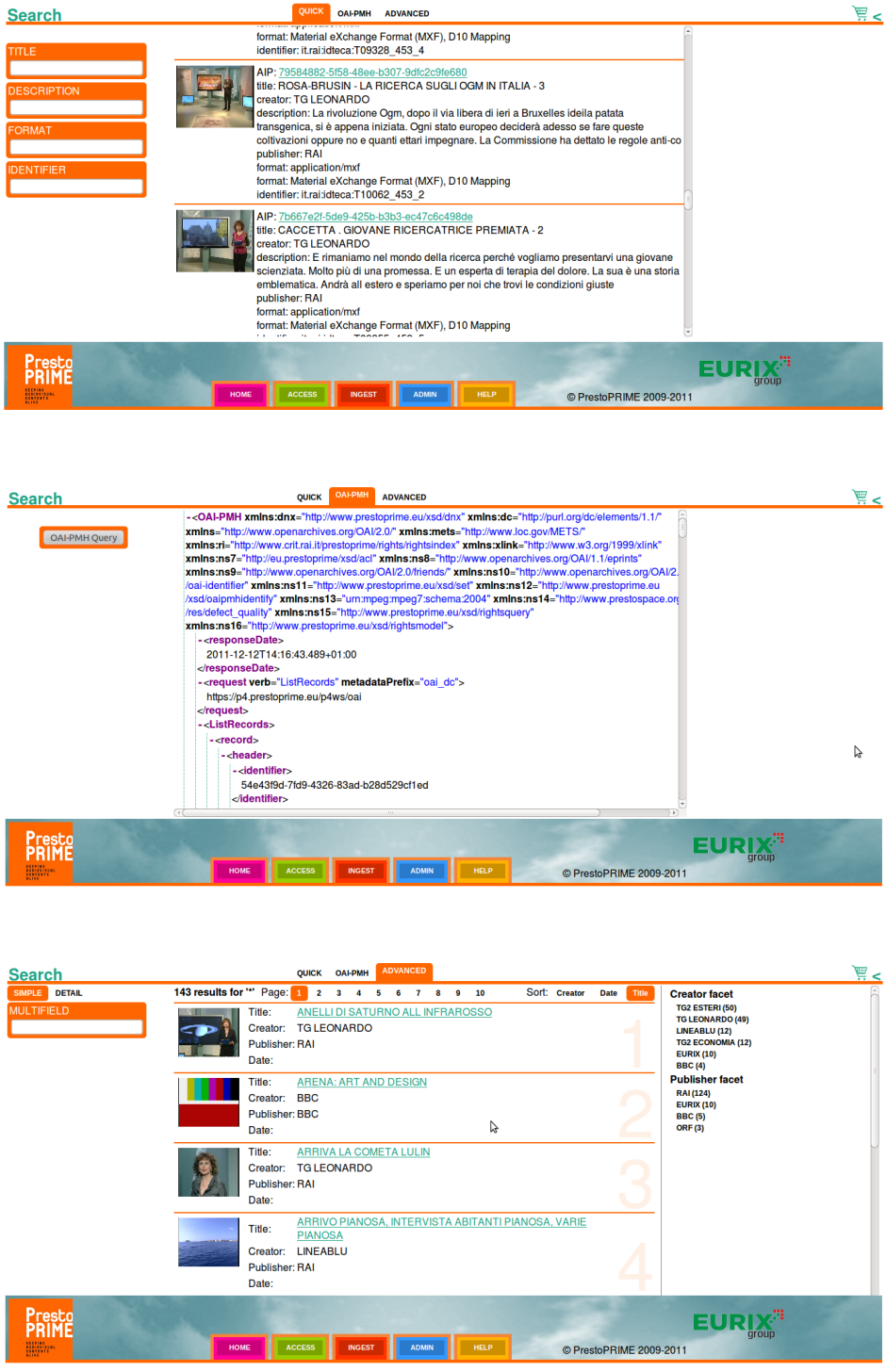


Figure 4: *P4GUI: Quick search interface on Dublin Core metadata (top); OAI-PMH standard search interface (middle); Solr based advanced search (bottom).*

### 6.1.2 Viewer

The viewer shows all the most important informations related to a single DIP: Dublin Core, rights (both graphic representation and original OWL), resources and source DIP. There are also preview of different low quality formats, to be more compatible with all the common browsers, a Java Web Start application with a Multivalent scope, able to play some obsolete video formats. Finally a list of frames lets the user to surf the video and select a portion of it to be extracted with an external tool; with a menu it is possible to monitor all the running extraction jobs.

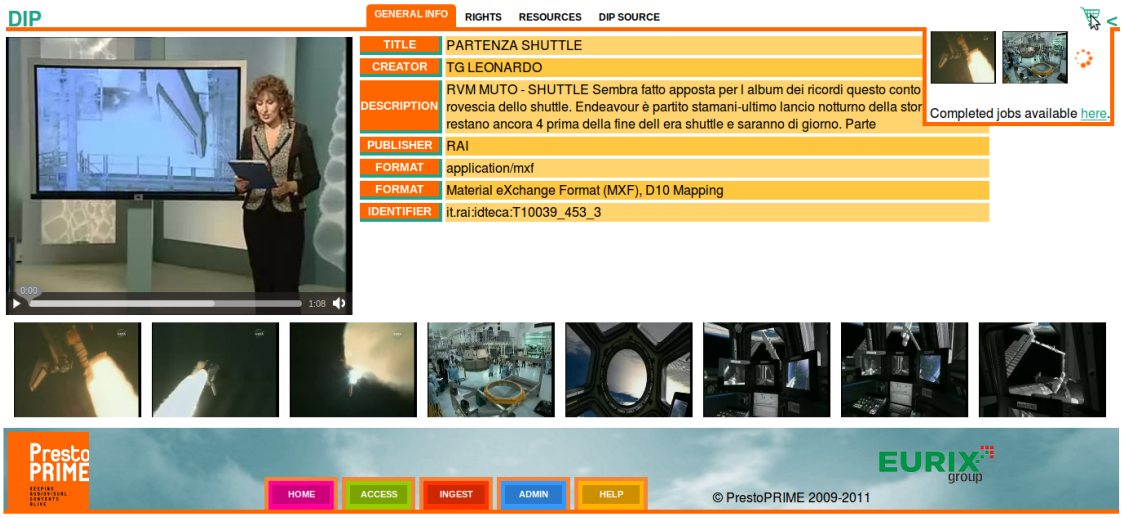


Figure 5: *P4GUI: Viewer interface in Access section*

## 6.2 Ingest

In this section we elaborate on the functions to ingest a SIP through the web interface.

### 6.2.1 New SIP

Here graphics tools help the compilation of the SIP with all the needed information as Dublin Core, rights, resources and locations and persistent storage solution. Each time, during the form compilation, it is possible to see the current SIP generated on the fly and download it to resume and finalize the ingestion again.


Figure 6: *P4GUI: Ingest main view*

### 6.2.2 Existing SIP

Uploading an existing and yet compiled XML file, the interface fill all the fields; the user can ingest the SIP ”as is” or make some last edits before sending to the P4WS.


Figure 7: *P4GUI: SIP preview before the ingestion*

## 6.3 Admin

In this last section, the administrator can keep under control all the running processes. In case of misfunctions it is possible to find the causes, in case of files corruption it is

possible to restore the original version and so on.

### 6.3.1 DB admin

This is a simple web interface that lets you see a screenshot of the permanent AIP store, giving the possibility to monitor the activity, disk usage adnd backups (both full and incremental). Moreover the same functions are available outside the browser on a Java Web Start application.

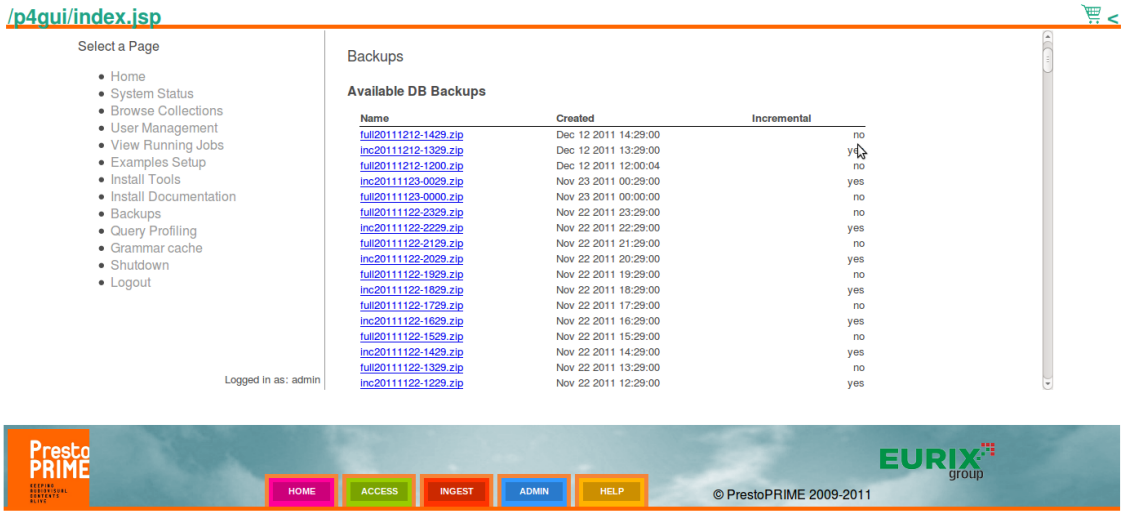


Figure 8: *P4GUI: DB Admin web interface*

### 6.3.2 Fixity Checks

The core of the P4 is the possibility, in case of corruption of data or damage of carriers, to restore the original file without any loss. As discussed, the platform gives the possibility to choose the tool to be used to store the permanent copy of the master quality file: iRODS and MServe check automatically the integrity and repair damaged files, while with LTO-5 solution, right now, requires that administrators check each single file manually and ask for a restore if there is some problem.

### 6.3.3 Jobs Monitoring

A simple list, divided into panels for all the jobs, running, completed and exited, provides informations about all the ingestion requests. The list is automatically updated to give also the progress for each job and the current step.

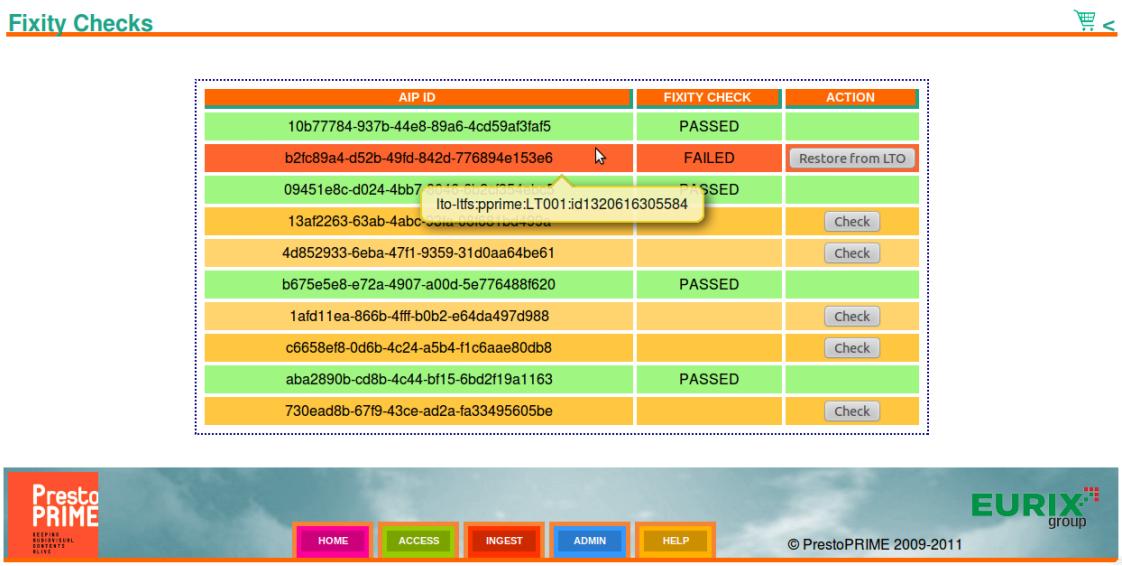


Figure 9: *P4GUI: Fixity Checks web interface*

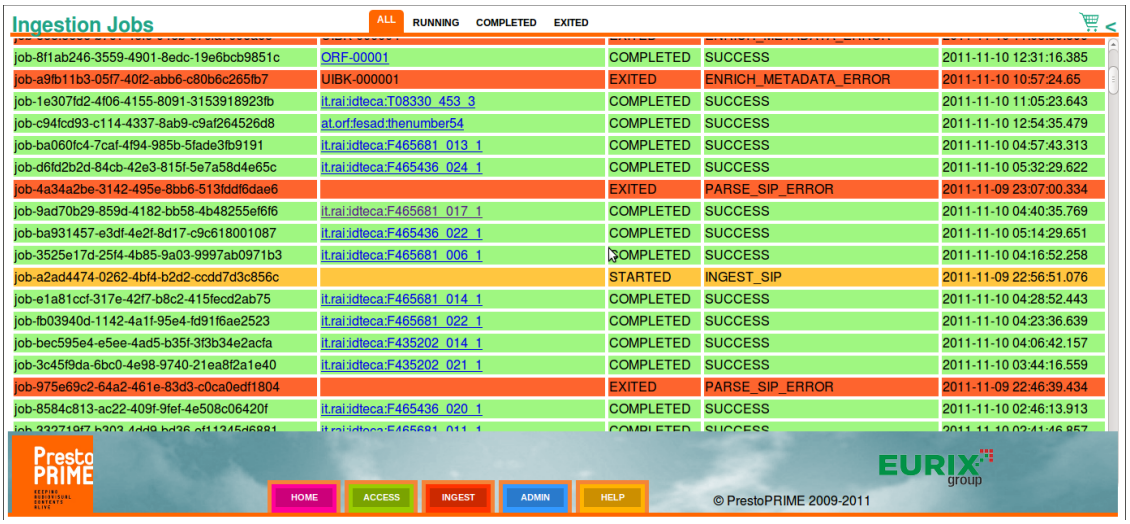


Figure 10: *P4GUI: Jobs Monitoring web interface*

### 6.3.4 DRACMA

DRACMA is an external tool provided by RAI. Its aim is extract only the segment useful for the postproduction process, preventing the download of the full master quality file, that could be too large to manage easily. For each user, P4GUI stores all the requested segments in a Derby DB and allow downloads directly from the web.

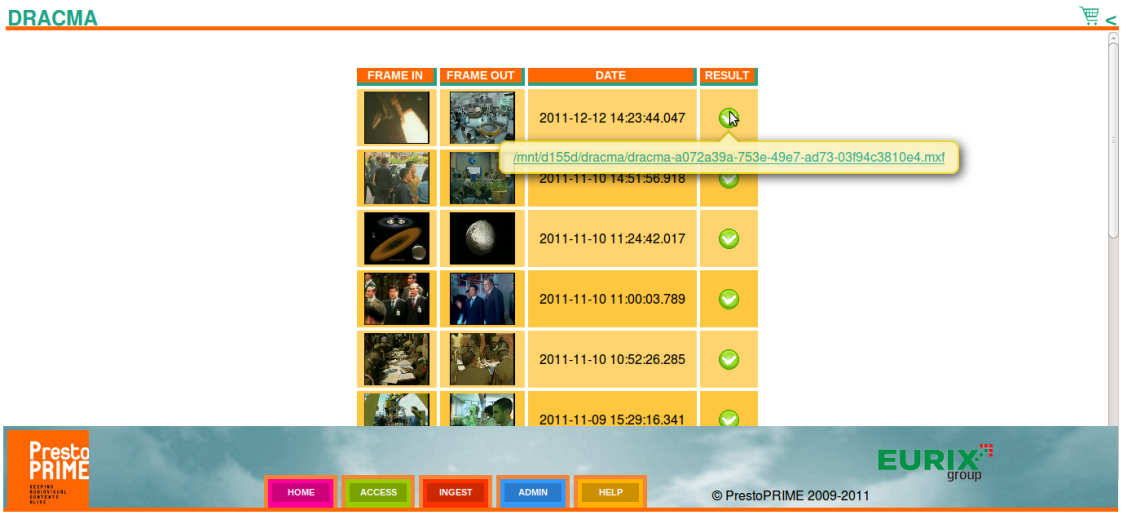


Figure 11: *P4GUI: DRACMA extracted segments*

### 6.3.5 User Management

Users roles are customizable, to control the access to each section. Roles are defined in a table of P4GUI Derby DB and could be remapped following the needs of final users.

# 7 External Tools

In this Section we describe external tools which have been developed in other PrestoPRIME technical work packages (WP2, WP3, WP4), which have been considered for integration in P4. For the first prototype the level of integration of the different tools is different, depending on several factors, such as the development status of the tool or of the P4 components for the integration and the definition of a scenario to be demonstrated at the test-bed event.

## 7.1 Preservation Planning Tools

WP2 has developed preservation planning tools that can be used alongside P4. These tools include: *iModel*, which is a simulation tool for storage, transcoding and file-format migration of digital audio visual assets; and *iWorkflow*, which is a simulation tool for digitisation/migration workflows of discrete assets (e.g. digital video tapes) and has been developed for a specific scenario at the BBC for their D3 project. These tools currently work in a 'stand-alone' mode, but work is already underway to more tightly integrate then with preservation systems in PrestoPRIME. The PrestoPRIME testbed in Turin presented the approach of using the WP2 modeling tools alongside P4 to deliver specific aspects of preservation planning.

The storage and format migration tool (*iModel*) is intended to allow a wide range of questions to be considered when planning, selecting or operating a storage and access system. The tool focuses on the storage and access to digital content in files using IT systems. The tool does not include metadata management, rights management and other issues that are of course important to consider. However, the tool does allow the following questions to be investigated:

- When storing content, how many copies should be made, what technologies should be used, how much will it cost, what are the long-term risks of losing files?
- What impact does the choice of codec (e.g. compressed or uncompressed video) have on costs and risks?
- What are the pros and cons of just in time generation of access copies compared to creating and storing a full set of proxies in advance?
- When storing data, how often should it be checked to make sure integrity is intact, and when does this become counter-productive (e.g. act of checking causes more damage than it might repair)?
- When should media migration take place (e.g. between LTO generations): regularly or at the point of obsolescence?
- What is the impact of ingest and access on shared resources for storage and data safety: what level of resources is needed to support both?

*iModel* is publicly available as open-source from IT Innovation[21], including documentation, example videos, user guide and FAQs as well as access to the source code, bug tracking and feature requests. Further details will be available also in deliverable D2.1.4 [6] which will be released at M45.

## 7.2 Storage

### 7.2.1 iRODS

iRODS masks the implementation through simple commands that are equivalent to the UNIX commands (ls, cp, mv, etc) for file management. This gives the possibility to easily manage data over a distributed storage environment. There is also a Java interface that maps all this commands in a Java object, in order to call them into the Java environment itself. Further details about iRODS can be found in internal deliverable ID3.4.3 [7], part of D3.3 [8].

### 7.2.2 LTFS Archiver

This tool, if chosen at ingestion time, stores a second copy on a LTO using the LTFS, a new file system that treats the LTO as a normal disk, allowing multiple files for each device. The tool exposes web interfaces for each of the main actions: writing file to LTO, restoring file from LTO, getting job status and results of the actions. At the end of each step, the server returns a status value plus other informations such as an identifier in the form of a URN (containing the information about the tape and the LTFS path for the video), which can be used for restore operations. Further details about the LTFS Archiver can be found in internal deliverable ID3.4.3 [7], part of D3.3 [8].

This tool has been integrated in P4 with the storage component, which provides polling methods to contact LTO server, analyses response and repeats requests until job is completed. This approach enables an asynchronous mechanism for non-blocking calls, as defined in D4.0.2 [9].

### 7.2.3 MServe

MServe is a data service providing configurable ingest, access, update and periodic workflows. The workflows are made up of tasks which can include file integrity checks and transcoding. MServe can execute these tasks on a cluster of worker nodes, thus providing a compute service as well as storage. MServe exposes REST, WebDAV and web interfaces.

[21] http://prestoprime.it-innovation.soton.ac.uk/

In the first prototype implementation which was demonstrated at the first test-bed event we proposed a minimal integration of MServe and P4, which was limited to sharing AV material through the WebDAV interface. MServe was simply used as a storage solution and the final location of AV material was stored in the AIP. A better integration of MServe and P4, including some MServe features, such as the workflow management, is planned for the next year and is described in Section 12.

### 7.2.4 DRACMA

DRACMA is a remote service implemented by RAI for indexing audiovisual files and for retrieving selected segments of master material. Even if it is not a storage component, it is strongly related to storage, so it is mentioned in this Section.

DRACMA is intended for post production and it has been integrated in the P4 user interface. Users can search AV material in the archive and then can use the P4 interface to select specific segments (even from different AIPs). Then the P4 interface is in charge of sending requests to DRACMA, monitoring the status of the jobs and allow the user to retrieve the selected segments. In order to implement this functionality, the indexing component in the ingest workflow must contact the DRACMA server and provide the required information. The DRACMA server returns a UMID identifier which is stored in the AIP and can be used to retrieve specific AV material from the DRACMA server. The DRACMA implementation is described elsewhere (see internal deliverable ID3.4.3 [7], part of D3.3 [8]), example of the P4 user interface functionalities related to DRACMA are shown in Section 6. It is worthwhile noticing that DRACMA currently supports only D10 files, so the P4 interface allows selection of segments only for MXF files with D10 mapping.

## 7.3 Advanced Search

The search component that facilitates the advanced search feature uses the SolrJ API to access a separately running instance of Solr. Its two basic functions are divided among two central classes of the component: the Indexer takes care of streaming metadata to Solr's Update Handler. The Searcher prepares incoming search requests, sends them to Solr and transforms its response for passing it further on to P4WS.

Adding data to the Solr index is handled via Java objects, that exactly reflect the index' schema. SolrJ provides Java annotations that allow to define a mapping on the object's attributes to the fields in the schema. During the ingestion workflow the Indexing Manager creates such an object for each AIP and fills in the according metadata. The Indexer takes care of connecting to the Solr Server and sends these objects as JavaBean data stream.

The Searcher class is responsible for accessing the index. It implements a method that takes a query string as input that has to be compliant to Solr's query syntax. After wrapping the query, including parameters for sorting, paging and faceting, into a SolrQuery

object, that is processed by the search server, the incoming response is processed and all relevant data is put into an object that makes use of JAXB. This way the results are ready to be marshalled into XML for returning it via P4 web server interface.

The final version of the advanced search engine will be described in deliverable D5.2.3 [4], which will be delivered at M45.

## 7.4 Multivalent

Multivalent is an external Java-based tool that is able to decode a variety of codecs without the need to resort to libraries or applications that may no longer exist. It is runnable as a Java Web Start application directly from the P4GUI Viewer. Further details can be found in internal deliverable ID3.3.2 [10], part of D3.3 [8].

## 7.5 RightsDraw

`RightsDraw` is a tool made up of a set of services developed by RAI, published under Apache Web Server, for CRUD functionalities on Rights instance documents in OWL and implementing integration with P4 regarding SIP submission, submission of rights queries in the form of particular OWL documents, and for accessing to P4 DIPs in order to import OWL instances for editing using RightsDraw. A separate package is `RightsIndex` (also from RAI) which can be fully integrated within P4, the core implementation is based on XSL (shell scripts for using the XSL stylesheets are also provided for Linux). RightsIndex provides the two following functionalities: `rightsindexinstance` (to be used in ingest workflow) and `rightscompare` (to be used on access when rights queries are submitted). For additional details about RightsDraw please refer to deliverable D4.0.2 [9].

## 7.6 Metadata Enrichment Tools

In the following we mention some tools used for metadata enrichment in the ingest workflow. These tools are available as open source from external projects or from project partners. An example of DIP showing technical metadata sections of the METS with the results of these tools can be found in Appendix C.

### 7.6.1 ffmbc and ffprobe

In the implementation we extensively used `ffmbc`[22], which is a popular open source tool providing a customization of `ffmpeg`[23] for broadcast and professional usage. ffmpeg is

[22] http://code.google.com/p/ffmbc/
[23] http://ffmpeg.org

probably the most famous open source multimedia framework, able to decode, encode and transcode, which has been used as a basis for hundreds of tools and is available for all platforms. ffmbc comes with a suite of tools, such as `ffprobe`, which is very useful for extracting general information for virtually any video format supported by ffmpeg. In the implementation we use ffprobe for extracting video and audio codecs, essence containers, duration, etc. from any AV material. This choice is motivated by the current limitation of other tools for metadata enrichment used in the prototype, which are described below, currently support MXF only files (and D10 only files for checksums). In order to process some AV materials in the test sample (specifically MXF with uncompressed video from BBC), RAI adapted a BBC patch for ffmpeg (v0.7.1) to ffmbc and provided this for the prototype. ffmbc is used for transcoding videos to low quality formats used in the access interface. The tool is integrated in P4 as a binary executable, the path of the AV file is required and several options can be specified. See the P4 tools component for further details.

### 7.6.2 MXF Technical MD Extractor

The ingestion workflow automatically runs this tools if the master quality file is in an MXF container. As the name explain, it extracts technical metadata information parsing the MXF header and is compliant to the SMPTE specification. The extracted metadata are stored into the AIP. An example of DIP showing technical metadata sections of the METS with MXF technical metadata can be found in Appendix C. Further details can be found in internal deliverable ID3.1.2 [11], part of D3.2 [12]..

### 7.6.3 D10 Sum Checker

D10SumChecker computes checksums for D10 files and stores them in the AIP. Checksums are computed for the whole AV material, for each edit unit(identified by a progressive number and a timecode), and for each frame and audio item. The results of this tool can be used, for example, to identify corrupted frames and as input for other systems (such as iRODS) capable of advanced fixity functionalities. An example of DIP showing technical metadata sections of the METS with D10 checksums can be found in Appendix C. Further details can be found in internal deliverable ID3.1.2 [11], part of D3.2 [12].

## 7.7 Quality Assessment

P4 web server provides an interface for updating AIP by adding an MPEG-7 file with QA results. The MPEG-7 file is added to the METS as an external reference. The MPEG-7 file is stored on the persistence DB and can be accessed by the user. Further details about the QA tools can be found in WP4 deliverables. The METS structure defined in the

data model allows references to external files in technical metadata sections. These file are stored in the persistence XML database described in other Sections.

## 7.8 Ting

Ting is a general service management infrastructure which monitors and manages other services. Ting is configured using management policies and flexible service level agreements (SLAs). It provides functions to create and publish new SLA templates; agree new SLAs and deploy associated services; monitor and manage existing service instances. In addition it has been integrated with iModel (see 7.1) to provide decision support for policy makers. This tool will be considered for integration with P4 in the next year, as described in Section 12.

# 8 Installation Guide

This guide shall outline how to setup and run P4 and has been tested on Ubuntu 10.04+ 32bit Desktop Edition. Basically, a P4 installation can be split into two parts the web service, consisting of p4ws and the eXist database, and the user interface p4gui. The subsections 8.1 and 8.2 explain common steps that are needed for both components and the last two sections give specific instructions on each of them. The configuration part of the guide is still under development, an improved version will be available in deliverable D5.2.3, which will describe the final prototype implementation.

## 8.1 Setup the Java 7 Runtime Environment

Download and install Java JDK 7 (the code has been tested with Oracle JVM) and configure the Java environment:

- `$JAVA_HOME` pointing to the JDK7 installation dir
- `$PATH` must include `$JAVA_HOME/bin`

## 8.2 Install Tomcat and Apache

1. Download and install Apache Tomcat 7 according to the instructions on the Tomcat website.
2. Adjust the configuration as follows:
   (a) Enable SSL in `$TOMCAT_HOME/conf/server.xml` and add the path to the keystore file (provided as `_.keystore`)

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="/usr/local/prestoprime/P4/.keystore"
URIEncoding="UTF-8"/>
```

   (b) Set `URIEncoding="UTF-8"` for each Connector in `$TOMCAT_HOME/conf/server.xml` as seen in the example above
   (c) Set users in `$TOMCAT_HOME/conf/tomcat-users.xml`

```
<role rolename="Admin"/>
<role rolename="manager-gui"/>
<user username="pprime"
 password="pprime09" roles="Admin"/>
<user username="admin"
 password="pprime09" roles="manager-gui"/>
```

3. Install Apache Web Server and Mod-JK:

   ```
   sudo apt-get install apache2 libapache2-mod-jk
   ```

4. Mod-JK is used to forward requests on port 80 to Tomcat for specified contexts. In newer versions of Ubuntu after installing libapache2-mod-jk there is already a worker defined, called 'ajp13_worker', that will take care of this. In this case you will just have to mount the contexts on this worker by adding the following lines (or at least one of them depending on which components of P4 will be running on the machine) to `/etc/apache2/sites-available` in the definition of the virtual host:

   ```
   JkMount /p4gui/* ajp13_worker
   JkMount /p4ws/* ajp13_worker
   ```

5. Restart Apache by running

   ```
   /etc/init.d/apache2 restart
   ```

6. If you get an error telling that no Mod-JK worker is defined add the following line at the very top of `/etc/apache2/sites-available` and restart Apache:

   ```
   JkWorkersFile /etc/libapache2-mod-jk/workers.properties
   ```

## 8.3 P4 Web Service

### 8.3.1 Install Utilities

Install the graphviz package:

```
sudo apt-get install graphwiz
```

Other necessary tools are included in P4 (as 32bit binaries). Anyway it could help for debugging to install the following packages using the same command as above:

- ffmpeg
- ffmbc
- ffmpeg2theora
- xsltproc

### 8.3.2 Deploy and Configure the Web Applications

1. Copy the following two web application archives to `$TOMCAT_HOME/webapps`:
   - p4ws.war: P4 Web Service

- p4db.war: P4 Persistence DB (eXist XML DB)

2. Start tomcat by executing `$TOMCAT_HOME/bin/startup.sh`
   The web application archives are thereby automatically extracted to the respective folders in `$TOMCAT_HOME/webapps`

3. Configure the database
   (a) Open the DB admin page and log in as user ”admin” with an empty password on:
   `http://localhost:8080/p4db/admin`
   Select the ”Browse Collections” page
   (b) Navigate to `/db/system/config` and create the collection `db`
   (c) Go into the created collection `/db/system/config/db` and add a trigger enabling AIP versioning by uploading the file `collection.xconf`
   (d) Add scheduler jobs for incremental and full backups in `$TOMCAT_HOME/webapps/p4db/WEB-INF/conf.xml`:

```
<!-- perform an incremental backup every hour -->
<job type="system" name="check1"
   class="org.exist.storage.ConsistencyCheckTask"
   cron-trigger="0 29 * * * ?">
   <parameter name="output" value="export"/>
   <parameter name="backup" value="yes"/>
   <parameter name="incremental" value="yes"/>
   <parameter name="incremental-check" value="no"/>
   <parameter name="max" value="2"/>
   </job>
<!-- perform a full backup every twelve hours -->
<job type="system" name="check2"
   class="org.exist.storage.ConsistencyCheckTask"
   cron-trigger="0 0 0/12 * * ?">
   <parameter name="output" value="export"/>
   <parameter name="backup" value="yes"/>
   <parameter name="incremental" value="no"/>
</job>
```

4. Restart Tomcat
   (Stopping is done with the script `$TOMCAT_HOME/bin/shutdown.sh`

5. Configure the P4 Web Service
   (a) Look at the init parameters in `$TOMCAT_HOME/webapps/p4ws/WEB-INF/web.xml` and create the following folders for P4 storage:
   - P4StorageVolume (default: /storage/pprime/p4test)

- P4StorageFolder (default: p4store)
- PermanentStorageVolume (default: /storage/pprime/p4test)
- PermanentStorageFolder (default: permstore)
- MServeVolume (default: /storage/pprime/p4test)
- MServeFolder (default: mservestore)

(b) Publish the folders P4StorageFolder, PermanentStorageFolder and MServeFolder via Apache Web Server by creating symlinks, e.g.:

```
> cd /var/www
> ln -s /storage/pprime/p4test/p4store
```

(c) Enable reading from all folders and enable following symlinks in apache configuration

(d) Set "p4.prestoprime.eu" as alias for "localhost" by adding the following line to `/etc/hosts`:

```
127.0.0.1 p4.prestoprime.eu
```

(e) Optional: In order to change the name of the p4 server, edit the init param "P4WebServer" (and the others) in p4ws/WEB-INF/web.xml

6. Install Solr

(a) Create the directory `/usr/local/prestoprime/P4/` and allow writing to this path for the user that runs Tomcat

(b) Extract `p4solr.tar.gz` and move the contained folder `p4solr` to this path which should yield the directory `/usr/local/prestoprime/P4/p4solr`, i.e. `$SOLR_HOME`

(c) Create a new context in Tomcat by creating the file `$TOMCAT_HOME/conf/Catalina/localhost/solr.xml` and setting its content to:

```
<?xml version="1.0" encoding="utf-8"?>
<Context
  docBase="/usr/local/prestoprime/P4/p4solr/solr.war"
  crossContext="true">
 <Environment
  name="solr/home"
  type="java.lang.String"
  value="/usr/local/prestoprime/P4/p4solr"
  override="true"/>
</Context>
```

(d) Edit the file `$SOLR_HOME/conf/solrconfig.xml` so that the value of `dataDir` points to the absolute path of `$SOLR_HOME/data`

(e) Restart Tomcat

(f) Check if Solr runs by accessing:
`http://localhost:8080/solr`

## 8.4 P4 User Interface

1. Copy `p4gui.war` to `$TOMCAT_HOME/webapps`
2. Configure properties in the `p4gui.xml` in `conf` folder like:

```
<properties>
  <comment>Properties for p4gui</comment>
  <entry key="keystore">
    /usr/local/prestoprime/P4/.keystore
  </entry>
  <entry key="derby.home">
    /usr/local/prestoprime/P4/p4guiDB
  </entry>
  <entry key="p4ws.server">p4.prestoprime.eu</entry>
  <entry key="ingestion.file.list">
    /usr/local/prestoprime/P4/ingestionFileList.txt
  </entry>
  <entry key="dracma.server">10.2.6.73:8080</entry>
  <entry key="dracma.storage">/mnt/d155d/dracma</entry>
</properties>
```

*Please, note that if the same `.keystore` file of the P4WS is not found by the P4GUI, the entire user interface will not work at all.*

3. Set ”p4.prestoprime.eu” as alias for the machine running p4ws by adding the following line to `/etc/hosts` (assuming that p4ws runs on localhost):

```
127.0.0.1 p4.prestoprime.eu
```

4. Start Tomcat and access `http://localhost/p4gui`

# 9 Software Documentation

The software documentation is available on-line at the following link:

http://prestoprime.eurixgroup.com/p4doc/api/

The documentation has been generated using Doxygen[24], a multi-language documentation system available with GPL license. Doxygen is available for Linux, Windows and Mac and is highly configurable for what concerns the layout, the choice of software modules to be documented and the level of information shown for each source file.

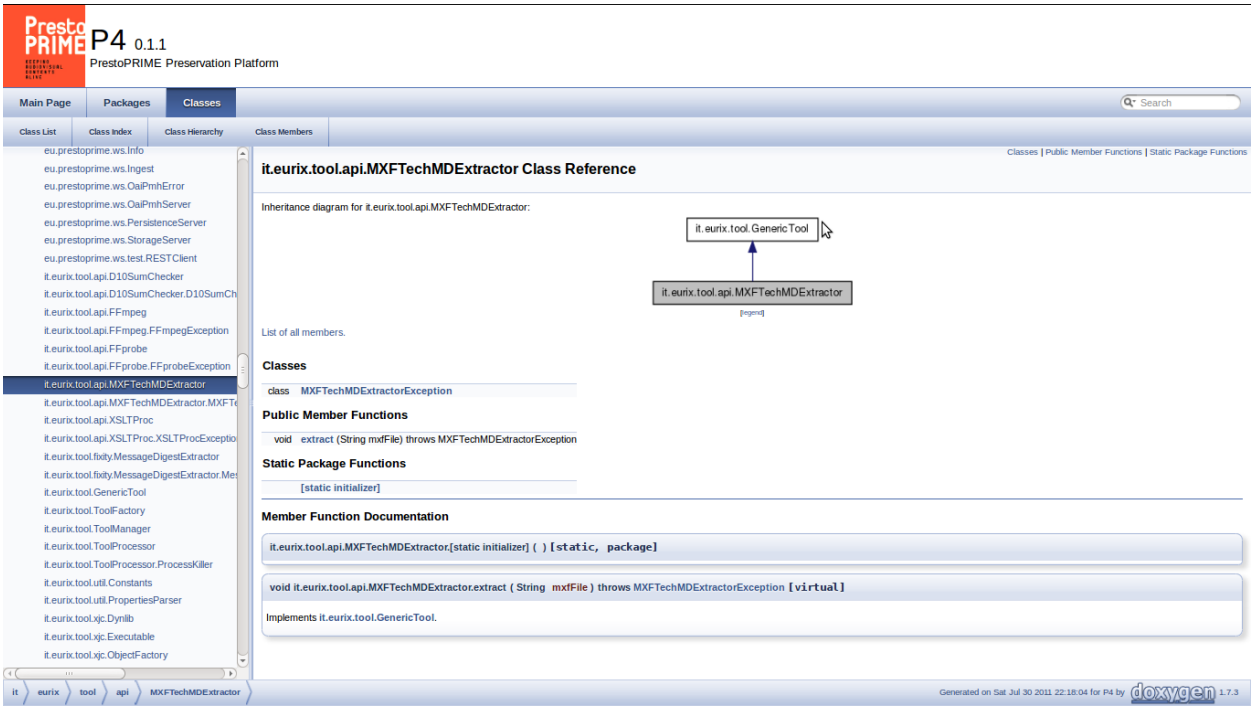


Figure 12: *P4 on-line documentation: class with public methods and interfaces*

The on-line documentation is made up of HTML pages with references to all classes and packages, to original source code, including diagrams and relationships among classes. The documentation includes also a search functionality for retrieving specific source files.

An examples from the on-line documentation is depicted in Figure 12.

[24] http://www.doxygen.org

# 10 Software License

The PrestoPRIME reference implementation is built from code developed within the project and makes use of external libraries with their own associated license. The choice of dependencies was driven by the selection of free public libraries with an open source license. The compatibility of different licenses for the core software and external dependencies is still under evaluation and detailed information will be provided in D5.2.3, when the final version of the integrated prototype will be released.

However, according to the project description of work and the previous deliverables (see for example D5.2.1 or D4.0.2), EURIX will release the source code for the PrestoPRIME preservation platform under the terms of the General Public License version 3.0[25]. In order to address legal issues possibly coming up from the use of GPL-incompatible libraries[26], the appropriate additional permission will be granted under section 7 of GPL3.

We are also evaluating other licenses in the GNU family, for example the GNU Affero GPL[27], which is a free, copyleft license for software and other kinds of works, specifically designed to ensure cooperation with the community in the case of network server software. Multi-licensing could also be considered according to specific requirements.

# 11 Demonstration of prototype

The firt prototype implementation described in this document has been demonstrated at the last test-bed event.

The demonstration was based on a AV data sample including videos from several project partners (RAI, BBC, ORF, University of Innsbruck), the size of the collected sample was about 9 TB, including MXF and AVI videos. Each partner prepared the SIP files for the related AV material, according to the AV data model specification described before. The test users could ingest some of these SIP files or create them from scratch using the prototype interface. The ingest workflow described before was run on all videos and real time monitoring was available. The total number of submitted jobs was about 300, processing about 2 TB videos (other sample material was used for testing before the event). Finally, the results of ingestion were available to the users through access interface. Other tools were also tested in parallel: for example the RightsDraw tool was used to search in the archive using rights information, edit the rights and update the AIP or user annotations from VUA tagging game were also to existing AIP, or iRODS was used to demonstrate automatic replication of ingested material, just to mention a few.

For more details about the demo shown at the test-bed please refer to deliverable D5.4.3 [3].

[25] http://www.gnu.org/licenses/gpl-3.0.txt
[26] http://www.gnu.org/licenses/gpl-faq.html#GPLIncompatibleLibs
[27] http://www.gnu.org/licenses/agpl.html

# 12 Future work

In this Section we describe the planned activities for the development of the preservation system, concerning missing components and a better integration with all the tools and services provided by project partners. The result will be reported in deliverable D5.2.3 at M45.

## 12.1 P4 development

The development activity in the next year will focus on the feedbacks from the users at the last test-bed event and will affect all core components of the platform. Concerning ingest, we plan to improve the interface and we will add also the possibility to submit SIP files with no AV content (this requirement is useful, for example, when a new AIP must be created in the archive containing just the metadata and the rights information). Concerning access, we plan to improve the interface and fix some issues in the DIP presentation which could be identified only in the test-bed session, when different sources of SIP were used extensively. Concerning preservation planning, we will evaluate available tools, focusing on the implementation of scheduled tasks related to preservation, such as fixity checks or storage migration. We will benefit from tools available from WP2. The data management component is quite complete, possible improvements could be possible in order to support additional formats.

In the following Sections we describe possible improvements for specific issues. For additional details about open issues and bugs in the prototype implementation please refer to D5.4.3 [3]. All these issues have been already identified during the test-bed session, some of them were already fixed on the fly, while others require deeper investigation and tests.

## 12.2 Workflow

In the final phase of the project we plan to integrate P4, MServe and Ting more closely. Specifically we plan to move the majority of the burden of running workflows from P4 to MServe to make use of the dynamically reconfigurable workflow system already existing in MServe. This will also entail: creating some additional tasks in MServe to extend the current workflows; writing code for P4 to query the properties (e.g. MD5 checksums) of files stored and processed in MServe using the existing REST interface.

By moving the workflows and associated tasks into MServe we also take advantage of the distributed job submission system already demonstrated, making the entire system more suited to large deployments. Job tasks run from within MServe are already monitored by the service management framework, Ting. In addition we will be instrumenting P4 to provide some additional metrics to Ting in order to gather more comprehensive data on

the performance of the system.

## 12.3 Search Component

While during the test beds only Dublin Core data was indexed, in the next versions of the search component the Solr Schema will be extended to cater for other common parts of P4's AIPs, e.g. technical metadata or user annotations added via the system by VUA. This will allow much more complex queries for instance on quality measures and vastly improve the usefulness of the search interface for the user.

A shortcoming of the quick search functionality was identified in its sensitivity to wrong query syntax. For instance a user would just get no results at all if a reserved character or keyword occurred in the search term as the query was just passed through to Solr. The comprehensive syntax was thought to be a powerful tool but a normal user would simply not expect the need to read a manual when using a quick search bar. The conclusion taken from this is a reconfiguration of Solr: another query parser (which is implemented by the DismaxQueryParserPlugin), will cater for the quick search feature and will automatically strip reserved characters from the search term. This way the tool becomes more easy to use but will also accept just a subset of the query language. It simply searches for whole terms occurring in any of the fields and caters for inclusion or exclusion of single terms when $+$ or $-$ are prepended. Field-specific and wildcard queries will be explicitly provided by the advanced search though. This will also induce a change of the interface in P4WS. The quick search method will be changed as described and another GET method will be introduced to fulfill the advanced search tasks.

These in turn will be complemented by a new graphical user interface: the multifield search page is now a static set of input fields and a query is dynamically constructed as a simple disjunction of the input values from these fields. Part of the next iteration in development will be building a fully dynamic interface that lets a user add input fields, set field names and choose the boolean operators that will be used to concatenate terms when the query is built. As technical metadata will contain various numeric values there is also the need to search on ranges.

## 12.4 Storage

The expected improvement of the storage component in the next year development will focus on a better integration with MServe and iRODS. In particular, we plan to make extensive use of Java APIs also for iRODS, in order to improve the current interface among the systems. A possible solution is the adoption of Jargon API[28] for full configuration of iRODS from Java code, with the possibility to execute all tasks concerning files from within the P4 storage component.

[28] https://www.irods.org/index.php/Jargon

# 13 Conclusions

In this document we presented the first integrated prototype of the PrestoPRIME preservation platform.

We first described the relationships with previous deliverables, mainly the design of the architecture and the AV data model and then we discussed the implementation of the platform and the technologies used. We presented the three main components of the system: the core modules, the REST web server and the user interface. For each core module we briefly reported the main features, then we referenced all external tools which have been integrated and those which will be considered for a possible integration in the next year. Then we provided information about the software delivery and installation. Finally we briefly discussed about future developments.

The further development of the prototype will be documented in D5.2.3, which will be released at M45 and will represent the final release of the PrestoPRIME integrated platform.

# 14 Glossary

***AIP***: Archive Information Package.

***AJAX***: Asynchronous JavaScript and XML.

***API***: Application Programming Interface.

***AV***: AudioVisual contents.

***BBC***: British Broadcasting Corporation.

***BPMN***: Business Process Model and Notation.

***CRUD***: Create - Read - Update - Delete.

***CSS***: Cascading Style Sheet.

***DB***: DataBase.

***DC***: Dublin Core.

***DIP***: Dissemination Information Package.

***DNX***: ExLibris XML Schema based on PREMIS.

***FAQ***: Frequently Asked Questions.

***GPL***: General Public License.

***HTML***: HyperText Markup Language.

***HTTP***: HyperText Transmission Protocol.

***IDE***: Integrated Development Environment.

***LTFS***: IBM Linear Tape File System.

***LTO***: Linear Tape-Open.

***J2SE***: Java Standard Edition version 2.

***J2EE***: Java Enterprise Edition version 2.

***JAR***: Java ARchive.

***JAX-RS***: Java API for RESTful Web Services.

***JAXB***: JAva XML Binding.

***JDK***: Java Development Kit.

***JMS***: Java Message Service.

***jPDL***: jBPM Process Definition Language.

***JVM***: Java Virtual Machine.

***METS***: Metadata Encoding and Transmission Standard.

***MD***: MetaData.

***MIME***: Multipurpose Internet Mail Extensions.

***MPEG***: Motion Pictures Expert Group.

***MXF***: Material Exchange Format.

***NFS***: Network File System.

***OAI-PMH***: Open Archives Initiative - Protocol for Metadata Harvesting.

***OAIS***: Open Archival Information System.

***ORF***: Österreichischer RundFunk (Austrian Broadcaster).

***OS***: Operating System.

***OWL***: Web Ontology Language.

***P4***: PrestoPRIME Preservation Platform.

***P4WS***: P4 Web Server.

***P4GUI***: P4 Graphical User Interface.

***QA***: Quality Assessment.

***RAI***: RadioTelevisione Italiana.

***REST***: REpresentational State Transfer.

***SIP***: Submission Information Package.

***SLA***: Service Level Agreements.

***SQL***: Structured Query Language.

***SSL***: Secure Socket Layer.

***TB***: TeraByte.

***UMID***: Unique Material Identifier.

***UML***: Unified Modeling Language.

***URI***: Uniform Resource Identifier.

***URL***: Uniform Resource Locator.

***URN***: Uniform Resource Name.

***W3C***: WWW Consortium.

***WADL***: Web Application Description Language.

***WebDAV***: Web-based Distributed Authoring and Versioning.

***WP***: Work Package.

***XML***: eXtensible Markup Language.

***XSL***: eXtensible Stylesheet Language.

***XSLT***: XSL Transformation.

# References

[1] PrestoPRIME Project. Deliverable 5.2.1: Architecture Design of the Integration Framework. http://www.prestoprime.eu/project/public.en.html, December 2010.

[2] PrestoPRIME Project. Deliverable 2.1.3: AV Data Model: Final Specification. http://www.prestoprime.eu/project/public.en.html, December 2010.

[3] PrestoPRIME Project. Deliverable 5.4.3: Report on Evaluation Phase 2. http://www.prestoprime.eu/project/public.en.html, December 2011.

[4] PrestoPRIME Project. Deliverable 5.2.3: Advanced Prototype of Open PrestoPRIME Reference Implementation and Audiovisual Registry Proof of Concept. to appear September 2012.

[5] PrestoPRIME Project. Deliverable 5.1.1: Definition of Scenarios. http://www.prestoprime.eu/project/public.en.html, December 2009.

[6] PrestoPRIME Project. Deliverable 2.1.4: Tools for quantitative comparison of preservation strategies. to appear September 2012.

[7] PrestoPRIME Project. Internal Deliverable 3.4.3: First version of prototype for distributing and storing audiovisual content using federated storage and compute services. Part of deliverable D3.3, http://www.prestoprime.eu/project/public.en.html, December 2011.

[8] PrestoPRIME Project. Deliverable 3.3: Second version of the audio-visual preservation toolkit. http://www.prestoprime.eu/project/public.en.html, December 2011.

[9] PrestoPRIME Project. Deliverable 4.0.2: Metadata and Rights APIs and Service Interfaces. http://www.prestoprime.eu/project/public.en.html, December 2010.

[10] PrestoPRIME Project. Internal Deliverable 3.3.2: First Version of Multivalent Media Engines for Common Media Formats. Part of deliverable D3.3, http://www.prestoprime.eu/project/public.en.html, December 2011.

[11] PrestoPRIME Project. Internal Deliverable 3.1.2: First Version of Migration-based Preservation Toolkit. Part of deliverable D3.2, http://www.prestoprime.eu/project/public.en.html, December 2010.

[12] PrestoPRIME Project. Deliverable 3.2: First version of the audiovisual preservation toolkit. http://www.prestoprime.eu/project/public.en.html, December 2011.

# Appendices

## A Data Model Namespaces

The namespaces used in SIP, AIP and DIP for the PrestoPRIME data model are listed in Table 1. For some namespaces the prefix has not been assigned yet, so it is not shown. For such namespaces the value is automatically generated by the application when the XML is serialized (see Appendix C for example).

| | |
|---|---|
| http://www.loc.gov/METS/ | mets |
| http://purl.org/dc/elements/1.1/ | dc |
| http://www.prestoprime.eu/xsd/dnx | dnx |
| http://www.w3.org/1999/xlink | xlink |
| http://www.openarchives.org/OAI/2.0/ | (default) |
| http://www.openarchives.org/OAI/2.0/oai-identifier | |
| http://www.openarchives.org/OAI/1.1/eprints | |
| http://www.crit.rai.it/prestoprime/rights/rightsindex | ri |
| http://www.prestoprime.eu/xsd/rightsmodel | |
| http://www.prestospace.org/res/defect_quality | TBC |
| http://www.prestoprime.eu/xsd/oaipmhidentify | |
| urn:mpeg:mpeg7:schema:2004 | TBC |
| http://eu.prestoprime/xsd/acl | |
| http://www.prestoprime.eu/xsd/set | |

Table 1: *PrestoPRIME Data Model Namespaces*

# B SIP Examples

In the following we present some examples of SIP files compliant to the PrestoPRIME AV data model which can be ingested into the platform.

## B.1 Example 1

This is an example of SIP ingested by RAI during the Test-bed.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<mets:mets xmlns:dnx="http://www.prestoprime.eu/xsd/dnx" xmlns:dc="http://purl.org/dc/elements
    /1.1/" xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:mets="http://www.loc.gov/METS/"
    xmlns:ri="http://www.crit.rai.it/prestoprime/rights/rightsindex" xmlns:xlink="http://www.w3.
    org/1999/xlink" xmlns:ns7="http://eu.prestoprime/xsd/acl" xmlns:ns8="http://www.openarchives.
    org/OAI/1.1/eprints" xmlns:ns9="http://www.openarchives.org/OAI/2.0/friends/" xmlns:ns10="
    http://www.openarchives.org/OAI/2.0/oai-identifier" xmlns:ns11="http://www.prestoprime.eu/xsd
    /set" xmlns:ns12="http://www.prestoprime.eu/xsd/oaipmhidentify" xmlns:ns13="
    urn:mpeg:mpeg7:schema:2004" xmlns:ns14="http://www.prestospace.org/res/defect_quality"
    xmlns:ns15="http://www.prestoprime.eu/xsd/rightsquery" xmlns:ns16="http://www.prestoprime.eu/
    xsd/rightsmodel" OBJID="urn:rightsdraw#2011-11-06T06:35:06.31418" LABEL="pPRIME SIP">
    <mets:metsHdr CREATEDATE="2011-11-06T06:35:06">
        <mets:agent ROLE="CREATOR">
            <mets:name>pprime</mets:name>
        </mets:agent>
    </mets:metsHdr>
    <mets:dmdSec ID="d001">
        <mets:mdWrap MDTYPE="DC">
            <mets:xmlData>
                <dc:record>
                    <dc:title>USCITA PORTOFINO, VARIE MARE, AVVICINAMENTO DELFINI, PESCATA
                        NASELLO</dc:title>
                    <dc:creator>LINEABLU</dc:creator>
                    <dc:publisher>RAI</dc:publisher>
                    <dc:identifier>it.rai:idteca:F465681_010_1</dc:identifier>
                </dc:record>
            </mets:xmlData>
        </mets:mdWrap>
    </mets:dmdSec>
    <mets:amdSec ID="a001">
        <mets:rightsMD ID="ppavro-rights-001" ADMID="acl001">
            <mets:mdRef xlink:href="http://10.2.6.192/rightsdraw/lau/export/PPRIME_it.
                rai:idteca:F465681_010_1.owl" LOCTYPE="URL" MIMETYPE="application/owl+xml" SIZE="
                3767" CHECKSUM="c167412167166444a51282807bc051a0" CHECKSUMTYPE="MD5" MDTYPE="
                OTHER" OTHERMDTYPE="PPAVRO"/>
        </mets:rightsMD>
        <mets:rightsMD ID="acl001">
            <mets:mdWrap MDTYPE="OTHER" OTHERMDTYPE="PPACL">
                <mets:xmlData>
                    <ns7:ACL>
                        <ns7:Entry type="user" permissions="rw">pprime</ns7:Entry>
                        <ns7:Entry type="group" permissions="rw">team</ns7:Entry>
                        <ns7:Entry type="group" permissions="r">partners</ns7:Entry>
                    </ns7:ACL>
                </mets:xmlData>
            </mets:mdWrap>
        </mets:rightsMD>
        <mets:sourceMD ID="sourcemd-001">
            <mets:mdWrap MDTYPE="DC">
                <mets:xmlData>
                    <dc:record>
                        <dc:identifier>00100096698874</dc:identifier>
                    </dc:record>
```

```
                </mets:xmlData>
            </mets:mdWrap>
        </mets:sourceMD>
    </mets:amdSec>
    <mets:fileSec>
        <mets:fileGrp>
            <mets:file ID="fid1" ADMID="acl001" MIMETYPE="application/mxf" CHECKSUM="0383
                a5f8222d6c74074f09bf9299801f" CHECKSUMTYPE="MD5">
                <mets:FLocat xlink:href="/mnt/d155a/RAI/54/00100096698874/ROBOT1294652415.29/
                    edobs_edt/01+F465681_010_1+0009669887/01+F465681_010_1+0009669887.mxf"
                    LOCTYPE="OTHER" OTHERLOCTYPE="FILE"/>
            </mets:file>
        </mets:fileGrp>
    </mets:fileSec>
    <mets:structMap TYPE="One-to-One" LABEL="PPRIME-AV">
        <mets:div TYPE="Audiovisual">
            <mets:fptr FILEID="fid1"/>
        </mets:div>
    </mets:structMap>
</mets:mets>
```

## B.2 Example 2

The following is an exhaustive example of SIP ingested during the Test-bed by BBC.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<mets:mets xmlns:dnx="http://www.prestoprime.eu/xsd/dnx" xmlns:dc="http://purl.org/dc/elements
    /1.1/" xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:mets="http://www.loc.gov/METS/"
    xmlns:ri="http://www.crit.rai.it/prestoprime/rights/rightsindex" xmlns:xlink="http://www.w3.
    org/1999/xlink" xmlns:ns7="http://eu.prestoprime/xsd/acl" xmlns:ns8="http://www.openarchives.
    org/OAI/1.1/eprints" xmlns:ns9="http://www.openarchives.org/OAI/2.0/friends/" xmlns:ns10="
    http://www.openarchives.org/OAI/2.0/oai-identifier" xmlns:ns11="http://www.prestoprime.eu/xsd
    /set" xmlns:ns12="http://www.prestoprime.eu/xsd/oaipmhidentify" xmlns:ns13="
    urn:mpeg:mpeg7:schema:2004" xmlns:ns14="http://www.prestospace.org/res/defect_quality"
    xmlns:ns15="http://www.prestoprime.eu/xsd/rightsquery" xmlns:ns16="http://www.prestoprime.eu/
    xsd/rightsmodel" OBJID="uk:bbc:rd:pprime:test2011:002" LABEL="pPRIME SIP">
    <mets:metsHdr CREATEDATE="2011-11-08T12:30:00">
        <mets:agent ROLE="CREATOR" TYPE="ORGANIZATION">
            <mets:name>BBC RD</mets:name>
        </mets:agent>
    </mets:metsHdr>
    <mets:dmdSec ID="DMD-001">
        <mets:mdWrap MDTYPE="DC">
            <mets:xmlData>
                <dc:record>
                    <dc:title>GRANDSTAND</dc:title>
                    <dc:creator>BBC</dc:creator>
                    <dc:publisher>BBC</dc:publisher>
                    <dc:identifier>uk:bbc:prog_no:LSAG135B:001</dc:identifier>
                    <dc:language>EN</dc:language>
                </dc:record>
            </mets:xmlData>
        </mets:mdWrap>
    </mets:dmdSec>
    <mets:amdSec>
        <mets:sourceMD ID="SOURCEMD-001">
            <mets:mdWrap MDTYPE="DC">
                <mets:xmlData>
                    <dc:record>
                        <dc:format>Panasonic D3 Tape</dc:format>
                        <dc:identifier>uk:bbc:spool_no:DA034600:001</dc:identifier>
                    </dc:record>
                </mets:xmlData>
```

```
            </mets:mdWrap>
        </mets:sourceMD>
    </mets:amdSec>
    <mets:amdSec>
        <mets:techMD ID="TECHMD-001">
            <mets:mdWrap MDTYPE="DC">
                <mets:xmlData>
                    <dc:record>
                        <dc:creator>uk:bbc:ia:d3_project:channel2</dc:creator>
                        <dc:description>BBC Archive MXF file created during the BBC IA D3 Project
                            .</dc:description>
                        <dc:date>2010-05-11T17:08:08</dc:date>
                        <dc:format>application/mxf</dc:format>
                    </dc:record>
                </mets:xmlData>
            </mets:mdWrap>
        </mets:techMD>
    </mets:amdSec>
    <mets:fileSec>
        <mets:fileGrp>
            <mets:file ID="FID-001" ADMID="SOURCEMD-001_TECHMD-001" MIMETYPE="application/mxf"
                SIZE="67596930985" CHECKSUM="0e0ddbda5bd5bfc5dc768eb8ae39f3fa25f2eb21"
                CHECKSUMTYPE="SHA-1">
                <mets:FLocat xlink:href="/mnt/d155a/BBC/bigs/LTA00853502.mxf" LOCTYPE="OTHER"
                    OTHERLOCTYPE="FILE"/>
            </mets:file>
        </mets:fileGrp>
    </mets:fileSec>
    <mets:structMap TYPE="One-to-One" LABEL="PPRIME-AV">
        <mets:div DMDID="DMD-001" TYPE="AudioVisual">
            <mets:fptr FILEID="FID-001"/>
        </mets:div>
    </mets:structMap>
</mets:mets>
```

# C DIP Example

In the following we present an example of DIP which can be accessed by the user through the platform interface. Compared to the SIP file, the DIP includes the results of ingestion process (such as technical metadata extracted during the ingestion) and updated references for all audiovisual files.

We use the same wrapper for SIP, AIP and DIP, based on the METS schema.

```
<mets:mets xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:dc="http://purl.org/dc/elements
    /1.1/" xmlns:ns16="http://www.prestoprime.eu/xsd/rightsmodel" xmlns:ns14="http://www.
    prestospace.org/res/defect_quality" xmlns:ns15="http://www.prestoprime.eu/xsd/rightsquery"
    xmlns:ns9="http://www.openarchives.org/OAI/2.0/friends/" xmlns:ri="http://www.crit.rai.it/
    prestoprime/rights/rightsindex" xmlns:ns12="http://www.prestoprime.eu/xsd/oaipmhidentify"
    xmlns:ns13="urn:mpeg:mpeg7:schema:2004" xmlns:ns7="http://eu.prestoprime/xsd/acl" xmlns:ns10=
    "http://www.openarchives.org/OAI/2.0/oai-identifier" xmlns:ns8="http://www.openarchives.org/
    OAI/1.1/eprints" xmlns:ns11="http://www.prestoprime.eu/xsd/set" xmlns:dnx="http://www.
    prestoprime.eu/xsd/dnx" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:mets="http://www.loc
    .gov/METS/" OBJID="aba2890b-cd8b-4c44-bf15-6bd2f19a1163" LABEL="PPRIME_SIP">
    <mets:metsHdr CREATEDATE="2011-11-07T00:10:56.636+01:00"/>
    <mets:dmdSec ID="dmd-001" ADMID="">
        <mets:mdWrap MDTYPE="DC">
            <mets:xmlData>
                <dc:record>
                    <dc:title>Short Example MXF/D10 Ingestion</dc:title>
                    <dc:creator>EURIX</dc:creator>
                    <dc:description>Sample video for ingestion test</dc:description>
                    <dc:publisher>EURIX</dc:publisher>
                    <dc:format>application/mxf</dc:format>
                    <dc:format>Material eXchange Format (MXF), D10 Mapping</dc:format>
                    <dc:identifier>it:pprime:eurix:008</dc:identifier>
                    <dc:source>urn:rai:crit:2011:pprime:001</dc:source>
                    <dc:language>IT</dc:language>
                </dc:record>
            </mets:xmlData>
        </mets:mdWrap>
    </mets:dmdSec>
    <mets:amdSec ID="a001">
        <mets:rightsMD ID="ppavro-rights-001" ADMID="">
            <mets:mdRef xlink:href="http://p4.prestoprime.eu/permstore/aba2890b-cd8b-4c44-bf15-6
                bd2f19a1163/PPRIME_Example_Full.owl" LOCTYPE="URL" MIMETYPE="application/owl+xml"
                 SIZE="8857" CHECKSUM="ccab8c64046ecafe97f91db9053515d6" CHECKSUMTYPE="MD5"
                MDTYPE="OTHER" OTHERMDTYPE="PPAVRO"/>
        </mets:rightsMD>
    </mets:amdSec>
    <mets:amdSec ID="ffprobe-amd">
        <mets:techMD ID="ffprobe-amd-tech">
            <mets:mdWrap MDTYPE="OTHER" OTHERMDTYPE="DNX">
                <mets:xmlData>
                    <dnx:dnx>
                        <dnx:section id="ffprobe-techmd">
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">ffprobe:filename</dnx:key
                                    >
                                <dnx:key id="significantPropertiesValue">/storage/pprime/p4test/
                                    samples/PPRIME_Example_Short_D10.mxf</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">ffprobe:nb_streams</
                                    dnx:key>
                                <dnx:key id="significantPropertiesValue">2</dnx:key>
                            </dnx:record>
                            <dnx:record>
```

```
    <dnx:key id="significantPropertiesType">ffprobe:format_name</
        dnx:key>
    <dnx:key id="significantPropertiesValue">mxf</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">ffprobe:format_long_name<
        /dnx:key>
    <dnx:key id="significantPropertiesValue">Material eXchange Format
        </dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">ffprobe:start_time</
        dnx:key>
    <dnx:key id="significantPropertiesValue">0.000000 </dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">ffprobe:duration</dnx:key
        >
    <dnx:key id="significantPropertiesValue">16.320000 </dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">ffprobe:size</dnx:key>
    <dnx:key id="significantPropertiesValue">127642112.000000 </
        dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">ffprobe:bit_rate</dnx:key
        >
    <dnx:key id="significantPropertiesValue">62569662.000000 </
        dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">ffprobe:TAG:timecode</
        dnx:key>
    <dnx:key id="significantPropertiesValue">46:26:26:20</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">ffprobe:codec_name_0</
        dnx:key>
    <dnx:key id="significantPropertiesValue">mpeg2video</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">ffprobe:codec_long_name_0
        </dnx:key>
    <dnx:key id="significantPropertiesValue">MPEG-2 video</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">ffprobe:codec_type_0</
        dnx:key>
    <dnx:key id="significantPropertiesValue">video</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">ffprobe:codec_time_base_0
        </dnx:key>
    <dnx:key id="significantPropertiesValue">1/50</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">
        ffprobe:codec_tag_string_0</dnx:key>
    <dnx:key id="significantPropertiesValue">[0][0][0][0]</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">ffprobe:codec_tag_0</
        dnx:key>
    <dnx:key id="significantPropertiesValue">0x0000</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">ffprobe:width_0</dnx:key>
```

```
        <dnx:key id="significantPropertiesValue">720</dnx:key>
    </dnx:record>
    <dnx:record>
        <dnx:key id="significantPropertiesType">ffprobe:height_0</dnx:key
            >
        <dnx:key id="significantPropertiesValue">608</dnx:key>
    </dnx:record>
    <dnx:record>
        <dnx:key id="significantPropertiesType">ffprobe:has_b_frames_0</
            dnx:key>
        <dnx:key id="significantPropertiesValue">0</dnx:key>
    </dnx:record>
    <dnx:record>
        <dnx:key id="significantPropertiesType">
            ffprobe:sample_aspect_ratio_0</dnx:key>
        <dnx:key id="significantPropertiesValue">152:135</dnx:key>
    </dnx:record>
    <dnx:record>
        <dnx:key id="significantPropertiesType">
            ffprobe:display_aspect_ratio_0</dnx:key>
        <dnx:key id="significantPropertiesValue">4:3</dnx:key>
    </dnx:record>
    <dnx:record>
        <dnx:key id="significantPropertiesType">ffprobe:pix_fmt_0</
            dnx:key>
        <dnx:key id="significantPropertiesValue">yuv422p</dnx:key>
    </dnx:record>
    <dnx:record>
        <dnx:key id="significantPropertiesType">ffprobe:level_0</dnx:key>
        <dnx:key id="significantPropertiesValue">5</dnx:key>
    </dnx:record>
    <dnx:record>
        <dnx:key id="significantPropertiesType">ffprobe:interlaced_0</
            dnx:key>
        <dnx:key id="significantPropertiesValue">tff</dnx:key>
    </dnx:record>
    <dnx:record>
        <dnx:key id="significantPropertiesType">ffprobe:r_frame_rate_0</
            dnx:key>
        <dnx:key id="significantPropertiesValue">25/1</dnx:key>
    </dnx:record>
    <dnx:record>
        <dnx:key id="significantPropertiesType">ffprobe:avg_frame_rate_0<
            /dnx:key>
        <dnx:key id="significantPropertiesValue">25/1</dnx:key>
    </dnx:record>
    <dnx:record>
        <dnx:key id="significantPropertiesType">ffprobe:time_base_0</
            dnx:key>
        <dnx:key id="significantPropertiesValue">1/25</dnx:key>
    </dnx:record>
    <dnx:record>
        <dnx:key id="significantPropertiesType">ffprobe:start_time_0</
            dnx:key>
        <dnx:key id="significantPropertiesValue">0.000000 </dnx:key>
    </dnx:record>
    <dnx:record>
        <dnx:key id="significantPropertiesType">ffprobe:duration_0</
            dnx:key>
        <dnx:key id="significantPropertiesValue">16.320000 </dnx:key>
    </dnx:record>
    <dnx:record>
        <dnx:key id="significantPropertiesType">ffprobe:codec_name_1</
            dnx:key>
        <dnx:key id="significantPropertiesValue">pcm_s16le</dnx:key>
    </dnx:record>
    <dnx:record>
        <dnx:key id="significantPropertiesType">ffprobe:codec_long_name_1
```

```
                    </dnx:key>
                <dnx:key id="significantPropertiesValue">PCM signed 16-bit little
                    -endian</dnx:key>
            </dnx:record>
            <dnx:record>
                <dnx:key id="significantPropertiesType">ffprobe:codec_type_1</
                    dnx:key>
                <dnx:key id="significantPropertiesValue">audio</dnx:key>
            </dnx:record>
            <dnx:record>
                <dnx:key id="significantPropertiesType">ffprobe:codec_time_base_1
                    </dnx:key>
                <dnx:key id="significantPropertiesValue">0/1</dnx:key>
            </dnx:record>
            <dnx:record>
                <dnx:key id="significantPropertiesType">
                    ffprobe:codec_tag_string_1</dnx:key>
                <dnx:key id="significantPropertiesValue">[0][0][0][0]</dnx:key>
            </dnx:record>
            <dnx:record>
                <dnx:key id="significantPropertiesType">ffprobe:codec_tag_1</
                    dnx:key>
                <dnx:key id="significantPropertiesValue">0x0000</dnx:key>
            </dnx:record>
            <dnx:record>
                <dnx:key id="significantPropertiesType">ffprobe:sample_rate_1</
                    dnx:key>
                <dnx:key id="significantPropertiesValue">48000.000000 </dnx:key>
            </dnx:record>
            <dnx:record>
                <dnx:key id="significantPropertiesType">ffprobe:channels_1</
                    dnx:key>
                <dnx:key id="significantPropertiesValue">4</dnx:key>
            </dnx:record>
            <dnx:record>
                <dnx:key id="significantPropertiesType">ffprobe:bits_per_sample_1
                    </dnx:key>
                <dnx:key id="significantPropertiesValue">16</dnx:key>
            </dnx:record>
            <dnx:record>
                <dnx:key id="significantPropertiesType">ffprobe:r_frame_rate_1</
                    dnx:key>
                <dnx:key id="significantPropertiesValue">0/0</dnx:key>
            </dnx:record>
            <dnx:record>
                <dnx:key id="significantPropertiesType">ffprobe:avg_frame_rate_1<
                    /dnx:key>
                <dnx:key id="significantPropertiesValue">0/0</dnx:key>
            </dnx:record>
            <dnx:record>
                <dnx:key id="significantPropertiesType">ffprobe:time_base_1</
                    dnx:key>
                <dnx:key id="significantPropertiesValue">1/25</dnx:key>
            </dnx:record>
            <dnx:record>
                <dnx:key id="significantPropertiesType">ffprobe:start_time_1</
                    dnx:key>
                <dnx:key id="significantPropertiesValue">0.000000 </dnx:key>
            </dnx:record>
            <dnx:record>
                <dnx:key id="significantPropertiesType">ffprobe:duration_1</
                    dnx:key>
                <dnx:key id="significantPropertiesValue">16.320000 </dnx:key>
            </dnx:record>
        </dnx:section>
    </dnx:dnx>
</mets:xmlData>
</mets:mdWrap>
```

```
        </mets:techMD>
    </mets:amdSec>
    <mets:amdSec ID="mxftechmd-amd">
        <mets:techMD ID="mxftechmd-amd-tech">
            <mets:mdWrap MDTYPE="OTHER" OTHERMDTYPE="DNX">
                <mets:xmlData>
                    <dnx:dnx>
                        <dnx:section id="mxf-techmd">
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:VideoActiveLinesPerFrame</dnx:key>
                                <dnx:key id="significantPropertiesValue">576</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:AudioChannelCount</dnx:key>
                                <dnx:key id="significantPropertiesValue">4</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:SampledYOffset</dnx:key>
                                <dnx:key id="significantPropertiesValue">0</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:DisplayXOffset</dnx:key>
                                <dnx:key id="significantPropertiesValue">0</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:EditRate</dnx:key>
                                <dnx:key id="significantPropertiesValue">25:1</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:Duration</dnx:key>
                                <dnx:key id="significantPropertiesValue">408</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:AudioSamplingRate</dnx:key>
                                <dnx:key id="significantPropertiesValue">48000:1</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:DisplayWidth</dnx:key>
                                <dnx:key id="significantPropertiesValue">720</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:SampledHeight</dnx:key>
                                <dnx:key id="significantPropertiesValue">304</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:DisplayYOffset</dnx:key>
                                <dnx:key id="significantPropertiesValue">16</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:VerticalSubSampling</dnx:key>
                                <dnx:key id="significantPropertiesValue">1</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:PartitionPack</dnx:key>
                                <dnx:key id="significantPropertiesValue">Header Partition: Closed
                                     and Complete</dnx:key>
```

```
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">
        MXFTechMDExtractor:AspectRatio</dnx:key>
    <dnx:key id="significantPropertiesValue">4:3</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">
        MXFTechMDExtractor:EssenceContainers</dnx:key>
    <dnx:key id="significantPropertiesValue">D10 Mapping</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">
        MXFTechMDExtractor:AudioQuantizationBits</dnx:key>
    <dnx:key id="significantPropertiesValue">16</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">
        MXFTechMDExtractor:StoredWidth</dnx:key>
    <dnx:key id="significantPropertiesValue">720</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">
        MXFTechMDExtractor:HorizontalSubSampling</dnx:key>
    <dnx:key id="significantPropertiesValue">2</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">
        MXFTechMDExtractor:PictureEssenceCoding</dnx:key>
    <dnx:key id="significantPropertiesValue">06 0e 2b 34 04 01 01 01
        04 01 02 02 01 02 01 01</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">
        MXFTechMDExtractor:SampledWidth</dnx:key>
    <dnx:key id="significantPropertiesValue">720</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">
        MXFTechMDExtractor:SoundEssenceCoding</dnx:key>
    <dnx:key id="significantPropertiesValue"/>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">
        MXFTechMDExtractor:MaterialPackageID</dnx:key>
    <dnx:key id="significantPropertiesValue">06 0a 2b 34 01 01 01 05
        01 01 0d 13 13 00 00 00 de 5a 00 00 46 30 05 80 66 d2 00 1a
        80 e9 6a 08</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">
        MXFTechMDExtractor:FrameLayoutName</dnx:key>
    <dnx:key id="significantPropertiesValue">SEPARATE_FIELDS</dnx:key
        >
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">
        MXFTechMDExtractor:ComponentDepth</dnx:key>
    <dnx:key id="significantPropertiesValue">8</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">
        MXFTechMDExtractor:SampledXOffset</dnx:key>
    <dnx:key id="significantPropertiesValue">0</dnx:key>
</dnx:record>
<dnx:record>
    <dnx:key id="significantPropertiesType">
        MXFTechMDExtractor:ActiveFormatDescriptor</dnx:key>
```

```
                                <dnx:key id="significantPropertiesValue"/>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:FrameLayout</dnx:key>
                                <dnx:key id="significantPropertiesValue">1</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:DisplayHeight</dnx:key>
                                <dnx:key id="significantPropertiesValue">288</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:StoredHeight</dnx:key>
                                <dnx:key id="significantPropertiesValue">304</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="significantPropertiesType">
                                    MXFTechMDExtractor:OperationalPattern</dnx:key>
                                <dnx:key id="significantPropertiesValue">OP1a</dnx:key>
                            </dnx:record>
                        </dnx:section>
                    </dnx:dnx>
                </mets:xmlData>
            </mets:mdWrap>
        </mets:techMD>
    </mets:amdSec>
    <mets:amdSec ID="file-id-001-amd">
        <mets:techMD ID="file-id-001-amd-tech">
            <mets:mdWrap MDTYPE="OTHER" OTHERMDTYPE="DNX">
                <mets:xmlData>
                    <dnx:dnx>
                        <dnx:section id="fileFixity">
                            <dnx:record>
                                <dnx:key id="agent">JAVA7_REG_FIXITY</dnx:key>
                                <dnx:key id="fixityType">MD5</dnx:key>
                                <dnx:key id="fixityValue">83a07b8ff47d129ede0b323b272bc073</
                                    dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="agent">JAVA7_REG_FIXITY</dnx:key>
                                <dnx:key id="fixityType">SHA-1</dnx:key>
                                <dnx:key id="fixityValue">
                                    b0e97f94170529d5954aa5f735c757100abc08c5</dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="agent">D10SumChecker</dnx:key>
                                <dnx:key id="fixityType">MD5</dnx:key>
                                <dnx:key id="fixityValue">83a07b8ff47d129ede0b323b272bc073</
                                    dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="agent">D10SumChecker</dnx:key>
                                <dnx:key id="fixityType">D10SumChecker EditUnit MD5</dnx:key>
                                <dnx:key id="fixityValue">
EDIT_UNIT_NUMBER: 000001          TIMECODE: 10:12:35:01     EDIT_UNIT_MD5:
    ec6f26d1bdc36b2bfaa14e1b54628cbf PICTURE_ITEM_MD5: 6dcd5e31a48e0e576eb5d32249a43811
    AUDIO_ITEM_MD5: d04459ab556b497133c2cd4aff47825b
                ...
EDIT_UNIT_NUMBER: 000408          TIMECODE: 10:12:51:08     EDIT_UNIT_MD5: 755
    b57d19bcad48ef41d42e5aadf0aa9 PICTURE_ITEM_MD5: 2f8b68c648d6a77803656c1f08acb31b
    AUDIO_ITEM_MD5: df63ed36f82b4fa7579ced031161247a
</dnx:key>
                            </dnx:record>
                        </dnx:section>
                    </dnx:dnx>
                </mets:xmlData>
```

```
            </mets:mdWrap>
        </mets:techMD>
    </mets:amdSec>
    <mets:amdSec ID="file-id-002-amd">
        <mets:techMD ID="file-id-002-amd-tech">
            <mets:mdWrap MDTYPE="OTHER" OTHERMDTYPE="DNX">
                <mets:xmlData>
                    <dnx:dnx>
                        <dnx:section id="fileFixity">
                            <dnx:record>
                                <dnx:key id="agent">JAVA7_REG_FIXITY</dnx:key>
                                <dnx:key id="fixityType">MD5</dnx:key>
                                <dnx:key id="fixityValue">edba2a29c0461583293135e53d5345c3</
                                    dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="agent">JAVA7_REG_FIXITY</dnx:key>
                                <dnx:key id="fixityType">SHA-1</dnx:key>
                                <dnx:key id="fixityValue">
                                    ef0be2ab432ef3a94a5d46e2300d5bcd9755d072</dnx:key>
                            </dnx:record>
                        </dnx:section>
                    </dnx:dnx>
                </mets:xmlData>
            </mets:mdWrap>
        </mets:techMD>
    </mets:amdSec>
    <mets:amdSec ID="file-id-003-amd">
        <mets:techMD ID="file-id-003-amd-tech">
            <mets:mdWrap MDTYPE="OTHER" OTHERMDTYPE="DNX">
                <mets:xmlData>
                    <dnx:dnx>
                        <dnx:section id="fileFixity">
                            <dnx:record>
                                <dnx:key id="agent">JAVA7_REG_FIXITY</dnx:key>
                                <dnx:key id="fixityType">MD5</dnx:key>
                                <dnx:key id="fixityValue">5a7f2b26fb0274a282059bfb95e717b9</
                                    dnx:key>
                            </dnx:record>
                            <dnx:record>
                                <dnx:key id="agent">JAVA7_REG_FIXITY</dnx:key>
                                <dnx:key id="fixityType">SHA-1</dnx:key>
                                <dnx:key id="fixityValue">1304
                                    b3cb19e2bb8ff11b2ee53982124acc26de01</dnx:key>
                            </dnx:record>
                        </dnx:section>
                    </dnx:dnx>
                </mets:xmlData>
            </mets:mdWrap>
        </mets:techMD>
    </mets:amdSec>
    <mets:amdSec>
        <mets:techMD>
            <mets:mdRef ID="user-annotations" xlink:href="http://p4.prestoprime.eu/permstore/
                aba2890b-cd8b-4c44-bf15-6bd2f19a1163/user-annotations-aba2890b-cd8b-4c44-bf15-6
                bd2f19a1163.xml" LOCTYPE="URL" MIMETYPE="application/xml" SIZE="280647" CREATED="
                2011-11-07T01:22:44.378+01:00" CHECKSUM="886856de0541db8112f79d4c73ba1e93"
                CHECKSUMTYPE="MD5" MDTYPE="OTHER" OTHERMDTYPE="USER_ANNOTATIONS"/>
        </mets:techMD>
    </mets:amdSec>
    <mets:amdSec>
        <mets:techMD>
            <mets:mdRef ID="quality-assessment" xlink:href="http://p4.prestoprime.eu/permstore/
                aba2890b-cd8b-4c44-bf15-6bd2f19a1163/quality-assessment-aba2890b-cd8b-4c44-bf15-6
                bd2f19a1163.xml" LOCTYPE="URL" MIMETYPE="application/xml" SIZE="204343" CREATED="
                2011-11-07T01:23:04.785+01:00" CHECKSUM="703a4479307820c3709bb84a247077a4"
                CHECKSUMTYPE="MD5" MDTYPE="OTHER" OTHERMDTYPE="QUALITY_ASSESSMENT"/>
        </mets:techMD>
```

```
    </mets:amdSec>
    <mets:fileSec>
        <mets:fileGrp ID="rep-001" ADMID="dmd-001">
            <mets:file ID="file-id-001" ADMID="" DMDID="" MIMETYPE="application/mxf" CHECKSUM="83
                a07b8ff47d129ede0b323b272bc073" CHECKSUMTYPE="MD5">
                <mets:FLocat ID="FLocat-001" xlink:href="/storage/pprime/p4test/permstore/
                    aba2890b-cd8b-4c44-bf15-6bd2f19a1163/PPRIME_Example_Short_D10.mxf"
                    xlink:title="Example of MXF/D10 (Master Quality) - File" LOCTYPE="OTHER"
                    OTHERLOCTYPE="FILE"/>
                <mets:FLocat ID="fLocat-MQ-URL" xlink:href="http://p4.prestoprime.eu/permstore/
                    aba2890b-cd8b-4c44-bf15-6bd2f19a1163/PPRIME_Example_Short_D10.mxf" LOCTYPE="
                    URL"/>
                <mets:FLocat ID="fLocat-MQ-LTO" xlink:href="lto-ltfs:pprime:LT001:id1320621047348
                    " LOCTYPE="URN"/>
                <mets:FLocat ID="fLocat-MQ-DRACMA" xlink:href="
                    it:rai:dracma:0x060A2B340101010501010D1313000000DE5A00004630058066D2001A80E96A08
                    " LOCTYPE="URN"/>
            </mets:file>
            <mets:file ID="file-id-002" MIMETYPE="video/webm" CHECKSUM="
                edba2a29c0461583293135e53d5345c3" CHECKSUMTYPE="MD5">
                <mets:FLocat ID="fLocat-LQ-FILE-002" xlink:href="/storage/pprime/p4test/permstore
                    /aba2890b-cd8b-4c44-bf15-6bd2f19a1163/PPRIME_Example_Short_D10.webm" LOCTYPE=
                    "OTHER" OTHERLOCTYPE="FILE"/>
                <mets:FLocat ID="fLocat-LQ-URL-002" xlink:href="http://p4.prestoprime.eu/
                    permstore/aba2890b-cd8b-4c44-bf15-6bd2f19a1163/PPRIME_Example_Short_D10.webm"
                     LOCTYPE="URL"/>
            </mets:file>
            <mets:file ID="file-id-003" MIMETYPE="video/ogg" CHECKSUM="5
                a7f2b26fb0274a282059bfb95e717b9" CHECKSUMTYPE="MD5">
                <mets:FLocat ID="fLocat-LQ-FILE-003" xlink:href="/storage/pprime/p4test/permstore
                    /aba2890b-cd8b-4c44-bf15-6bd2f19a1163/PPRIME_Example_Short_D10.ogv" LOCTYPE="
                    OTHER" OTHERLOCTYPE="FILE"/>
                <mets:FLocat ID="fLocat-LQ-URL-003" xlink:href="http://p4.prestoprime.eu/
                    permstore/aba2890b-cd8b-4c44-bf15-6bd2f19a1163/PPRIME_Example_Short_D10.ogv"
                    LOCTYPE="URL"/>
            </mets:file>
        </mets:fileGrp>
    </mets:fileSec>
</mets:mets>
```

# D Application.wadl

In the following we report the `application.wadl` interface descriptor for all the REST interfaces published by Jersey. The descriptor is represented using WADL (Web Application Description Language)[29]. WADL reminds of WSDL for SOAP web servers, but currently there is only a formal submission request to W3C for discussion, so WADL is not a W3C standard. Anyway WADL is getting more and more popular in all implementations of Java REST Web Services. The WADL is automatically built with all the information available from the compiled code and includes the URL, resources, methods with request and response format and so on. The generated WADL for the current P4 web server is listed in the following.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<application xmlns="http://research.sun.com/wadl/2006/10">
    <doc xmlns:jersey="http://jersey.dev.java.net/" jersey:generatedBy="Jersey: 1.5 01/14/2011
        12:36 PM"/>
    <resources base="http://localhost:8080/p4ws/">
        <resource path="/test">
            <resource path="/createAIP">
                <method id="createAIP" name="POST">
                    <request>
                        <representation mediaType="multipart/form-data"/>
                    </request>
                    <response>
                        <representation mediaType="text/plain"/>
                    </response>
                </method>
            </resource>
            <resource path="/createAIP/form">
                <method id="createAIPWithForm" name="POST">
                    <response>
                        <representation mediaType="text/plain"/>
                    </response>
                </method>
            </resource>
            <resource path="/findACL">
                <method id="findAclListByAIPId" name="GET">
                    <request>
                        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="aipId"
                            style="query" type="xs:string"/>
                    </request>
                    <response>
                        <representation mediaType="application/xml"/>
                    </response>
                </method>
            </resource>
            <resource path="/findDNX">
                <method id="findDnxListByAIPId" name="GET">
                    <request>
                        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="aipId"
                            style="query" type="xs:string"/>
                    </request>
                    <response>
                        <representation mediaType="application/xml"/>
                    </response>
                </method>
            </resource>
            <resource path="/findAllRights">
                <method id="findAllRights" name="GET">
                    <response>
```

[29] http://www.w3.org/Submission/wadl/

```
                <representation mediaType="application/xml"/>
            </response>
        </method>
    </resource>
    <resource path="/dracma/cgi-bin/index_request.sh">
        <method id="indexToDracma" name="GET">
            <request>
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="file"
                    style="query" type="xs:string"/>
            </request>
            <response>
                <representation mediaType="text/plain"/>
            </response>
        </method>
    </resource>
    <resource path="/dracma/cgi-bin/get_status.sh">
        <method id="checkDracmaStatus" name="GET">
            <request>
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id"
                    style="query" type="xs:string"/>
            </request>
            <response>
                <representation mediaType="text/plain"/>
            </response>
        </method>
    </resource>
    <resource path="/pprime/cgi-bin/WriteToLTO">
        <method id="writeToLTO" name="GET">
            <request>
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="command"
                    style="query" type="xs:string"/>
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="FileName"
                    style="query" type="xs:string"/>
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="TaskID"
                    style="query" type="xs:string"/>
            </request>
            <response>
                <representation mediaType="text/plain"/>
            </response>
        </method>
    </resource>
    <resource path="/deleteAIP">
        <method id="deleteAIP" name="GET">
            <request>
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="aipIdToDelete"
                    style="query" type="xs:string"/>
            </request>
            <response>
                <representation mediaType="text/plain"/>
            </response>
        </method>
    </resource>
</resource>
<resource path="/admin">
    <resource path="/fixitycheck/{id}">
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
            type="xs:string"/>
        <method id="fixityChecks" name="GET">
            <response>
                <representation mediaType="text/plain"/>
            </response>
        </method>
    </resource>
    <resource path="/restore">
        <method id="restoreFile" name="POST">
            <response>
                <representation mediaType="text/plain"/>
            </response>
```

```
            </method>
        </resource>
        <resource path="/dip/checklist">
            <method id="getDIPToBeChecked" name="GET">
                <response>
                    <representation mediaType="text/plain"/>
                </response>
            </method>
        </resource>
        <resource path="/jobs/{status}">
            <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="status"
                style="template" type="xs:string"/>
            <method id="getAllJobs" name="GET">
                <response>
                    <representation mediaType="text/plain"/>
                </response>
            </method>
        </resource>
    </resource>
    <resource path="/search">
        <resource path="/quick">
            <method id="quickSearch" name="GET">
                <request>
                    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="term"
                        style="query" type="xs:string"/>
                    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="from"
                        style="query" type="xs:string"/>
                    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="resultCount"
                        style="query" type="xs:string"/>
                    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="sortAsc"
                        style="query" type="xs:string"/>
                    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="sortField"
                        style="query" type="xs:string"/>
                    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="dateFacet"
                        style="query" type="xs:string"/>
                    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="creatorFacet"
                        style="query" type="xs:string"/>
                    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="publisherFacet"
                        style="query" type="xs:string"/>
                </request>
                <response>
                    <representation mediaType="application/xml"/>
                </response>
            </method>
        </resource>
        <resource path="/detail">
            <method id="detailSearch" name="GET">
                <request>
                    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="title"
                        style="query" type="xs:string"/>
                    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="description"
                        style="query" type="xs:string"/>
                    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="format"
                        style="query" type="xs:string"/>
                    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="identifier"
                        style="query" type="xs:string"/>
                </request>
                <response>
                    <representation mediaType="application/xml"/>
                </response>
            </method>
        </resource>
        <resource path="/checkCredentials">
            <method id="checkCredentials" name="GET">
                <response>
                    <representation mediaType="text/plain"/>
                </response>
            </method>
```

```
        </resource>
    </resource>
    <resource path="/ingest/update">
        <resource path="/overwrite">
            <method id="updateAIP" name="POST">
                <response>
                    <representation mediaType="text/plain"/>
                </response>
            </method>
        </resource>
        <resource path="/rights">
            <method id="updateAIPRights" name="POST">
                <response>
                    <representation mediaType="text/plain"/>
                </response>
            </method>
        </resource>
        <resource path="/qa">
            <method id="updateAIPQA" name="POST">
                <response>
                    <representation mediaType="text/plain"/>
                </response>
            </method>
        </resource>
        <resource path="/usermd">
            <method id="updateAIPUserAnnotations" name="POST">
                <response>
                    <representation mediaType="text/plain"/>
                </response>
            </method>
        </resource>
    </resource>
    <resource path="/access">
        <method id="findDIPByGET" name="GET">
            <request>
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="query"
                    type="xs:string"/>
            </request>
            <response>
                <representation mediaType="text/plain"/>
            </response>
        </method>
        <resource path="/dip/thumb/{id}">
            <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
                type="xs:string"/>
            <method id="getDIPThumb" name="GET">
                <response>
                    <representation mediaType="text/plain"/>
                </response>
            </method>
        </resource>
        <resource path="/dip/multivalent/{id}">
            <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
                type="xs:string"/>
            <method id="getMultivalentJNLP" name="GET">
                <response>
                    <representation mediaType="text/plain"/>
                </response>
            </method>
        </resource>
        <resource path="/dip/video/{id}">
            <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
                type="xs:string"/>
            <method id="getQuickPlayer" name="GET">
                <response>
                    <representation mediaType="text/html"/>
                </response>
            </method>
```

```
</resource>
<resource path="/dip/frames/{id}">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
        type="xs:string"/>
    <method id="getFrames" name="GET">
        <response>
            <representation mediaType="text/plain"/>
        </response>
    </method>
</resource>
<resource path="/dip/rights/{id}">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
        type="xs:string"/>
    <method id="getOWL" name="GET">
        <response>
            <representation mediaType="text/xml"/>
        </response>
    </method>
</resource>
<resource path="/dip/graph/{id}">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
        type="xs:string"/>
    <method id="getOWLGraph" name="GET">
        <response>
            <representation mediaType="text/plain"/>
        </response>
    </method>
</resource>
<resource path="/searchDCRecords">
    <method id="searchDCRecords" name="GET">
        <request>
            <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="title"
                style="query" type="xs:string"/>
            <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="description"
                style="query" type="xs:string"/>
            <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="format"
                style="query" type="xs:string"/>
            <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="identifier"
                style="query" type="xs:string"/>
        </request>
        <response>
            <representation mediaType="text/plain"/>
        </response>
    </method>
</resource>
<resource path="/dip/dc/{id}">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
        type="xs:string"/>
    <method id="findDcRecordsByAIPId" name="GET">
        <response>
            <representation mediaType="application/xml"/>
        </response>
    </method>
</resource>
<resource path="/dip/qa/{id}">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
        type="xs:string"/>
    <method id="findQAByAIPId" name="GET">
        <response>
            <representation mediaType="application/xml"/>
        </response>
    </method>
</resource>
<resource path="/dip/usermd/{id}">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
        type="xs:string"/>
    <method id="findUserMDByAIPId" name="GET">
        <response>
```

```
            <representation mediaType="application/xml"/>
        </response>
    </method>
</resource>
<resource path="/dip/rights/query-by-sample">
    <method id="queryRightsByOWL" name="POST">
        <response>
            <representation mediaType="application/xml"/>
        </response>
    </method>
</resource>
<resource path="/dip/{id}">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
        type="xs:string"/>
    <method id="getDIP" name="GET">
        <response>
            <representation mediaType="text/xml"/>
        </response>
    </method>
</resource>
<resource path="/checkCredentials">
    <method id="checkCredentials" name="GET">
        <response>
            <representation mediaType="text/plain"/>
        </response>
    </method>
</resource>
<resource path="/dip/list">
    <method id="getAllDIP" name="GET">
        <response>
            <representation mediaType="text/plain"/>
        </response>
    </method>
</resource>
<resource path="/dip/preview/{id}">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
        type="xs:string"/>
    <method id="getDIPPreview" name="GET">
        <response>
            <representation mediaType="text/plain"/>
        </response>
    </method>
</resource>
<resource path="/dip/info/container/{id}">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
        type="xs:string"/>
    <method id="checkContainer" name="GET">
        <response>
            <representation mediaType="text/plain"/>
        </response>
    </method>
</resource>
<resource path="/dip/info/dracma/{id}">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
        type="xs:string"/>
    <method id="checkMaterialID" name="GET">
        <response>
            <representation mediaType="text/plain"/>
        </response>
    </method>
</resource>
<resource path="/dip/info/history/{id}">
    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
        type="xs:string"/>
    <method id="getDIPHistory" name="GET">
        <response>
            <representation mediaType="text/plain"/>
        </response>
```

```
            </method>
        </resource>
        <resource path="/dip/{id}/rev/{rev}">
            <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
                type="xs:string"/>
            <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="rev" style="template"
                type="xs:string"/>
            <method id="getDIPRevision" name="GET">
                <response>
                    <representation mediaType="text/plain"/>
                </response>
            </method>
        </resource>
    </resource>
    <resource path="/oai">
        <method id="oaiRequest" name="GET">
            <request>
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="verb" style="query"
                    type="xs:string"/>
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="identifier"
                    style="query" type="xs:string"/>
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="metadataPrefix"
                    style="query" type="xs:string"/>
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="from" style="query"
                    type="xs:string"/>
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="until"
                    style="query" type="xs:string"/>
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="set" style="query"
                    type="xs:string"/>
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="resumptionToken"
                    style="query" type="xs:string"/>
            </request>
            <response>
                <representation mediaType="application/xml"/>
            </response>
        </method>
        <resource path="/createIdentify">
            <method id="addOAIPMHWithForm" name="POST">
                <response>
                    <representation mediaType="text/plain"/>
                </response>
            </method>
        </resource>
        <resource path="/createSet">
            <method id="addOAIPMHSets" name="POST">
                <response>
                    <representation mediaType="text/plain"/>
                </response>
            </method>
        </resource>
    </resource>
    <resource path="/ingest">
        <method id="ingestSIP" name="POST">
            <response>
                <representation mediaType="text/plain"/>
            </response>
        </method>
        <resource path="/GetStatus">
            <method id="getStatus" name="GET">
                <request>
                    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="jobId"
                        style="query" type="xs:string"/>
                </request>
                <response>
                    <representation mediaType="text/plain"/>
                </response>
            </method>
        </resource>
```

```
            <resource path="/GetResults">
                <method id="getResults" name="GET">
                    <request>
                        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="jobId"
                            style="query" type="xs:string"/>
                    </request>
                    <response>
                        <representation mediaType="text/plain"/>
                    </response>
                </method>
            </resource>
            <resource path="/CheckIdentifier/{id}">
                <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="id" style="template"
                    type="xs:string"/>
                <method id="checkIdentifier" name="GET">
                    <response>
                        <representation mediaType="text/plain"/>
                    </response>
                </method>
            </resource>
        </resource>
    </resources>
</application>
```

# List of Figures



# List of Tables