ECP-2008-DILI-518001

BHL-Europe

Implement plans for all components in WP3, including data models, technology standards etc.

Deliverable number	*D3.4*
Dissemination level	*Restricted*
Delivery date	*28 February 2010*
Status	*Final*
Author(s)	*AIT, ATOS, MfN, NHM*

e*Content*plus

This project is funded under the *e*Content*plus* programme[1],
a multiannual Community programme to make digital content in Europe more accessible, usable and exploitable.

[1] OJ L 79, 24.3.2005, p. 1.

Table of contents

1 Document History

This chapter describes the document’s creation events and contributors.

1.1 Contributors

Person	Partner
Roger Essoh	ATOS
Alexander Herzog	AIT
Walter Koch	AIT
Lee Namba	ATOS
Henning Scholz	MfN
Adrian Smales	NHM
Sami Smati	ATOS

1.2 Revision History

Revision Date	Author	Version	Change Reference & Summary
Jan 15, 2010	ATOS	0.1	Consolidated document
Feb 24, 2010	BGBM	0.2	Added External Interoperability
Feb 25, 2010	AIT	0.3	Revised version after TMB, external and internal review
Feb 28, 2010	MfN	1.0	Final version

1.3 Reviewers and Approvals

This document requires the following reviews and approvals.

Name	Position	Date	Version
Rod Page	Professor, University of Glasgow	Jan 26, 2010	0.1
Lee Namba	Deliverable responsible	Feb 24, 2010	1.0
Adrian Smales	Work Package Leader	Feb 24, 2010	1.0
Henning Scholz	Project coordinator	Feb 28, 2010	1.0

1.4 Distribution

This document has been distributed to:

Group	Date of issue	Version
BHL-Europe consortium	Jan 18, 2010	0.1
Rod Page	Jan 18, 2010	0.1
BHL-Europe consortium	Mar 01, 2010	1.0

2 Purpose and document structure

The primary aim of this document is to describe the implementation plans for all components in WP3, including data models, technology standards, etc. This document complements and in certain cases extends the descriptions of deliverable D3.2 (Document agreed standards, best practice and system components). As a result there is some overlap with that document and certain sections such as Processes and Standards are repeated although with revisions.

As these are not Technical Specifications the goal is provide an high level overview of how the system and its components will be implemented including architectural and deployment diagrams, chosen technologies, included sub-modules, and low level interface protocols. More detailed implementation such as sequence diagrams are provided where the design has stabilized.

The structure of the document follows D3.2 with the exception of Technical Considerations which has been renamed Common Services to be more in line with the OAIS specification. Chapter 3 describes the common services of the system and chapter 4 goes into more detailed implementation plans for each component.

3 Common Services

Modern, distributed computing applications assume a number of supporting services such as inter-process communication, name services, temporary storage allocation, exception handling, security, backup and directory services. Next to the functional implementation of the components these common services are needed.

3.1 Operating System and Virtualization

Operating system services provide the core services needed to operate and administer the application platform, and provide an interface between application software and the platform.

BHL-Europe will use virtualization technology from VMWare whereby larger physical servers are used in place of many small physical servers to increase the utilization of costly hardware resources such as CPU. Each OS running on a physical server becomes converted to a distinct OS running inside a virtual machine. The large server can "host" many such "guest" virtual machines.

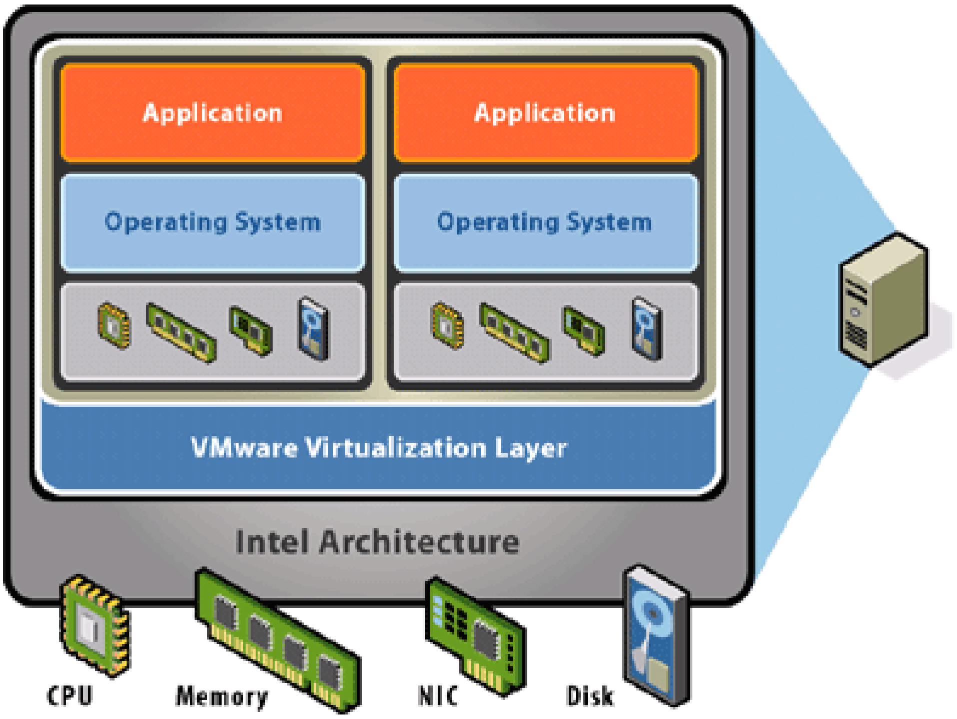


Figure 3-1: Virtualization of Servers

A virtual machine can be more easily controlled and inspected from outside than a physical one, and its configuration is more flexible. New virtual machines can be provisioned as needed without the need for any up-front hardware purchases. Also, a virtual machine can easily be relocated from one physical machine to another as needed aiding in disaster recovery scenarios.

Whenever possible BHL-Europe will use Linux based operating systems. The preferred distribution is Debian.

3.2 Network Services

Network services provide the capabilities and mechanisms to support distributed applications requiring data access and applications interoperability in heterogeneous, networked environments.

3.2.1 Service Interoperability

BHL-Europe is designed as a distributed system, and that raises the question of component interoperability. To minimize tight coupling of components, for scalability purposes, and to be able to replace underlying component technologies in the future we have decided to define our interfaces in the most technology neutral way available today. As such, data will always travel around as XML, and we will define XML containers containing payload data (like METS data, and other important information).

The OAIS Data Flow shown above revolves around a set of web services that act as data sinks and sources. To facilitate the communication between these the BHL-Europe Archive System implements service endpoints in a way that complies to the architectural style called **Representational state transfer** (**REST**).

The REST architectural style describes six constraints applied to the architecture:

1. Client-server: Clients are separated from servers by a uniform interface. This separation of concerns means that, for example, clients are not concerned with data storage, which remains internal to each server, so that the portability of client code is improved. Servers are not concerned with the user interface or user state, so that servers can be simpler and more scalable. Servers and clients may also be replaced and developed independently, as long as the interface is not altered.
2. Stateless: The client-server communication is further constrained by no client context being stored on the server between requests. Each request from any client contains all of the information necessary to service the request, and any state is held in the client. This not only makes servers more visible for monitoring, but also makes them more reliable in the face of partial or network failures as well as further enhancing their scalability.
3. Cacheable: As on the World Wide Web, clients are able to cache responses. Responses must therefore, implicitly or explicitly, define themselves as cacheable or not to prevent clients reusing stale or inappropriate data in response to further

requests. Well-managed caching partially or completely eliminates some client-server interactions, further improving scalability and performance.

4. Uniform interface: The uniform interface between clients and servers, discussed below, simplifies and decouples the architecture, which enables each part to evolve independently. The four guiding principles of this interface are detailed below.
5. Layered system: A client cannot ordinarily tell whether it is connected directly to the end server, or to an intermediary along the way. Intermediary servers may improve system scalability by enabling load balancing and by providing shared caches. They may also enforce security policies.

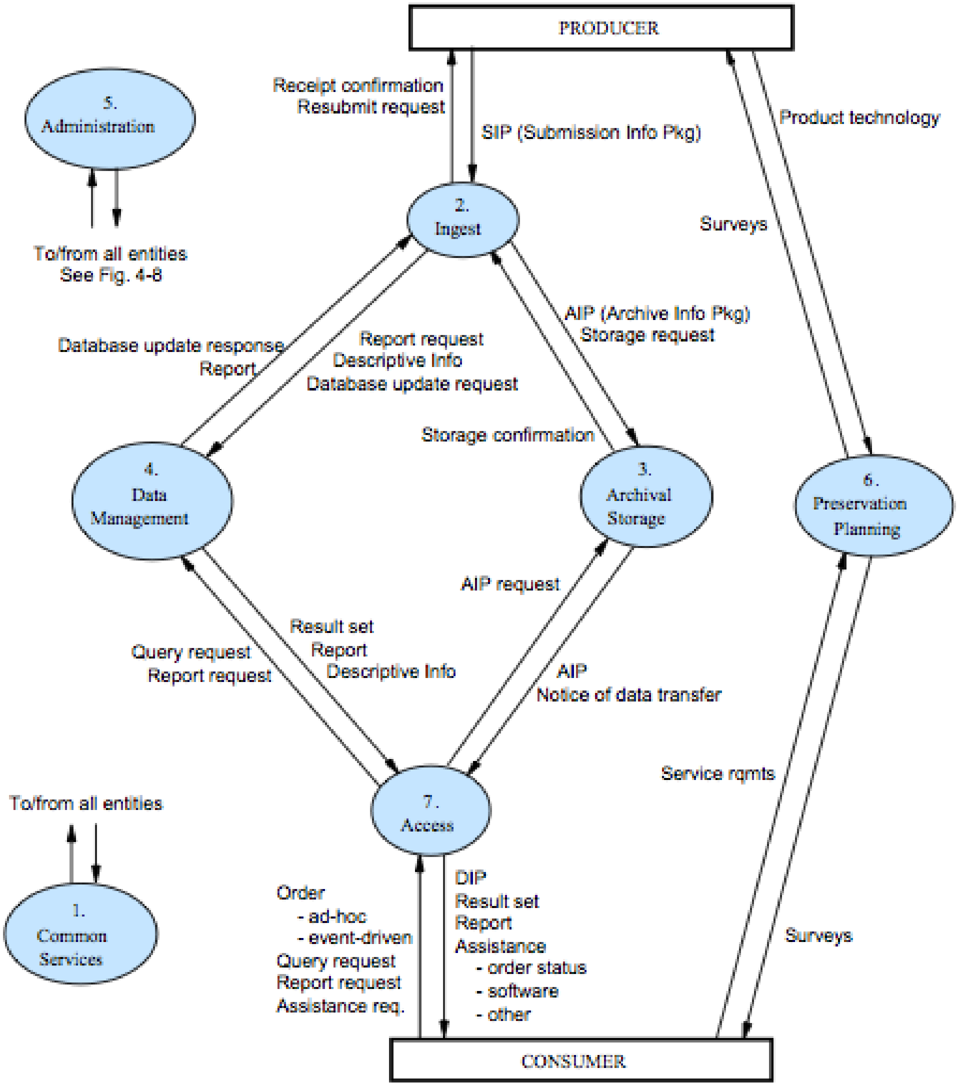


Figure 3-2: OAIS Data Flow Diagram

The web services implemented by the BHL-Europe Archive System fully comply to RESTful web services:

- There is base URI for the web service, such as http://bhl.org:8088/archive/.

- There is a MIME type of the data supported by the web service, we decided for using XML.
- The set of operations is using the HTTP methods POST, GET, PUT or DELETE.

The XML containers moved around by these services contain the payload and possibly routing information in case a rule or content based routing is used to orchestrate some sort of sequence of service calls.

3.2.2 File Storage

There are three types of file storage requirements for the system: working, nearterm, and longterm. The following diagram shows how the different OAIS components interface with the different types of storage.

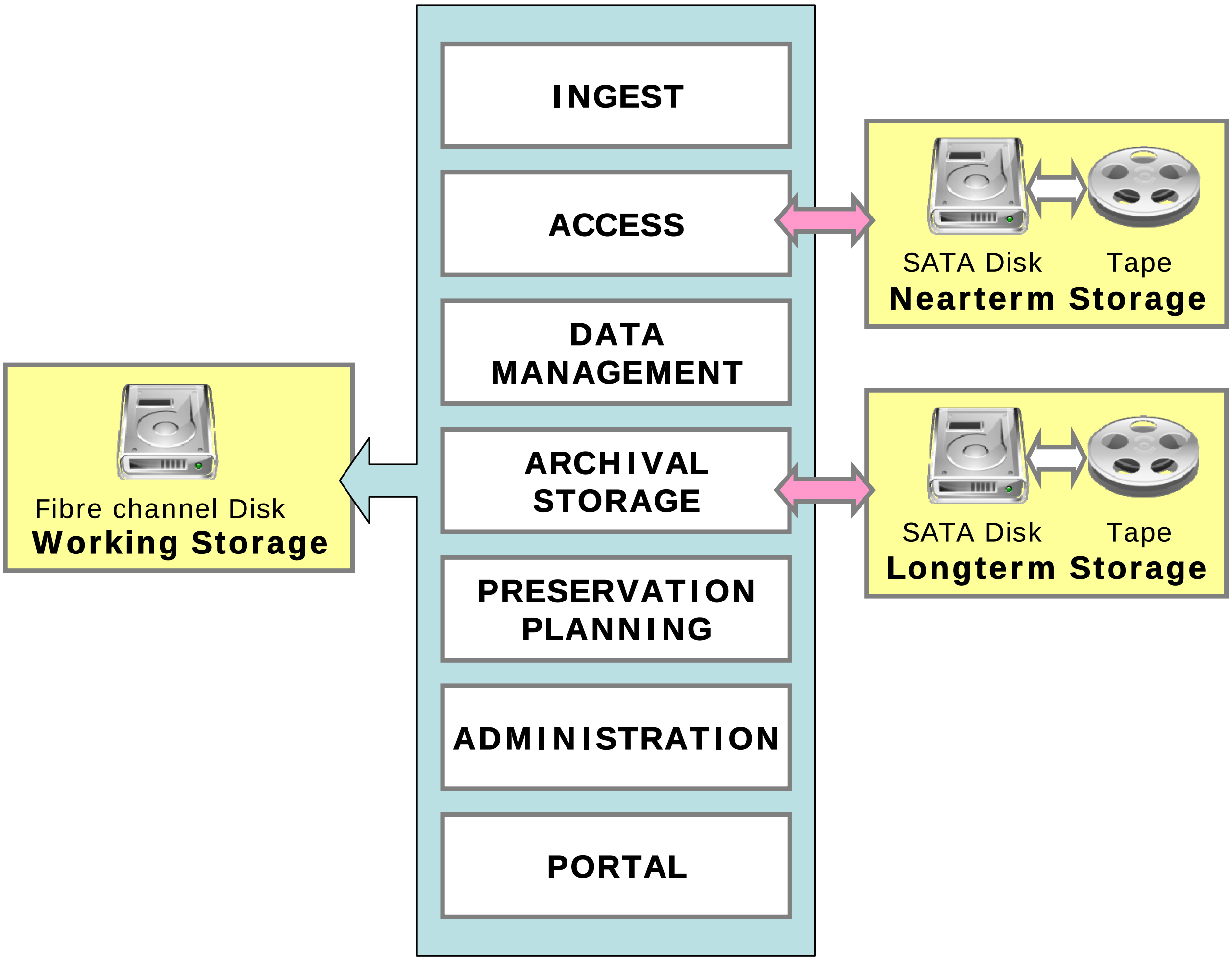


Figure 3-3: File Storage Types

3.2.2.1 Working Storage

Working storage is very high speed, dynamic in nature, very frequent acces, and temporary. The 5TB of disk only fibre channel storage will be used for:

- the operating systems of the VMWare virtual machines
- working storage for temporary file manipulation by the OAIS components.

Only the operating systems require back up.

3.2.2.2 Nearterm Storage

Nearterm storage is high speed, static in nature, frequent access, and non-permanent. It is composed of SATA disk storage with possibility the of automatic transfer to tape.

The files stored here are all of the disseminations created by the Access component such as thumbnails, PDFs, custom dissemination requests, etc.

The files are intended to be accessed frequently. They are not required to be stored for long periods of time. The response time should be under 0.5 seconds till start of transfer. Data health monitoring is optional.

3.2.2.3 Longterm Storage

Longterm storage is static in nature, infrequent access, and permanent. It is composed of SATA disk storage with automatic or manual transfer to tape.

The files stored here are the original ingested sources and associated metadata. They are stored by the Archival Storage component.

Files stored in longterm storage are intended to be stored indefinitely. For BHL-Europe we will take this to mean at least 10 years. Data health must be monitored regularly in order to proactively manage data loss. The storage should be redundant.

3.2.3 Physical Infrastructure

The physical storage system is based on IBM hardware and software solutions. The IBM General Parallel File System (GPFS) supports full Posix filesystem semantics and supports Linux operating systems. The system is composed of:

- 5 TB of fibre channel disk storage will be available for working storage.
- 100 TB of SATA disk storage will be available for nearterm and longterm storage.
- Tape storage is virtually unlimited.

The following diagram describes the storage system.

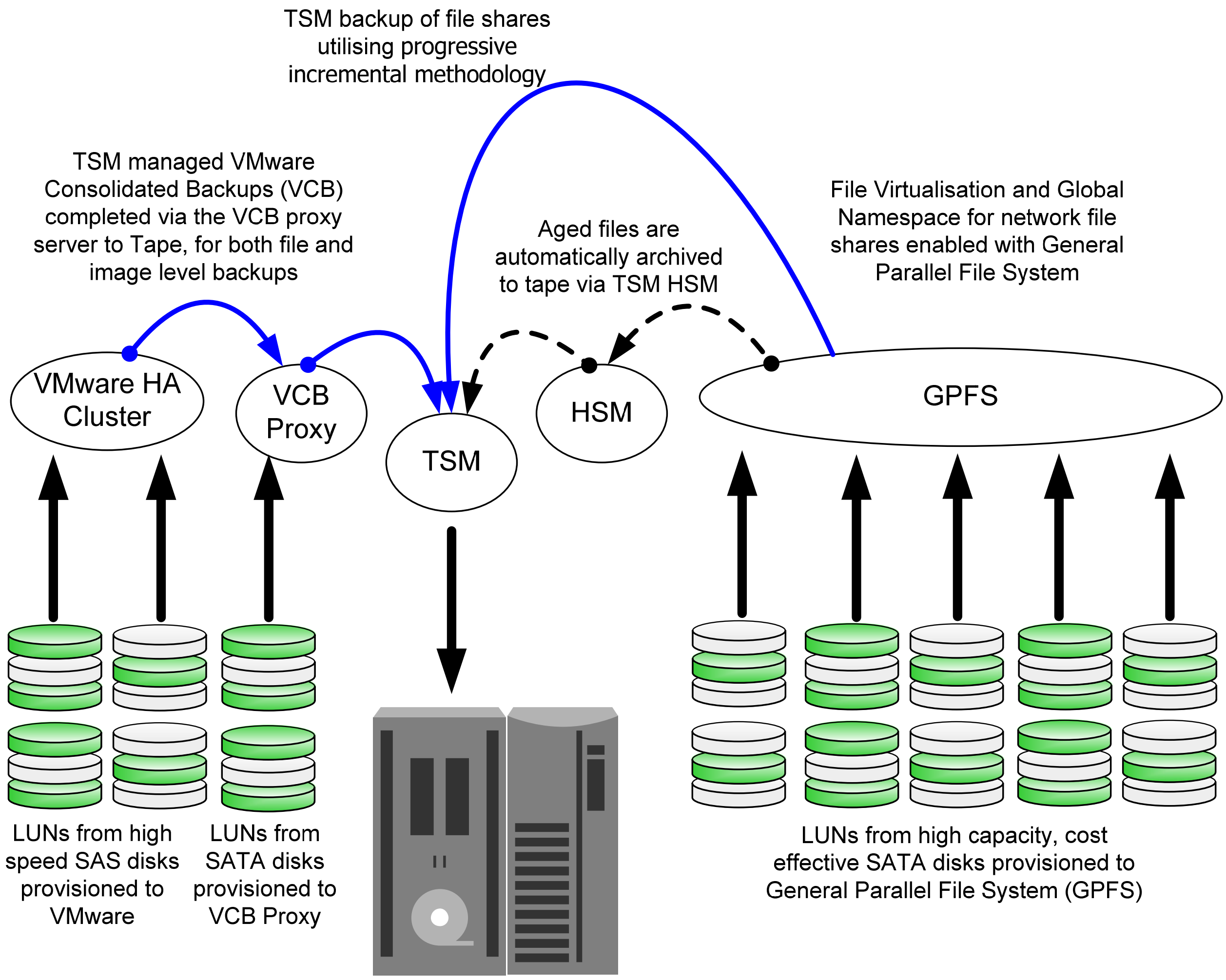

TSM backup of file shares utilising progressive incremental methodology
TSM managed VMware Consolidated Backups (VCB) completed via the VCB proxy server to Tape, for both file and image level backups
Aged files are automatically archived to tape via TSM HSM
File Virtualisation and Global Namespace for network file shares enabled with General Parallel File System
VMware HA Cluster
VCB Proxy
TSM
HSM
GPFS
LUNs from high speed SAS disks provisioned to VMware
LUNs from SATA disks provisioned to VCB Proxy
LUNs from high capacity, cost effective SATA disks provisioned to General Parallel File System (GPFS)

Figure 3-4: Storage System with Automated Tape Archival

The following diagram describes the servers, SAN, and network configuration.

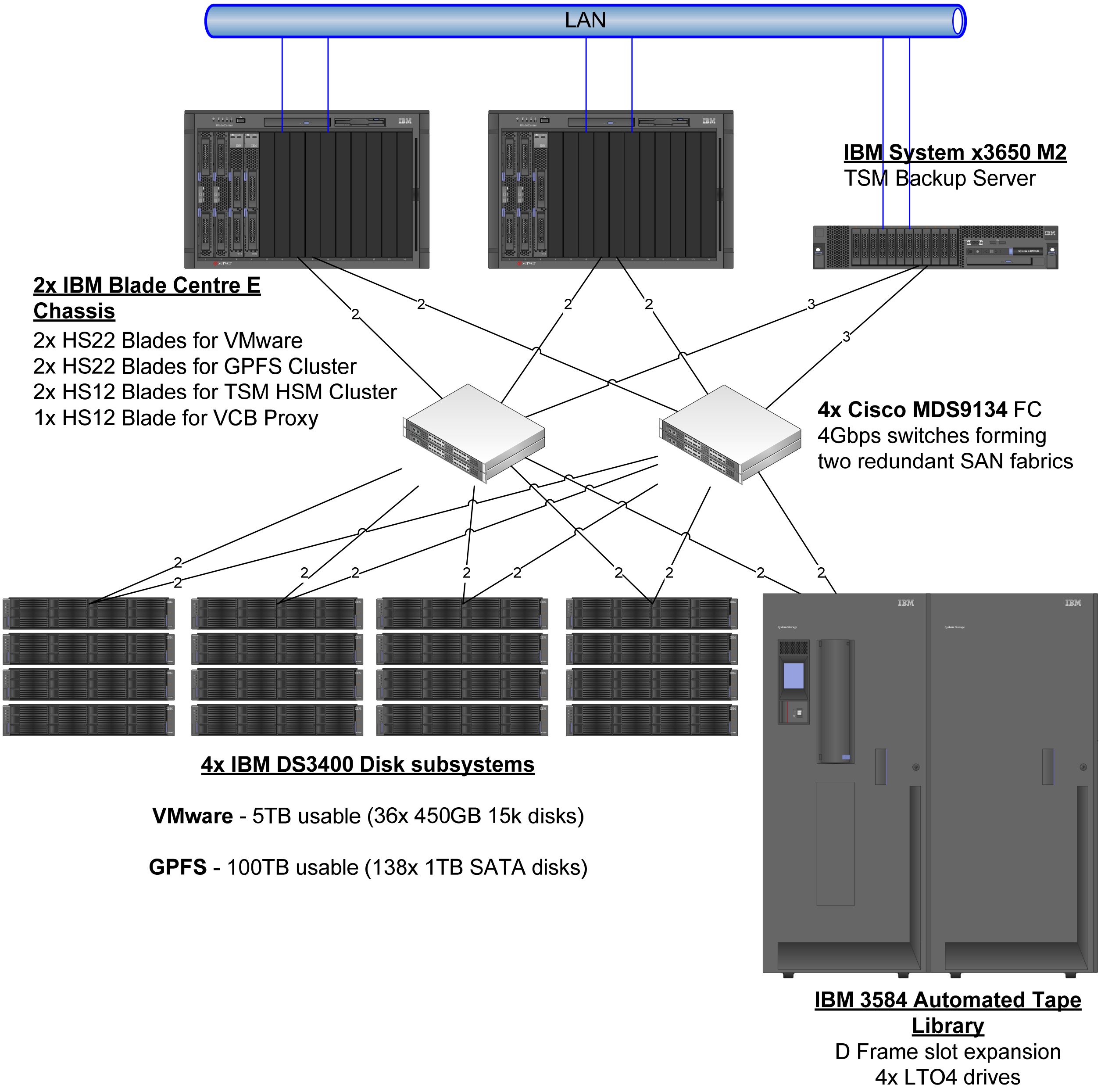

LAN
IBM System x3650 M2
TSM Backup Server
2x IBM Blade Centre E Chassis
2x HS22 Blades for VMware
2x HS22 Blades for GPFS Cluster
2x HS12 Blades for TSM HSM Cluster
1x HS12 Blade for VCB Proxy
4x Cisco MDS9134 FC 4Gbps switches forming two redundant SAN fabrics
2
3
4x IBM DS3400 Disk subsystems
VMware - 5TB usable (36x 450GB 15k disks)
GPFS - 100TB usable (138x 1TB SATA disks)
IBM 3584 Automated Tape Library
D Frame slot expansion
4x LTO4 drives

Figure 3-5: BHL-Europe Hardware and Network Configuration

3.3 Security

From a high level security standpoint the BHL-Europe system is a web based centralized system. It can be separated into three different categories which have appropriate levels of protection based upon the value of the information to the application end-users and the perception of threats to it:

1. An external public interface via the Portal component.

2. External private interfaces with partners to the OAIS_Ingest and OAIS_Access components.
3. Internal private interfaces between the components and for the administration of the system via OAIS_Administration

3.3.1 Network Architecture

The following diagram describes the different areas of the network.

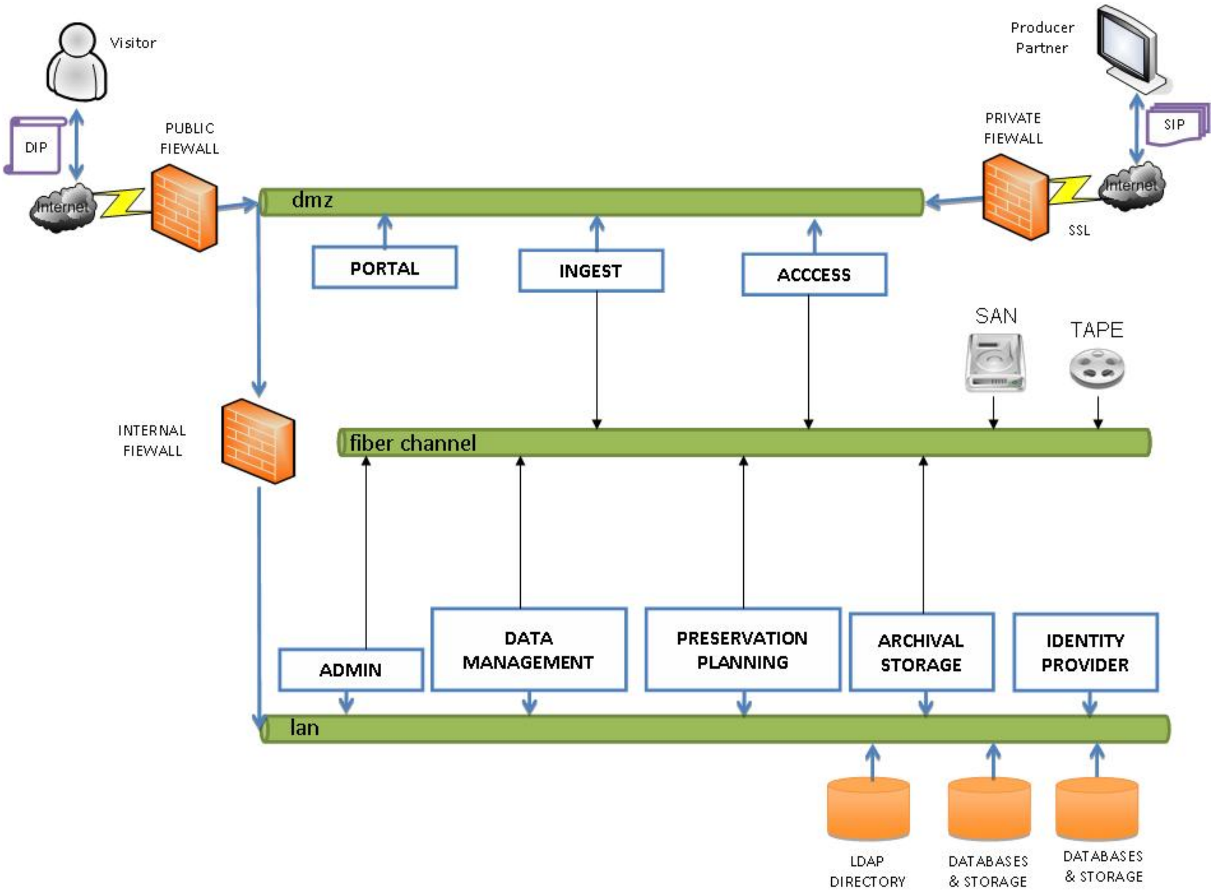


Figure 3-6: Network Security Configuration

This architecture aims to create different security zones depending on how sensitive the information and processes are in the information system. The different zones are:

- Public access to a DMZ with firewall interface to access the Portal component
- External private access to a DMZ with firewall interface to access the Ingest component and Access. This external private access will be via HTTPS / SSL
- Internal LAN between the remaining components and the databases
- A separate high speed fibre channel network giving all components access to the SAN and tape storage.

Each security zone is protected by firewalls with different policies of access defined in the firewalls. With a such architecture Administration, Archival Storage, Data Management, and Preservation Planning can only be accessed by the Portal , Access and Ingest components.

3.3.2 Software Implementation

3.3.2.1 Identification/authentication

Identification/authentication service confirms the identities of requesters for use of information system resources. In addition, authentication can apply to providers of data. The authentication service may occur at the initiation of a session or during a session.

Identity of data providers / Producer / Partner will be managed by an Identity Provider will be an Ldap directory using OpenLdap Server. Each OpenId account will be stored in the LDAP directory during the creation of a new data provider account and the will be attached to defined access policies stored in the Ldap Directory. Identity provider will be managed by Administraion component.

OpenID is an open, decentralized standard for authenticating users which can be used for access control, allowing users to log on to different services with the same digital identity where these services trust the authentication body.

3.3.2.2 Identity Provider component

Identity provider component a sub-module of Administration stores and provides all access policies to BHL-Europe components. The Identity Provider will be implemented using an open source LDAP server called OpenLDAP. This Identity Provider will store external producer / data provider OpenId identification and attach them to access policies.

3.3.2.3 Access control

Access control service prevents the unauthorized use of information system resources. This service also prevents the use of a resource in an unauthorized way. This service may be applied to various aspects of access to a resource (e.g., access to communications to the resource, the reading, writing, or deletion of an information/data resource, the execution of a processing resource) or to all accesses to a resource. Access policies will be stored in the LDAP directory. Access data and policies will be managed by Administraion.

Each BHL-Europe components will manage its access control by querying the Ldap Directory. Depending on the component, different client APIs can exist:

- Php-Ldap extension for php services
- JLDAP libraries for java services

4 Components

This chapter describes the components of the BHL-Europe system. As described in chapter 2 the components are based on the OAIS reference model. Each component within the BHL-Europe system will be implemented as already available systems. PreIngest and Ingest as well as Access and the Portal will make use of DISMARC multilingual search functionality concepts. As DISMARC does not have any form of preservation, ArchivalStorage will be implemented using Fedora. Each component described in the following subsections will contain best practice and standards. An overall view of the system can be seen in Figure 4-1 OAIS architecture.

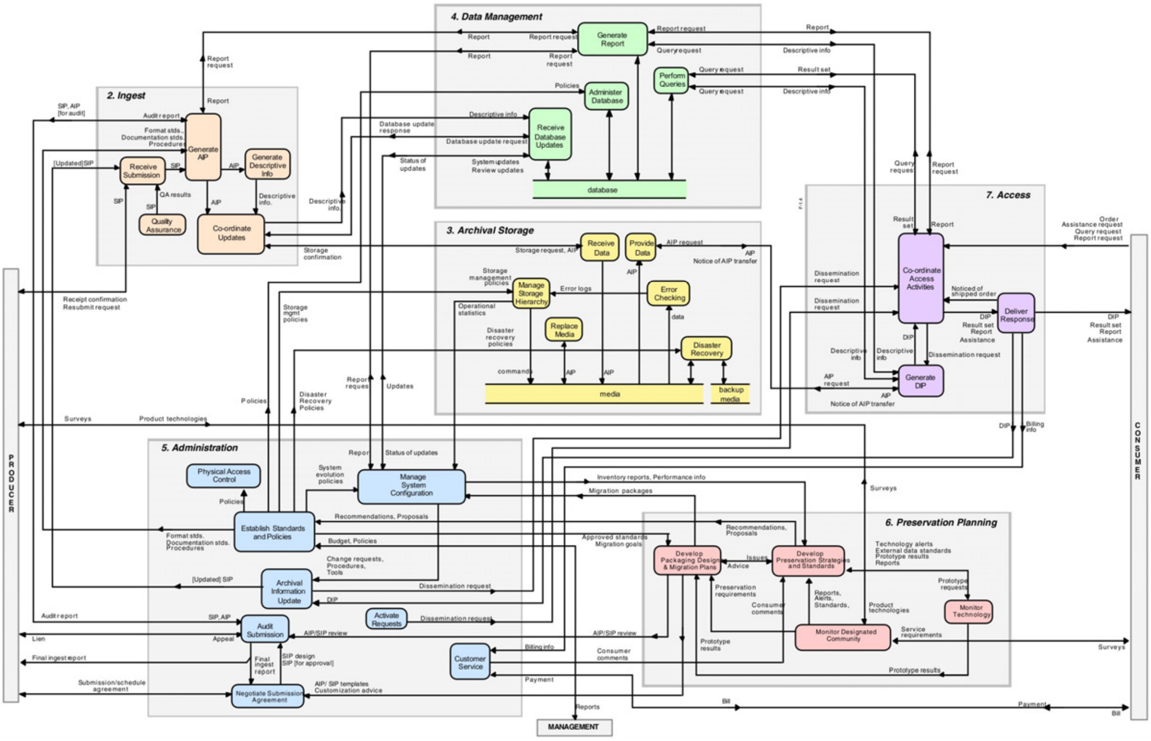


Figure 4-1 OAIS architecture

4.1 PreIngest

This component is the interface to the archives and acts as an adapter for the Ingest module. As external partners store meta data in various formats, the native formats need to be converted, harmonized, enriched and prepared for ingestion. This step is needed for the ingestion, multilingual search, data harmonization, indexation and search requirement.

4.1.1 Processes

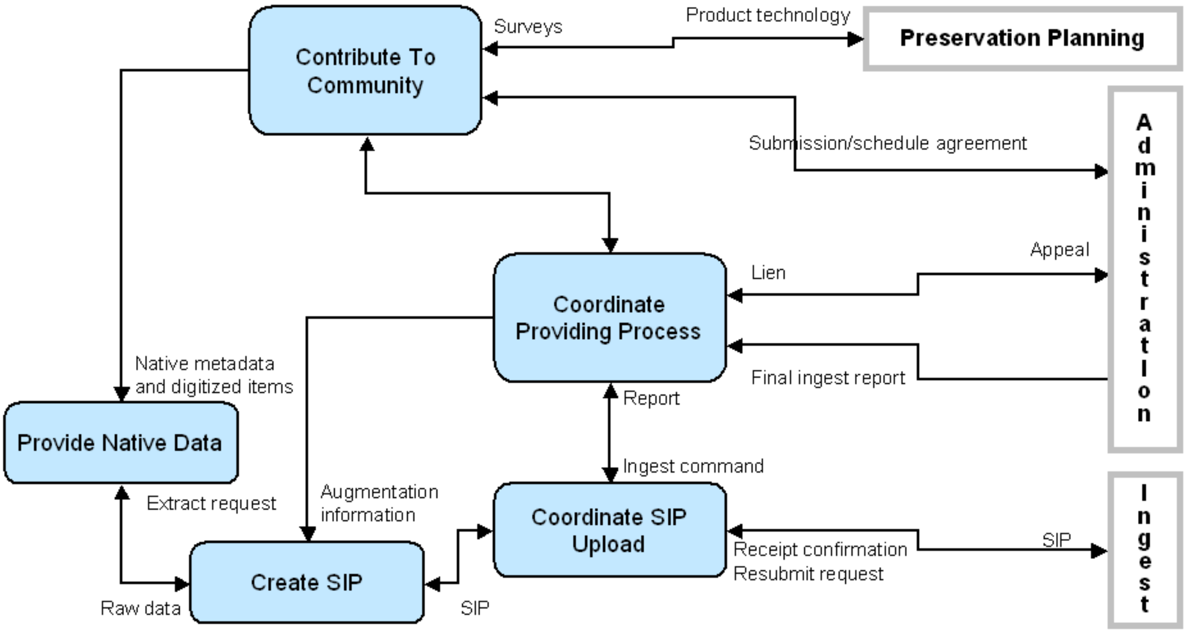


Figure 4-2 PreIngest processes

The **Contribute to Community** function initializes the contribution to BHL. Preservation Planning will be informed about the product technology in use. Administration will negotiate a submission agreement. The Provide Native Data function will be initialized as well as the Coordinate Providing Process.

The **Provide Native Data** function can not be defined generally for all providers. Each provider will carry out metadata collection and digitizing processes on their own with specific tools.

The **Coordinate Providing Process** function has to setup data augmentation information like mappings of raw data and services used to augment the digitized data while creating the SIP. When SIPs have been sent to Ingest, Administration may return Liens for these SIPs. These Liens are appealed by this function or a new ingest command will be sent to Coordinate SIP Upload to resend the SIP with new augmentation information. The final ingest report provided by Administration will be presented by this function as well.

The **Coordinate SIP Upload** function receives an ingest command and starts to request SIPs from the Create SIP function. These SIPs will be sent to Ingest and possible resubmit requests are handled. After all SIPs affected by the ingest command have been sent, a report for the Coordinate Providing Process will be created.

The **Create SIP** function starts by receiving a SIP request from the Coordinate SIP Upload function. Augmentation information defined within the Coordinate Providing Process are used to transform the raw data provided by the Provide Native Data function into a SIP. The raw data will be harmonization and augmented with services like OCR or vocabulary alignment.

4.1.2 Standards

The standards used within this component are based on vocabulary and metadata augmentation. Since this augmented information is used at the Portal, the standards will be used there as well.

4.1.2.1 Vocabulary Representation SKOS [W3C standard]

The Simple Knowledge Organisation Systems (SKOS)[2] is a family of formal languages designed for representation of thesauri, classification schemes, taxonomies, subject-heading systems, or any other type of structured controlled vocabulary. SKOS is built upon RDF[3] and RDFS[4], and its main objective is to enable easy publication of controlled structured vocabularies for the Semantic Web. SKOS is currently developed within the W3C framework.

It is used for export and import of controlled vocabulary within some thesaurus services used by the portal.

4.1.2.2 ISO 2788:1986 Guidelines for the Establishment and Development of Monolingual Thesauri [ISO standard][5]

An International Organization for Standardization (ISO) standard for the construction of monolingual thesauri. Includes guidelines for dealing with references, definitions, abbreviations, vocabulary control, indexing terms, compound terms, basic relationships, display, and management. Available for sale at www.iso.org

4.1.2.3 ANSI/NISO Z39.19-2005 Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies. [ANSI standard]

This American standard shows how to formulate descriptors, establish relationships among terms, and present the information in print and on a screen. Included are thesaurus maintenance procedures and recommended features for thesaurus management systems. This standard is available online.

[2] http://www.w3.org/2004/02/skos/ [30.10.2009]

[3] http://www.w3.org/TR/rdf-concepts/ [30.10.2009]

[4] http://www.w3.org/TR/rdf-schema/ [30.10.2009]

[5] http://www.iso.org/iso/catalogue_detail.htm?csnumber=7776 [30.10.2009]

4.1.2.4 ISO 5964 Guidelines for the Establishment and Development of Multilingual Thesauri [ISO standard][6]

An International Organization for Standardization (ISO) standard for the construction of multilingual thesauri. This standard is to be used as an extension of the monolingual guidelines (ISO 2788). Includes guidelines for dealing with degrees of term equivalence and non-equivalence, single-to-multiple term equivalence, etc. Also includes examples of thesaural displays. Print version available for sale.

4.1.2.5 ISO/DIS 25964

“Information and documentation -- Thesauri and interoperability with other vocabularies -- Part 1: Thesauri for information retrieval” revises the monolingual thesaurus standard ISO 2788:1986 and the multilingual thesaurus standard ISO 5964:1985.[7]

4.1.3 Implementation

The Pre-Ingest component has to be customized for each of the providers. Figure 4-3 shows the custom made modules within this component.

The *custom mapping and data augmentation* module has to transform and augmented the native data with external vocabulary services. In the first phase, *uBio* will be used for this functionality. The remaining modules are generic and will be set up as either local Java clients or PHP/Java based web application based on the requirements of the partner. Within Pre-Ingest, three main processes exist, one for registration, one for creating the SIP and one for uploading.

An overview of the registration process is shown by Figure 4-4. As Pre-Ingest is not part of the OAIS reference model, it has to register at Administration which is in control of adding providers and their data. All theses steps can be run manually via email or phone. Pre-Ingest modules can be located at the partner location where data is extracted from a local repository and submit as various SIPs to Ingest.

When all policies, agreements and contracts are signed, the actual ingest can start. This can be a one procedure or, as described here, a complete extract, transform, load (ETL) cycle. Figure 4-5 and Figure 4-6 show these steps.

[6] http://www.iso.org/iso/catalogue_detail.htm?csnumber=12159 [30.10.2009]

[7] http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=53657 [18.01.2010]

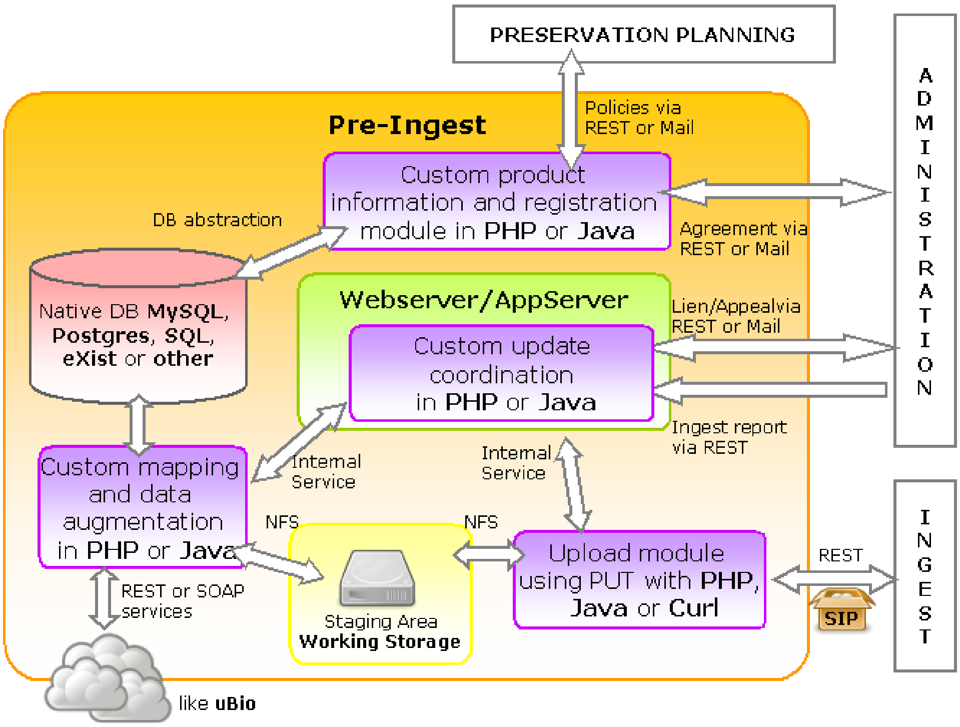

PRESERVATION PLANNING
Pre-Ingest
Policies via REST or Mail
Custom product information and registration module in PHP or Java
DB abstraction
Agreement via REST or Mail
Native DB MySQL, Postgres, SQL, eXist or other
Webserver/AppServer
Lien/Appealvia REST or Mail
Custom update coordination in PHP or Java
Ingest report via REST
Custom mapping and data augmentation in PHP or Java
Internal Service
Internal Service
NFS
NFS
Upload module using PUT with PHP, Java or Curl
REST
SIP
REST or SOAP services
Staging Area Working Storage
like uBio
ADMINISTRATION
INGEST

Figure 4-3 Pre-Ingest Architecture

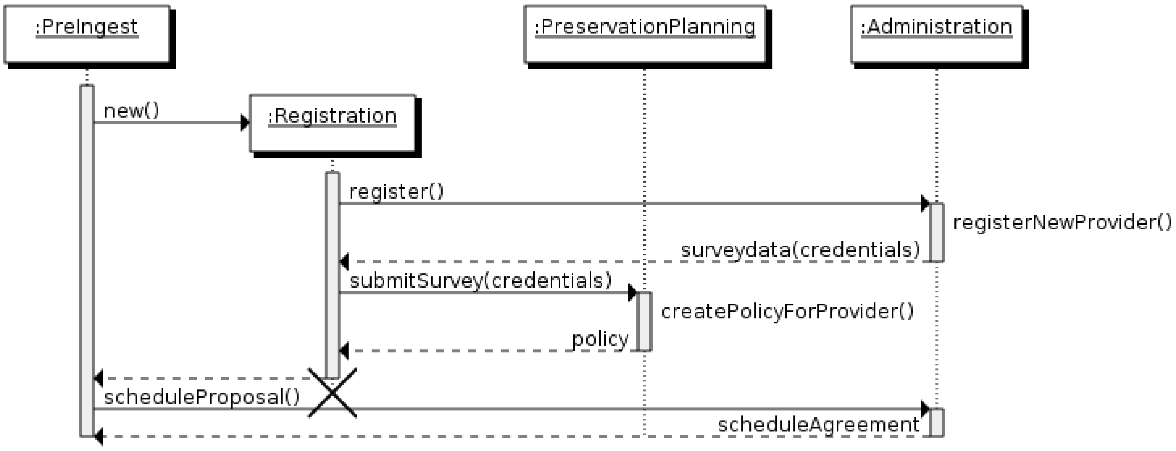


Figure 4-4 PreIngest componant registration

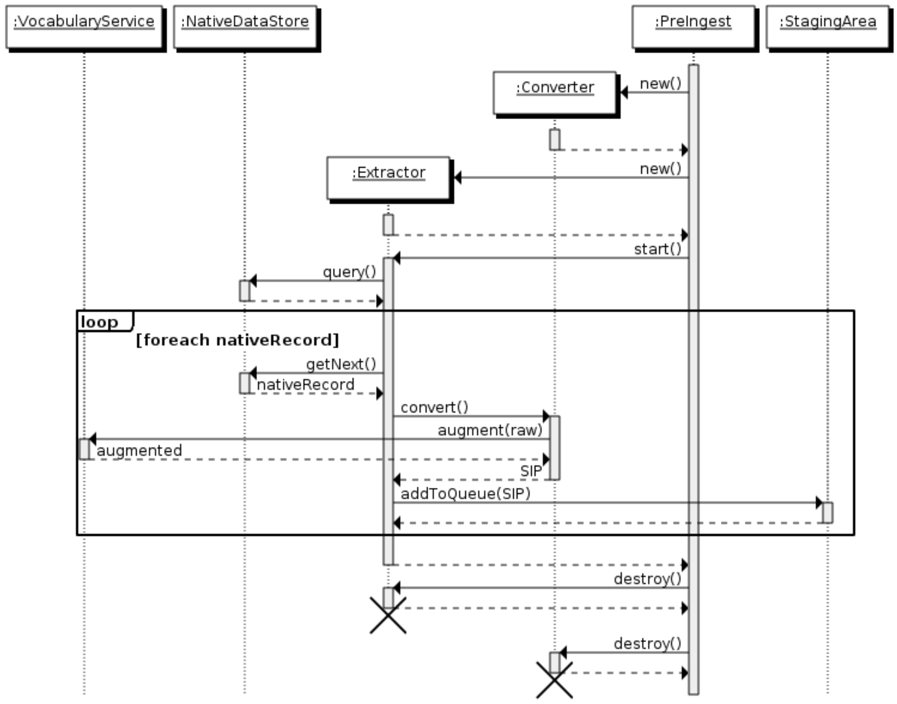


Figure 4-5 Extract and Transform native data to SIPs

As soon as the SIPs are stored in the staging area (they can also be created on the fly) they can be uploaded to the Ingest module. When uploaded, the SIPs need to be checked and external resources need to be downloaded there. For BHL-Europe, an asynchronous check procedure utilizing polling will be used.

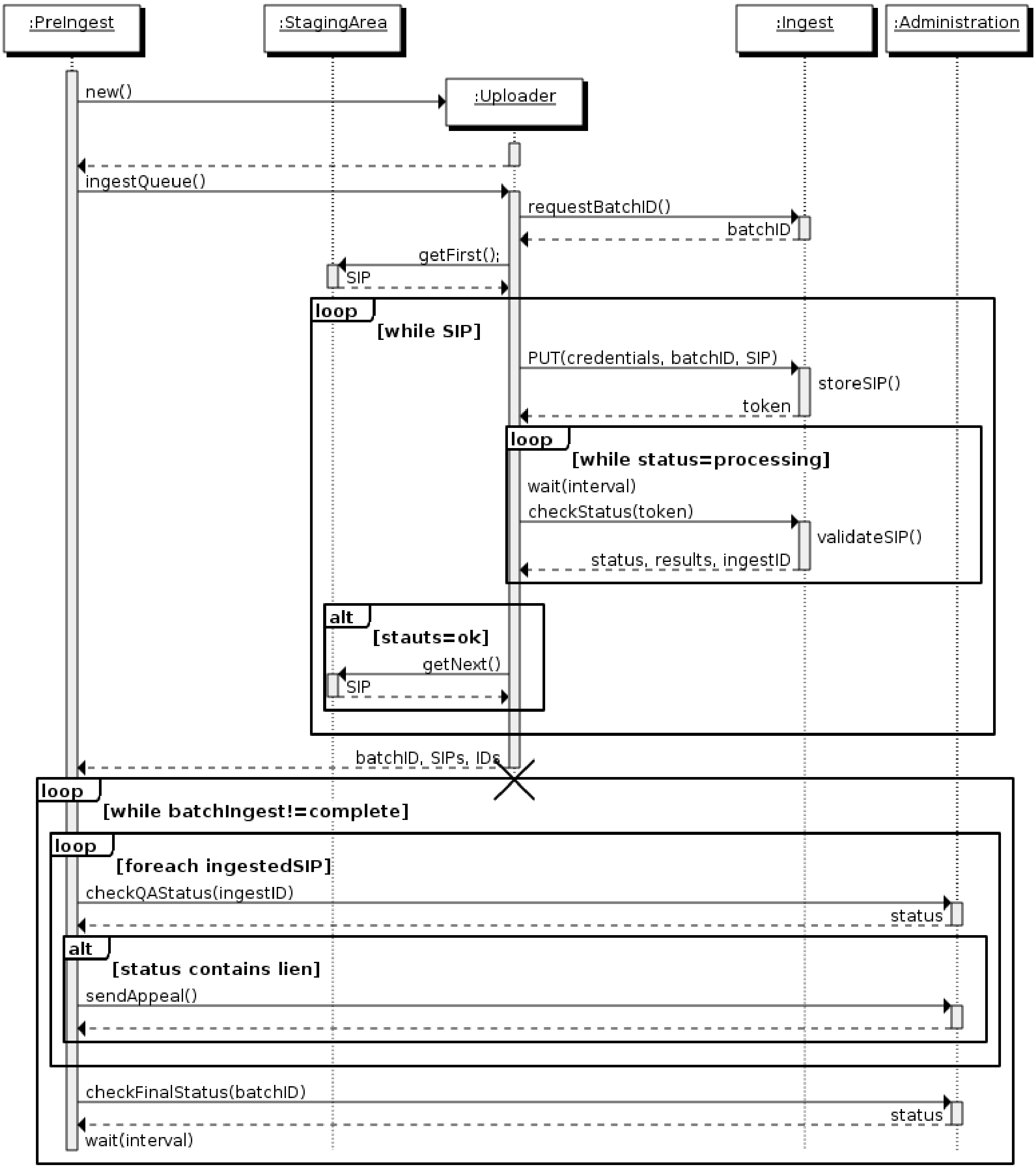


Figure 4-6 Ingest process as seen by Pre-Ingest

When all SIPs are stored successfully in Ingest’s staging area, the Administration component can be polled to check for quality assurance reports. OAIS’s Lien/Appeal procedure is run there. The final ingest report can be obtained in the end of an ingest procedure.

For replication, other OAIS Systems like Archivematica[8] will be used as Pre-Ingest modules.

4.2 Ingest

This component provides the services and functions to accept Submission Information Packages (SIPs) from the PreIngest module or Producers directly (or from internal elements under Administration control) and prepare the contents for storage and management within the archive. Ingest functions include receiving SIPs, performing quality assurance on SIPs, generating an Archival Information Package (AIP) which complies with the archive's data formatting and documentation standards, extracting Descriptive Information from the AIPs for inclusion in the archive database, and coordinating updates to Archival Storage and Data Management.

4.2.1 Processes

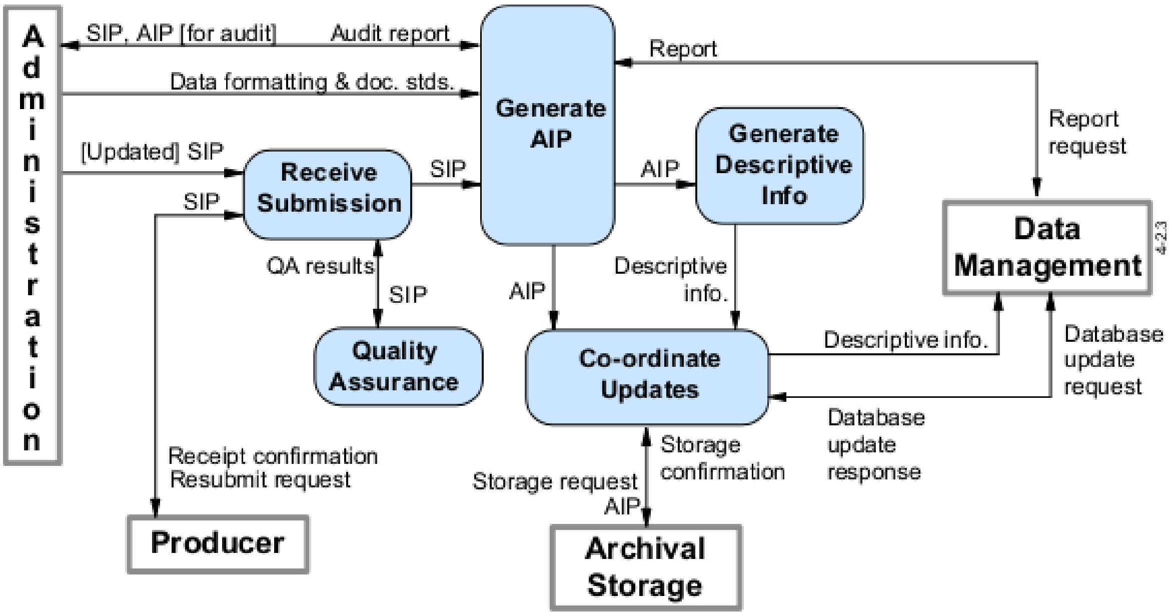


Figure 4-7 Ingest processes

The **Receive Submission** function provides the appropriate storage capability or devices to receive a SIP from the Producer (or from Administration). The SIPs are delivered via electronic transfer (e.g., FTP, REST Service, Cloud Storage Service). The Receive Submission function will check for access rights with the payload send within the SIP. This

[8] Archivematica: http://archivematica.org

function provides a confirmation of receipt of a SIP to the Producer, which may include a request to resubmit a SIP in the case of errors resulting from the SIP submission.

The **Quality Assurance** function validates (QA results) the successful transfer of the SIP to the staging area. For digital submissions, these mechanisms include Cyclic Redundancy Checks (CRCs) or checksums associated with each data file as well as syntactical schema validation.

The **Generate AIP** function transforms SIPs into one or more AIPs that conform to the archive's data formatting and documentation standards. This involves file format conversions and file aggregation. The Generate AIP function may issue report requests to Data Management to obtain reports of information needed by the Generate AIP function to produce the Descriptive Information that completes the AIP. This function sends SIPs or AIPs for audit to the Audit Submission function in Administration, and receives back an audit report.

The **Generate Descriptive Information** function extracts Descriptive Information from the AIPs and collects Descriptive Information from other sources to provide to Coordinate Updates, and ultimately Data Management. This includes metadata to support searching and retrieving AIPs (e.g., who, what, when, where, why), and will also include special browse products (thumbnails, images, document structures) populated by Access and used by the Portal.

The **Coordinate Updates** function is responsible for transferring the AIPs to Archival Storage and the Descriptive Information to Data Management. Transfer of the AIP includes a storage request which results in an electronic transfer to a cloud like storage infrastructure. After the transfer is completed and verified, Archival Storage returns a storage confirmation indicating the storage identification information for the AIP. The Coordinate Updates function also incorporates the storage identification information into the Descriptive Information for the AIP and transfers it to the Data Management entity along with a database update request. In return, Data Management provides a database update response indicating the status of the update. Data Management updates may take place without a corresponding Archival Storage transfer when the SIP contains Descriptive Information for an AIP already in Archival Storage.

4.2.2 Standards

As this component is a adapter and facade for other components, next to the payload and interfaces are no standards available or are part of other components.

4.2.2.1 SIP - METS Profile with MARCXML

The standards in this component are reduced to the ingest format which is based on a METS profile and the MARCXML metadata format. These XML schemata are very common in libraries. Please see the SIP for more information about these standards.

4.2.3 Implementation

The Ingest component has must validate the SIP and run a basic QA before sending it to the Administration component. When Administration approves the SIPs, they will be transferred to Data Management and Archival Storage. Figure 4-3 shows the custom made modules within this component.

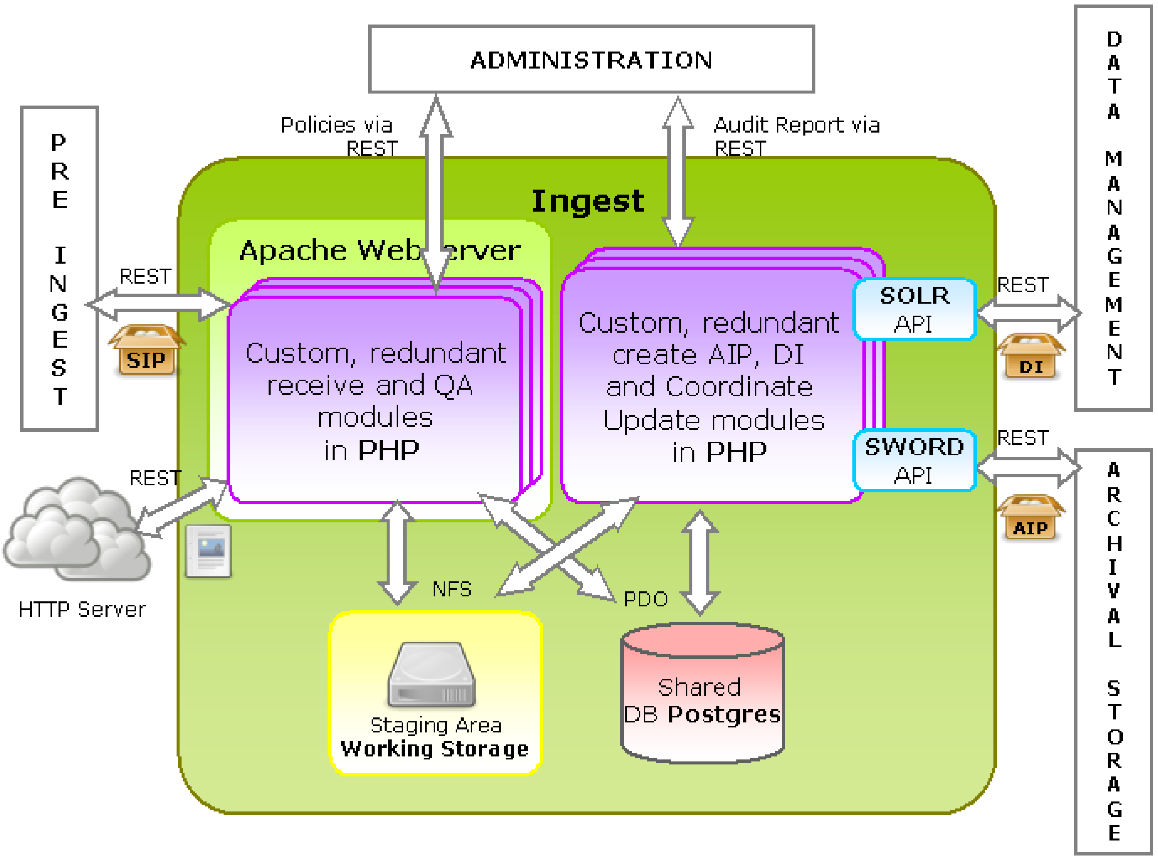


Figure 4-8 Ingest Architecture

Derived from the modules, the sequence diagrams in Figure 4-9 and Figure 4-10 represent the activities within this component.

First, SIPs are uploaded to Ingest where they are stored in a staging area. External resources referenced within the SIP are downloaded to the staging area as well. SIP and resources are validated and converted for the AIP. Meanwhile, the PreIngest component will poll the Ingest component for the status of the validation. This step can take longer in case many resources are referenced and not sent inline with the SIP.

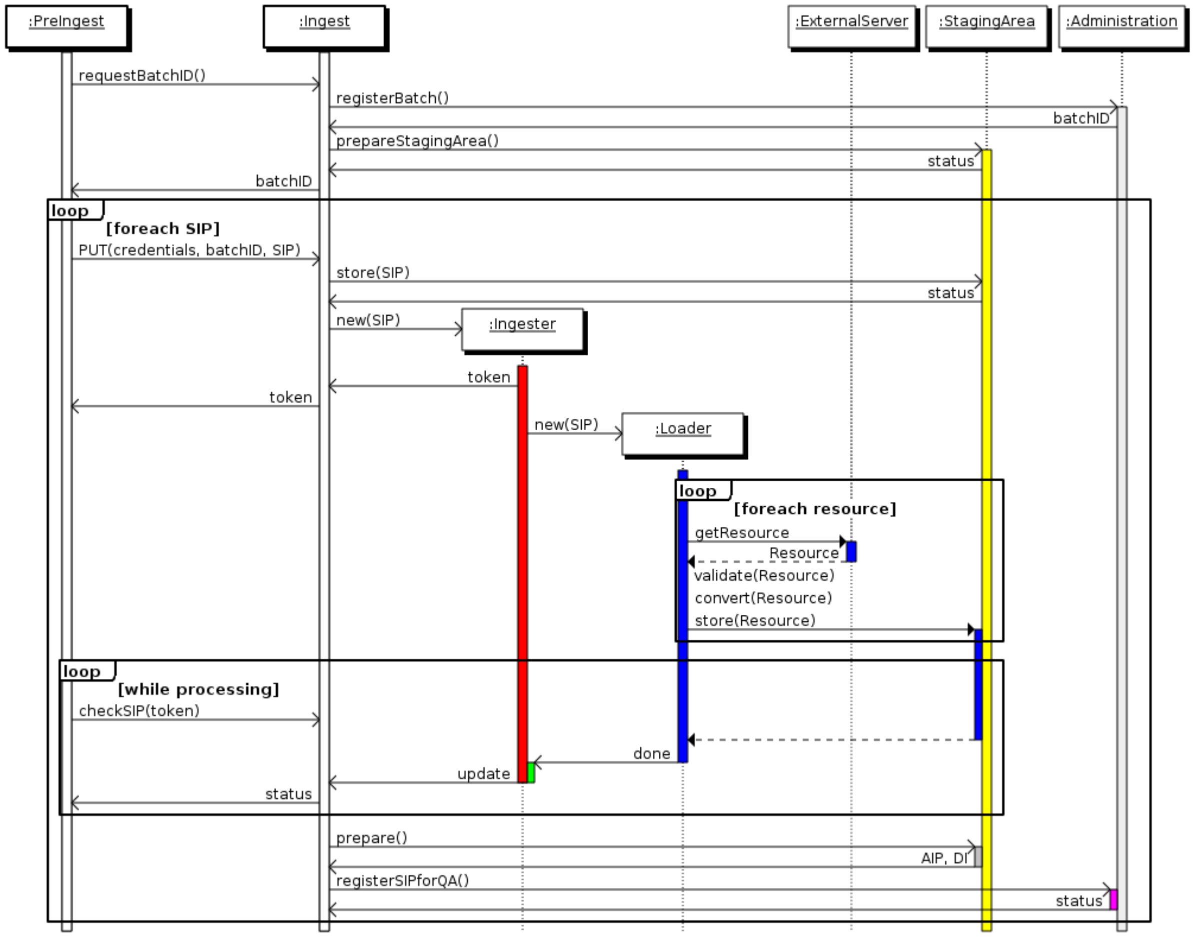


Figure 4-9 Ingesting to staging area

As soon as the SIP is complete and converted the polled check will return a status to either resubmit the SIP or that the SIP is now registered in Administration as part of the batch for quality assurance. Quality assurance and the lien/appeal process is part of Administration. When Administration acknowledges the batch, it can be transferred to DataManagement and ArchivalStorage. Items are not available in Access until they are transferred to these components.

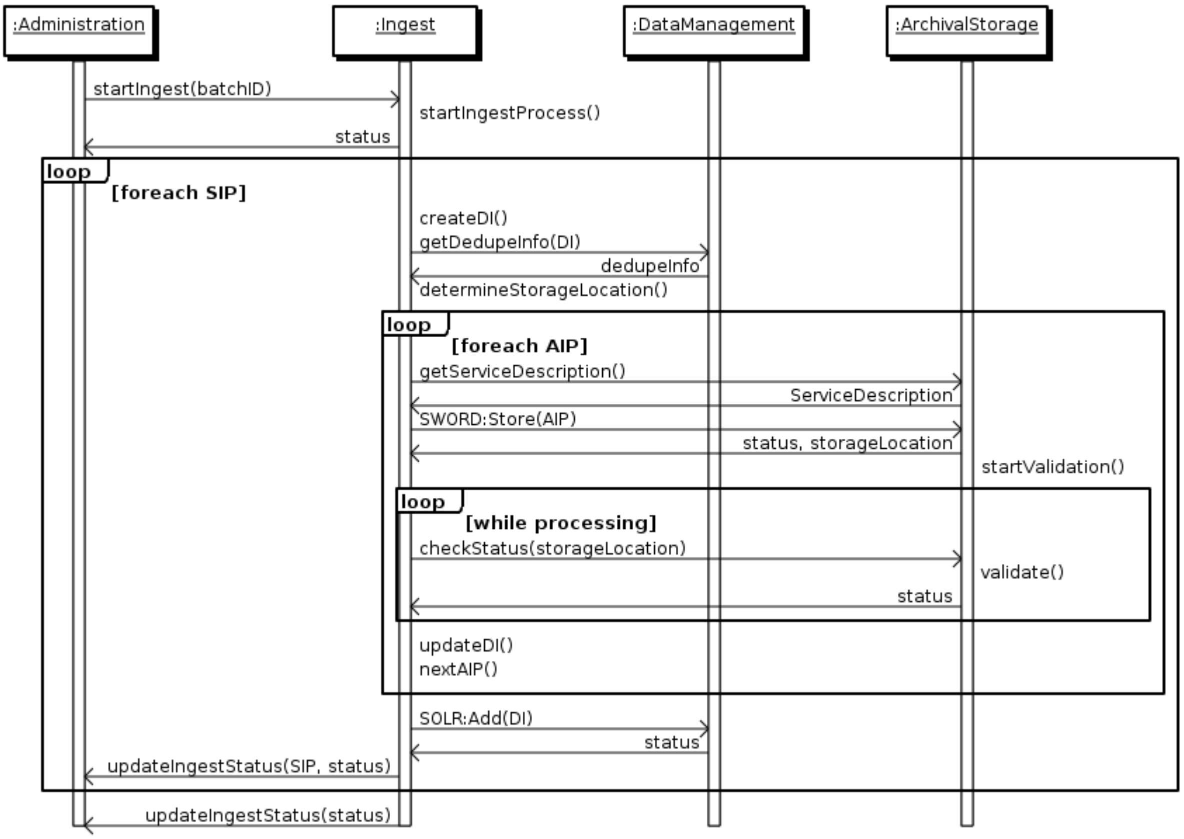


Figure 4-10 Ingesting to DataManagement and ArchivalStorage

The Descriptive Information of an item has to be updated with the actual storage location of the AIP before it’s sent to DataManagement so it can be found.

4.3 ArchivalStorage

This component provides the services and functions for the storage, maintenance and retrieval of AIPs. Archival Storage functions include receiving AIPs from Ingest and adding them to permanent storage, managing the storage hierarchy, refreshing the media on which archive holdings are stored, performing routine and special error checking, providing disaster recovery capabilities, and providing AIPs to Access to fulfill orders.

4.3.1 Process

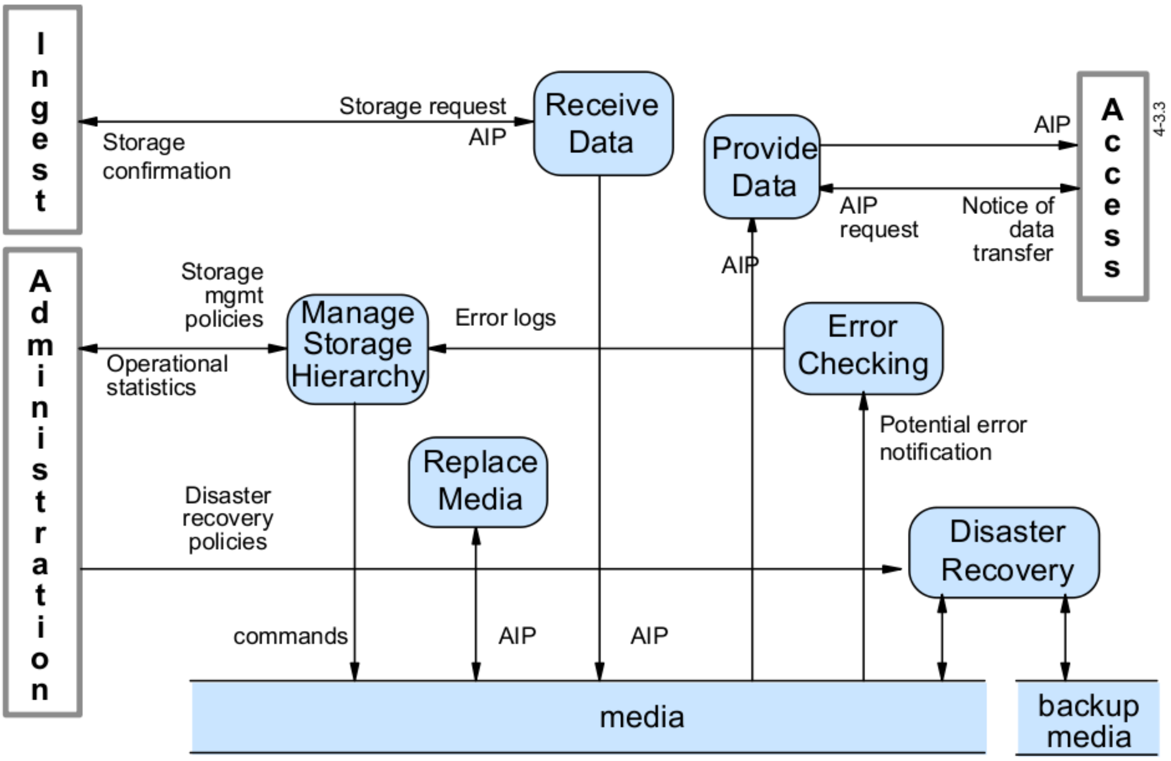


Figure 4-11 Archival Storage processes

The **Receive Data** function receives a *storage request* and an *AIP* from Ingest and moves the *AIP* to permanent storage within the archive. The transfer request may need to indicate the anticipated frequency of utilization of the data objects comprising the AIP in order to allow the appropriate storage devices or media to be selected for storing the AIP. This function will select the media type, prepare the devices or volumes, and perform the physical transfer to the Archival Storage volumes. Upon completion of the transfer, this function sends a *storage confirmation* message to Ingest, including the storage identification of the AIPs.

The **Manage Storage Hierarchy** function positions, via *command*s, the contents of the AIPs on the appropriate media based on storage *management policie*s, operational statistics, or directions from Ingest via the storage request. It will also conform to any special levels of service required for the AIP, or any special security measures that are required, and ensures the appropriate level of protection for the AIP. These include on-line, off-line or near-line storage, required throughput rate, maximum allowed bit error rate, or special handling or backup procedures. It monitors *error logs* to ensure AIPs are not corrupted during transfers. This function also provides *operational statistics* to Administration summarizing the inventory of media on-hand, available storage capacity in the various tiers of the storage hierarchy, and usage statistics.

The **Replace Media** function provides the capability to reproduce the *AIPs* over time. Within the Replace Media function the Content Information and Preservation Description Information (PDI) must not be altered. However, the data constituting the Packaging Information may be changed as long as it continues to perform the same function and there is Implementation and

4.3.2 Standards

4.3.2.1 SWORD http://www.swordapp.org/

SWORD is a lightweight protocol for depositing content from one location to another. It stands for Simple Web-service Offering Repository Deposit and is a profile of the Atom Publishing Protocol (known as APP or ATOMPUB).

SWORD has been funded by the Joint Information Systems Committee to develop the SWORD profile and a number of demonstration implementations.

4.3.2.2 AtomPub http://www.ietf.org/rfc/rfc4287.txt

The Atom Publishing Protocol (AtomPub) is an application-level protocol for publishing and editing Web resources. The protocol is based on HTTP transfer of Atom-formatted representations. The Atom format is documented in the Atom Syndication Format.

4.3.2.3 Network File System (NFS) http://tools.ietf.org/html/rfc3530

Network File System (NFS) is a network file system protocol originally developed by Sun Microsystems in 1984,[1] allowing a user on a client computer to access files over a network in a manner similar to how local storage is accessed. NFS, like many other protocols, builds on the Open Network Computing Remote Procedure Call (ONC RPC) system. The Network File System is an open standard defined in RFCs, allowing anyone to implement the protocol.

4.3.3 Implementation

The following diagram shows the main software modules used to implement Archival Storage.

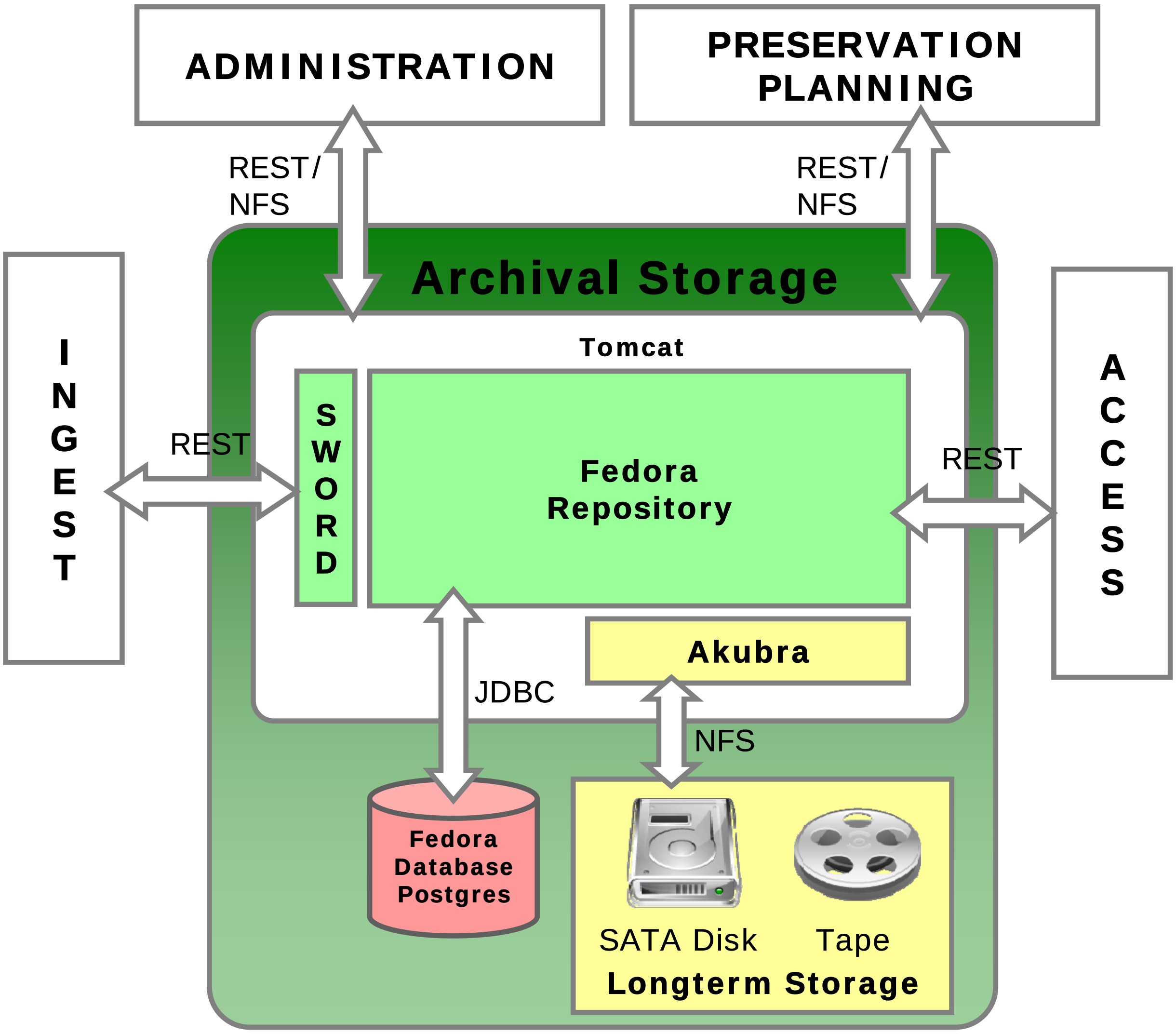


Figure 4-12 Archival Storage Architecture

4.3.3.1 SWORD plugin for Fedora

The **Ingest module** deposits AIPs via the RESTful SWORD protocol. The SWORD plugin for Fedora is a WAR file deployed on the Tomcat server.

Depositing is a two-stage process. First, a request from Ingest is sent to the SWORD plugin for the 'service document', this returns details of the collections that user is allowed to deposit to within the repository. At this point, Ingest deposits their file into the chosen collection.

The SWORD interface can accept single image files (gif or jpeg) which it stores in a new Fedora object with a single datastream, or it can accept zip files where it stores each file contained in the zip file in a separate datastream. Alternatively it can accept METS documents where it separates each dmdSec (descriptive metadata section) into a datastream, or a zip file with a METS manifest which it can handle in a similar manner.

Various things may prevent success, for example lack of authentication credentials, unacceptable file format or a corrupt MD5 checksum. The repository will send a respond indicating the success, or otherwise of the deposit.

SWORD detailed specifications: http://www.swordapp.org/docs/sword-profile-1.3.html

SWORD plug-in for fedora information: (http://www.fedora-commons.org/confluence/display/FCSVCS/SWORD-Fedora+1.2)

4.3.3.2 Fedora (http://www.fedora-commons.org)

Fedora is the core software component of the Archival Storage module. It is used to manage, store, and link the digital assets to be stored. The only component used in the Archival Storage component is the core Fedora Repository. Other Fedora Commons modules may or may not be used as described below:

- RISearch, GSearch, are not used as they will be implemented separately in the Data Management component.
- SIPIngest is not used as a custom implementation is required in the Ingest component.
- Data Transformation WAR's such as Imagemanip, saxon, and fop may be used in the Access component.

The Access component will retrieve AIPs via the **Fedora Access** service which exposes REST interface for accessing digital objects. The access operations include methods to do reflection on a digital object and to request disseminations. In our case the disseminations will not include any transformations as that will be handled by the Access component. The digital objects may be internal to the repository, or it may be a web service external to the repository. In our case all digital objects will be internal to the repository.

The Fedora REST API exposes a subset of the Fedora Access and Management APIs as a RESTful (Representational State Transfer) Web Service.
REST API detailed specifications: http://fedora-commons.org/confluence/display/FCR30/REST+API

4.3.3.3 Tomcat Application Server

The Fedora repository, and the SWORD-fedora plug-in, as well as other fedora framework services such as the disseminators are all packaged as J2EE web archives (WAR). They will be deployed in a J2EE servlet container Tomcat.

Tomcat requires a Java Development Kit (JDK). We will use JDK 6 or higher.

4.3.3.4 Postgresql Database

Fedora uses a relational database to support some of its functions. Fedora supports three external databases: MySQL, Oracle and PostgreSQL. PostgreSQL will be used.

4.3.3.5 Akubra (http://www.fedora-commons.org/confluence/display/AKUBRA/Akubra+Project)

Akubra provides a pluggable file storage interface that can be adapted to almost any storage subsystem. Akubra supports both ordinary and transactional storage systems, but makes simplifying assumptions in order to achieve a high level of interoperability between storage systems:
In Akubra:

- a Blob is a finite-length bitstream with an id (a URI)
- a Blob Store is primarily concerned with providing read/write access to blobs.

Akubra will allow the file storage system to be more easily replaced if necessary.

4.4 PreservationPlanning

This component provides the services and functions for monitoring the environment of the OAIS and providing recommendations to ensure that the information stored in the OAIS remains accessible to the Designated User Community over the long term, even if the original computing environment becomes obsolete. Preservation Planning functions include evaluating the contents of the archive and periodically recommending archival information updates to migrate current archive holdings, developing recommendations for archive standards and policies, and monitoring changes in the technology environment and in the Designated Community’s service requirements and Knowledge Base. Preservation Planning also designs IP templates and provides design assistance and review to specialize these templates into SIPs and AIPs for specific submissions. . Preservation Planning also develops detailed Migration plans, software prototypes and test plans to enable implementation of Administration migration goals.

4.4.1 Process

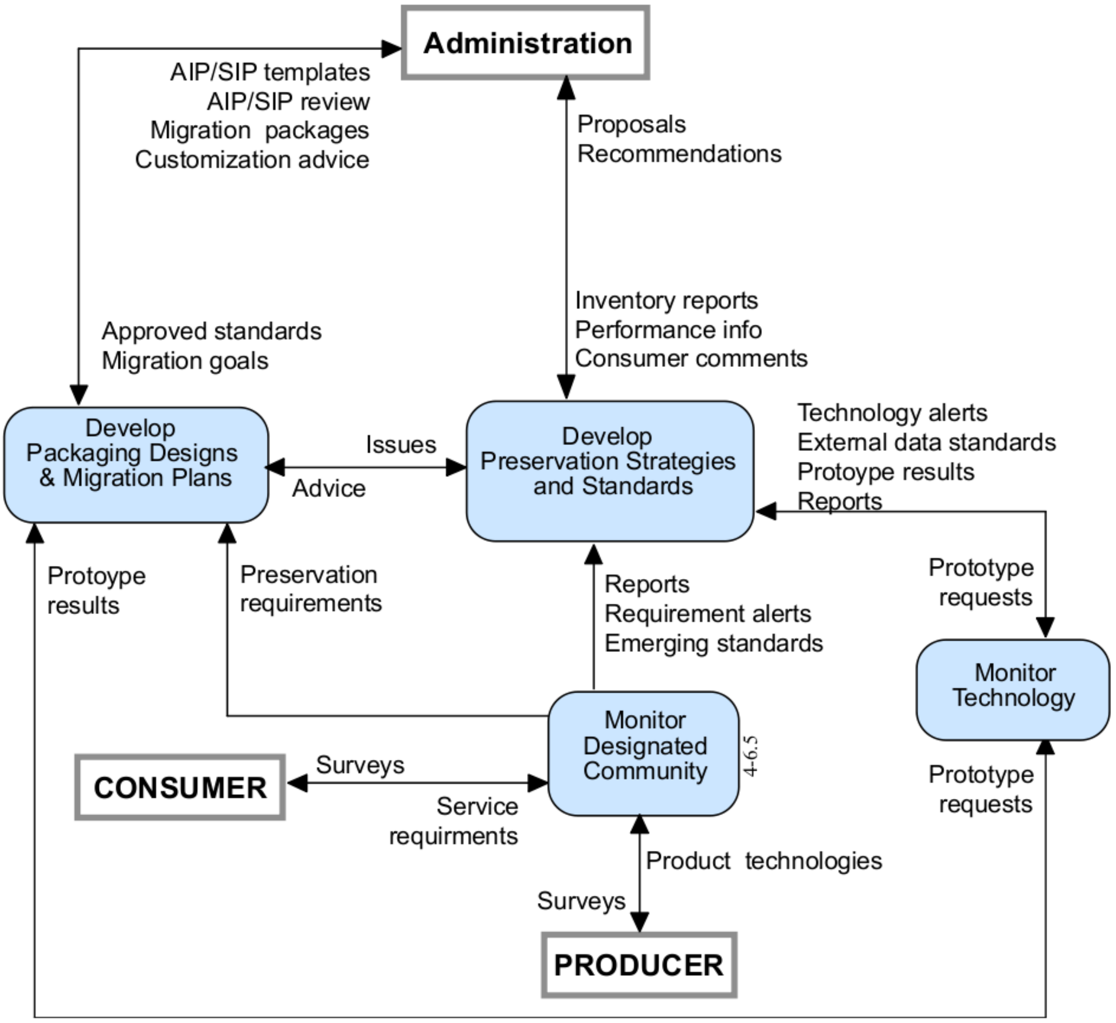


Figure 4-13 Preservation Planning processes

The **Monitor Designated Community** function interacts with archive Consumers and Producers to track changes in their service requirements and available product technologies. Such requirements might include data formats, media choices, preferences for software packages, new computing platforms, and mechanisms for communicating with the archive. This function may be accomplished via surveys, via a periodic formal review process, via community workshops where feedback is solicited or by individual interactions. It provides reports, requirements alerts and emerging standards to the Develop Preservation Strategies and Standards function. It sends preservation requirements to Develop Packaging Designs.

The **Monitor Technology** function is responsible for tracking emerging digital technologies, information standards and computing platforms (i.e., hardware and software) to identify technologies which could cause obsolescence in the archive's computing environment and prevent access to some of the archives current holdings. This function may contain a prototyping capability for better evaluation of emerging technologies and receive prototype requests from Develop Preservation Strategies and Standards and from Develop Package Designs and Migration Plans. This function sends reports, external data standards, prototype results and technology alerts to Develop Preservation Strategies and Standards. It also sends prototype results to Develop Package Designs and Migration Plans.

The **Develop Preservation Strategies and Standards** function is responsible for developing and recommending strategies and standards to enable the archive to better anticipate future changes in the Designated Community service requirements or technology trends that would require migration of some current archive holdings or new submissions. This function receives reports from the Monitor Designated Communities and Monitor Technology functions, and it receives performance information, inventory reports and summarized consumer comments from Administration. This function sends recommendations on system evolution to Administration. This function also receives external data standards from Monitor Technology and produces profiles of those standards that are sent to Administration as proposals on their potential usage. This function also receives issues from Develop Packaging Designs and Migration Plans in the case of unanticipated submission requirements, and responds with advice to handle the new requirements.

The **Develop Packaging Designs and Migration Plans** function develops new IP designs and detailed migration plans and prototypes, to implement Administration policies and directives. This activity also provides advice on the application of these IP designs and Migration plans to specific archive holdings and submissions. This function receives archive approved standards and migration goals from Administration. The standards include format standards, metadata standards and documentation standards. It applies these standards to preservation requirements and provides AIP and SIP template designs to Administration.This function also provides customization advice and AIP/SIP review to Administration on the application of those designs. If this function encounters submissions that are not covered by existing standards and procedures, it can send issues to Develop Preservation Strategies and Standards and receive advice, including new standards, to assist in meeting the new submission requirements.

The migration goals received by this function tend to involve transformations of the AIP, including transformations of the Content Information to avoid loss of access due to technology obsolescence. The response to the migration goals may involve the development of new AIP designs, prototype software, test plans, community review plans and implementation plans for phasing in the new AIPs. This process may call on expertise or resources from other functions within Preservation Planning, such as prototype development from the Monitor Technology. This effort also will require consultation from the other functional areas and from the Designated Community. Once the migration plan, associated AIP designs, and software have been tested and approved, this function will send the entire migration package to Administration, which will schedule and perform the actual migration.

4.4.2 Implementation

We are currently evaluating Plato, the Planets Preservation Planning tool (http://www.ifs.tuwien.ac.at/dp/plato/intro.html) to see if it meets the functional requirements of BHL-Europe.

4.4.2.1 Plato

The Planets preservation planning methodology defines measurable requirements for preservation strategies in a hierarchical form and evaluates them in a standardized setting to arrive at a recommendation for a solution. The procedure is independent of the solutions considered; it can be applied for any class of strategy, be it migration, emulation or different approaches, and has been validated in a series of case studies.

The Planets preservation planning workflow consists of three main stages:

1. Requirements definition is the natural first step in the planning procedure, collecting requirements from the wide range of stakeholders and influence factors that have to be considered for a given institutional setting. This includes the involvement of curators and domain experts as well as IT administrators and consumers. Requirements are specified in a quantifiable way, starting at high-level objectives and breaking them down into measurable criteria, thus creating an objective tree which forms the basis of the evaluation of alternative strategies. Furthermore, as this evaluation would be infeasible on the potentially very large collection of objects, the planner selects representative sample objects that should cover the range of essential characteristics present in the collection at hand.

2. The evaluation of potential strategies is carried out empirically by applying selected tools to the defined sample content and evaluating the outcomes against the specified requirements.

3. Analysis of the results takes into account the different weighting of requirements and allows the planner to arrive at a well-informed recommendation for a solution to adopt.

Plato implements this three stage workflow and includes additional external services to automate the process. It further extends it with a fourth phase in which an executable preservation plan is created, based on the well-documented recommendation. The software itself is a J2EE web application relying on open frameworks such as Java Server Faces and AJAX for the presentation layer and Enterprise Java Beans for the backend. It is integrated in an interoperability framework that guarantees loose coupling of services and registries through standard interfaces and provides common services such as user management, security, and a common workspace. Based on this technical foundation, the aim is to create an interactive and highly supportive software environment that advances the insight of preservation planners and enables proactive preservation planning.

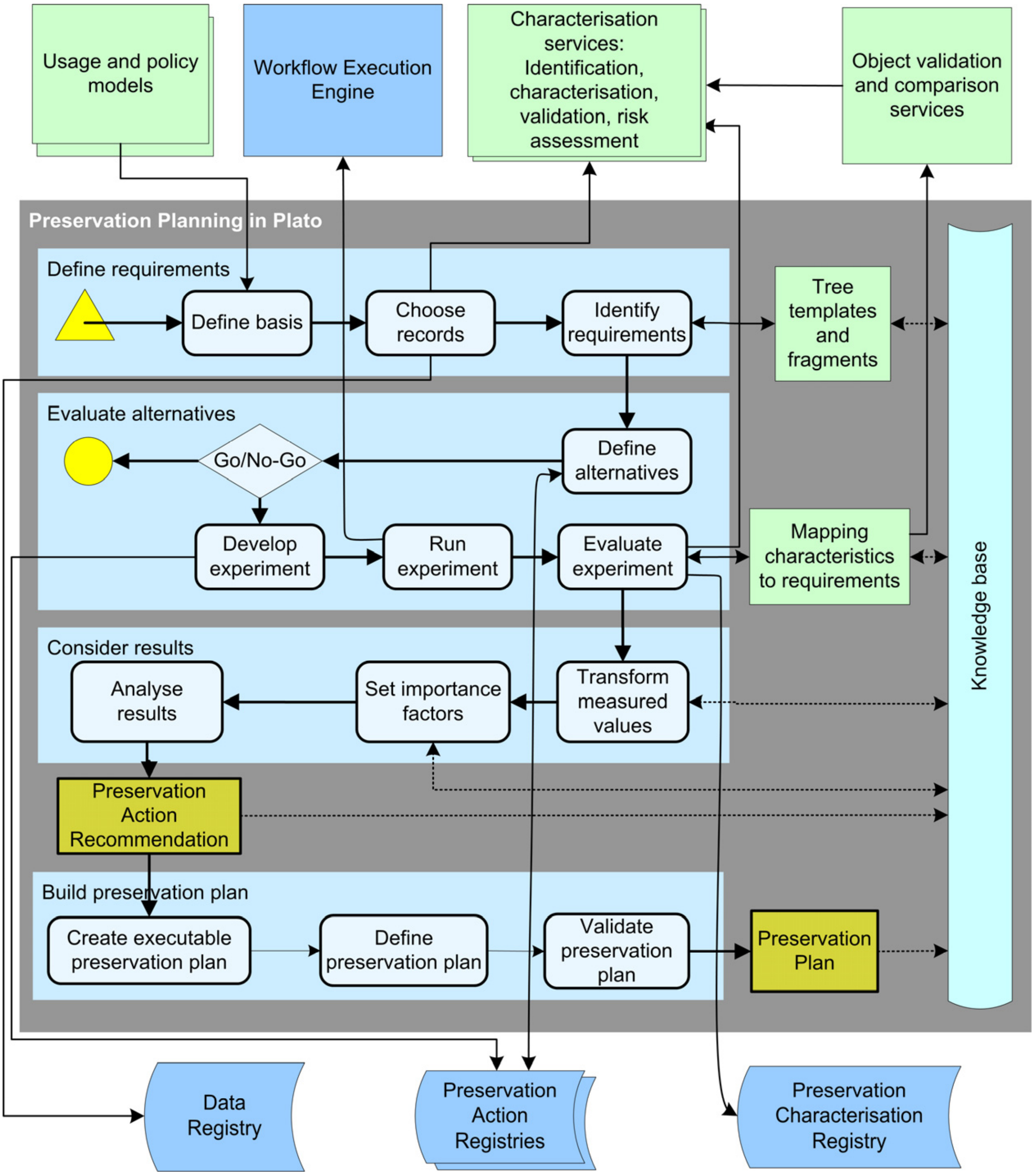
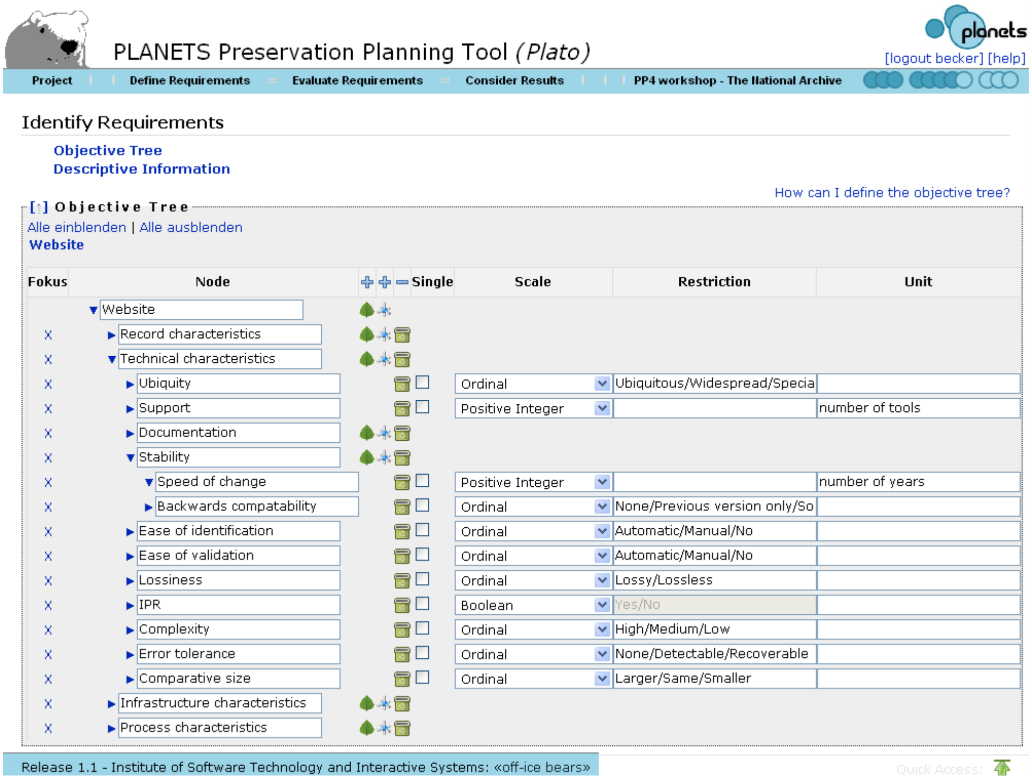

Usage and policy models
Workflow Execution Engine
Characterisation services: Identification, characterisation, validation, risk assessment
Object validation and comparison services
Preservation Planning in Plato
Define requirements
Define basis
Choose records
Identify requirements
Tree templates and fragments
Evaluate alternatives
Go/No-Go
Define alternatives
Develop experiment
Run experiment
Evaluate experiment
Mapping characteristics to requirements
Knowledge base
Consider results
Analyse results
Set importance factors
Transform measured values
Preservation Action Recommendation
Build preservation plan
Create executable preservation plan
Define preservation plan
Validate preservation plan
Preservation Plan
Data Registry
Preservation Action Registries
Preservation Characterisation Registry

Figure 4-14 Preservation Planning workflow



PLANETS Preservation Planning Tool *(Plato)*

[logout becker] [help]

Project | Define Requirements | Evaluate Requirements | Consider Results | PP4 workshop - The National Archive

Identify Requirements

Objective Tree
Descriptive Information

How can I define the objective tree?

[↑] Objective Tree

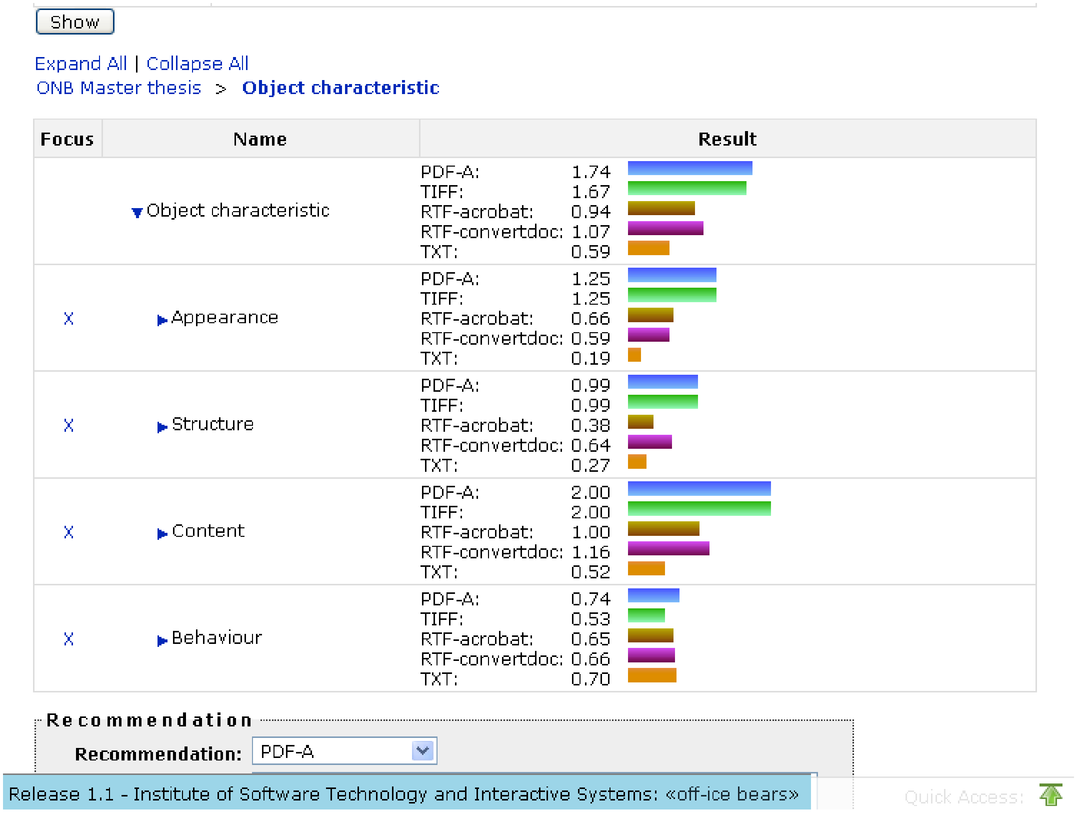
Alle einblenden | Alle ausblenden
Website

Fokus	Node	Single	Scale	Restriction	Unit
	▼ Website				
X	▶ Record characteristics				
X	▼ Technical characteristics				
X	▶ Ubiquity	□	Ordinal	Ubiquitous/Widespread/Specia	
X	▶ Support	□	Positive Integer		number of tools
X	▶ Documentation				
X	▼ Stability				
X	▼ Speed of change	□	Positive Integer		number of years
X	▶ Backwards compatability	□	Ordinal	None/Previous version only/So	
X	▶ Ease of identification	□	Ordinal	Automatic/Manual/No	
X	▶ Ease of validation	□	Ordinal	Automatic/Manual/No	
X	▶ Lossiness	□	Ordinal	Lossy/Lossless	
X	▶ IPR	□	Boolean	Yes/No	
X	▶ Complexity	□	Ordinal	High/Medium/Low	
X	▶ Error tolerance	□	Ordinal	None/Detectable/Recoverable	
X	▶ Comparative size	□	Ordinal	Larger/Same/Smaller	
X	▶ Infrastructure characteristics				
X	▶ Process characteristics				

Release 1.1 - Institute of Software Technology and Interactive Systems: «off-ice bears»

Quick Access:

Figure 4-15 Requirements definition



Show

Expand All | Collapse All
ONB Master thesis > **Object characteristic**

Focus	Name	Result
	▼ Object characteristic	PDF-A: 1.74 TIFF: 1.67 RTF-acrobat: 0.94 RTF-convertdoc: 1.07 TXT: 0.59
X	▶ Appearance	PDF-A: 1.25 TIFF: 1.25 RTF-acrobat: 0.66 RTF-convertdoc: 0.59 TXT: 0.19
X	▶ Structure	PDF-A: 0.99 TIFF: 0.99 RTF-acrobat: 0.38 RTF-convertdoc: 0.64 TXT: 0.27
X	▶ Content	PDF-A: 2.00 TIFF: 2.00 RTF-acrobat: 1.00 RTF-convertdoc: 1.16 TXT: 0.52
X	▶ Behaviour	PDF-A: 0.74 TIFF: 0.53 RTF-acrobat: 0.65 RTF-convertdoc: 0.66 TXT: 0.70

Recommendation

Recommendation: PDF-A

Release 1.1 - Institute of Software Technology and Interactive Systems: «off-ice bears»

Quick Access:

Figure 4-16 Visualisation of results

4.5 Data Management

Data Management provides services and functions for populating, maintaining, and accessing Descriptive Information which identifies and documents archive holdings and administrative data used to manage the archive.

Data management revolves around use of a graph describing data collections and objects. This graph is usually known as a data graph. It is oriented and non-cyclical. The relations associating a node with its descending nodes are inheritance and composition relations. A data set, also known as a terminal collection, thus inherits the characteristics of all the collections above it. The graphs contain either explicit metadata or references to external files or documents.

Data Management functions include:

- catalogues and inventories on what may be retrieved from Archival Storage,
- administering the archive database functions (maintaining schema and view definitions, and referential integrity),
- performing database updates (loading new descriptive information or archive administrative data),
- performing queries on the data management data to generate result sets,
- producing reports from these result sets,
- processing algorithms that may be run on retrieved data,
- consumer access statistics,
- consumer billing,
- event Based Orders,
- security controls,
- OAIS schedules, policies, and procedures.

For BHL-Europe, the catalogue schema is derived from the fields and semantics of the BHL Deduplication Subset (BDS), which conforms to core Dublin Core fields. BDS was designed as a means to facilitate deduplication, to support fast table views on fields most likely to be used, and for indexing. BDS Fields serve as common denominators, their fields types are known, and they can be indexed generically using an Open Source Indexer like Apache Lucene with SOLR Search engine.

4.5.1 Processes

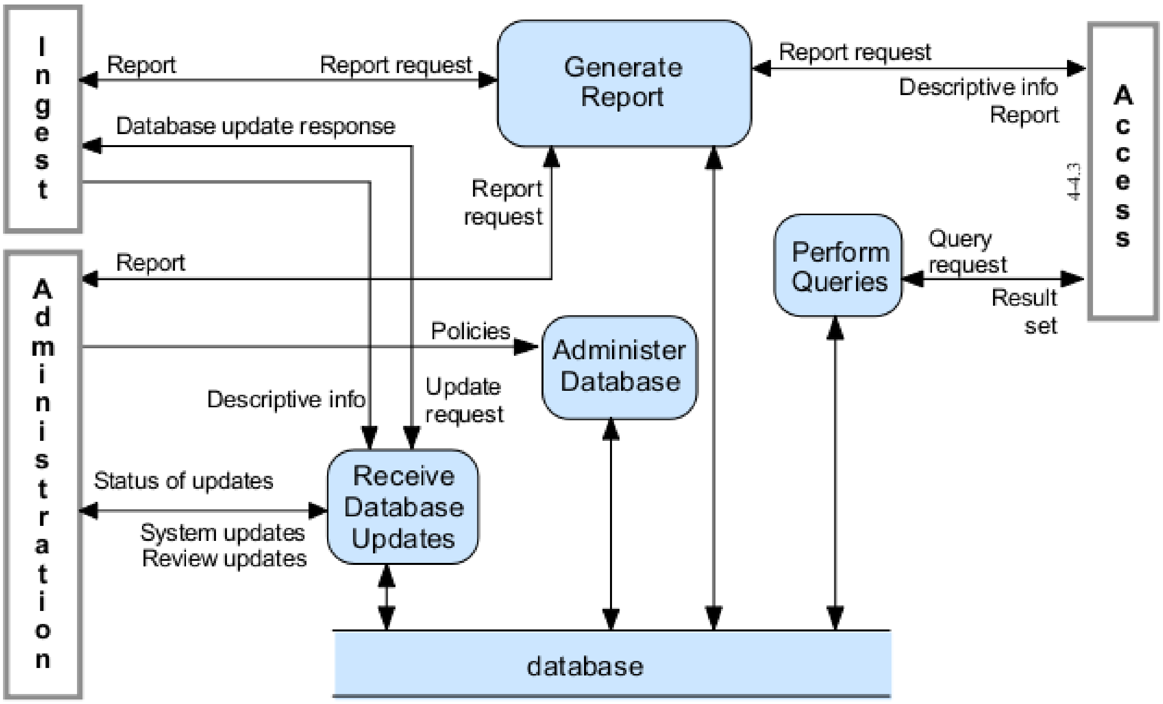


Figure 4-17: Data Management as part of the OAIS model.

For BHL-Europe, the Interfaces are being implemented using the REST architectural style, data is encoded as XML. Different versions of interfaces implementations may co-exist.

The Administer Database function is responsible for maintaining the integrity of the Data Management database, which contains both Descriptive Information and system information. Descriptive Information identifies and describes the archive holdings, and system information is used to support archive operations. The Administer Database function is responsible for creating any schema definitions required to support Data Management functions; for providing the capability to create, maintain and access customized user views of the contents of this storage; and for providing internal validation (e.g., referential integrity) of the contents of the database. The Administer Database function is carried out in accordance with policies received from Administration.

The Perform Queries function receives a SOLR query request from Access and executes the query to generate a result set that is transmitted to the requester.

OAIS distinguishes three categories of Consumer requests:

- Query Requests, which are executed in Data Management and return immediate SOLR result sets for presentation to the user. The result sets always contain record-wise one globally unique ID (GUID) to identify the records and their related archival contents properly. Records in these result sets may also contain any number of BDS fields.
- Report Requests, which may require a number of queries and produce reports for delivery to the Consumer; and orders, which may access either or both Data

Management and Archival Storage to prepare a formal Dissemination Information Package (DIP) for on- or off-line delivery. An order may be an Adhoc Order that is executed only once, or an Event Based Order that will be maintained by the Activate Requests function in Administration, and initiated by a dissemination request that may result in periodic deliveries of requested items.

- The Archival Information Update function in Administration also submits Dissemination Requests to obtain DIPs needed to perform its update functions. Other special request types are allowed, but are not detailed. This function will determine if resources are available to perform a request, assure that the user is authorized to access and receive the requested items, and notify the Consumer that a request has been accepted or rejected (possibly with an estimate of request cost and an option to cancel the request). It will then transfer the request to Data Management or to the Generate DIP function for execution.

The Generate Report function receives a report request from Ingest, Access or Administration and executes any queries or other processes necessary to generate the report that it supplies to the requester. Typical reports might include summaries of archive holdings by category, or usage statistics for accesses to archive holdings. It may also receive a report request from Access and provides descriptive information for a specific AIP. An example for a report is the Deduplication Report. OAIS itself does not cover deduplication but rather assumes that records are deduplicated before ingesting. As for the BHL-Europe archive, deduplication will on the long run become very important for the usability of the system, BHL-Europe must address the issue. One option to handle this is by rejecting ingest if duplicates are found. The Deduplication Report would inform the provider which records have been identified as duplicates.

The Receive Database Updates function adds, modifies or deletes information in the Data Management persistent storage. The main sources of updates are Ingest, which provides Descriptive Information for the new AIPs, and Administration, which provides system updates and review updates. Ingest transactions consist of Descriptive Information which identifies new AIPs stored in the archive. Updating the database always includes that the index (Apache SOLR) is being updated.

System updates include all system-related information (operational statistics, Consumer information, and request status). Review updates are generated by periodic reviewing and updating of information values (e.g., contact names, and addresses).

The Receive Database Updates function provides regular reports to Administration summarizing the status of updates to the database, and also sends a database update response to Ingest.

The Generate AIP function transforms one or more SIPs into one or more AIPs that conform to the archive’s data formatting and documentation standards. This may involve file format conversions, data representation conversions or reorganization of the content information in the SIPs. The Generate AIP function may issue report requests to Data Management to obtain reports of information needed by the Generate AIP function to produce the Descriptive Information that completes the AIP. This function sends SIPs or AIPs for audit to the Audit Submission function in Administration, and receives back an audit report. Fields used for search and retrieval are stored in the Apache Lucene Index.

The Generate Descriptive Information function extracts Descriptive Information from the AIPs and collects Descriptive Information from other sources to provide to Coordinate Updates, and ultimately Data Management. This includes metadata to support searching and retrieving AIPs (e.g., who, what, when, where, why), and could also include special browse products (thumbnails, images) to be used by Finding Aids.

The Coordinate Updates function is responsible for transferring the AIPs to Archival Storage and the Descriptive Information to Data Management. Transfer of the AIP includes a storage request and may represent an electronic, physical, or a virtual (i.e., data stays in place) transfer. After the transfer is completed and verified, Archival Storage returns a storage confirmation indicating (or verifying) the storage identification information for the AIP. The Coordinate Updates function also incorporates the storage identification information into the Descriptive Information for the AIP and transfers it to the Data Management entity along with a database update request. In return, Data Management provides a database update response indicating the status of the update. Data Management updates may take place without a corresponding Archival Storage transfer when the SIP contains Descriptive Information for an AIP already in Archival Storage.

OAIS is a high level model of pervasive computing and describes how constant flows of data can be ingested into an archive. It does not discuss in detail how a system is to be designed that would comply to OAIS, and is not meant as an implementation blueprint. Real world problems like transactional isolation levels and two phase commits in a distributed system that implements an OAIS system are not in the scope of the OAIS specification. However, BHL-Europe must take care that data loss cannot occur at any point during data transfer from the provider library through the ingestion of the data, and during the update database and update indexes process.

4.5.2 Standards

BHL-Europe is using a subset of core Dublin Core (DC) to define the catalogue schema. Dublin Core is a standard defined by the Dublin Core Metadata Initiative. DC covers a set of 15 metadata field which has been standardized as ISO Standard 15836:2009.

Data management follows OAIS recommendations. The use of data packages named DIP, SIP, and AIP for data transmission complies with the OAIS model. The METS file format is used for storing structural information, metadata, page level image and text data. METS is a Standard maintained by the Library of Congress.

The BHL-Europe Archive System implements service endpoints in a way that complies to the architectural style called Representational State Transfer (REST). REST (Representational

State Transfer) was first described in Roy Thomas Fielding: Architectural Styles and the Design of Network-based Software Architectures (Dissertation from 2000), and it is widely accepted as a means to implement HTTP based applications that exchange data. The BHL-Europe service components will use REST with data encoded as XML. XML is a standard specified by the W3C consortium.

4.5.3 Implementation

The following diagram shows the main software modules used to implement Data Management.

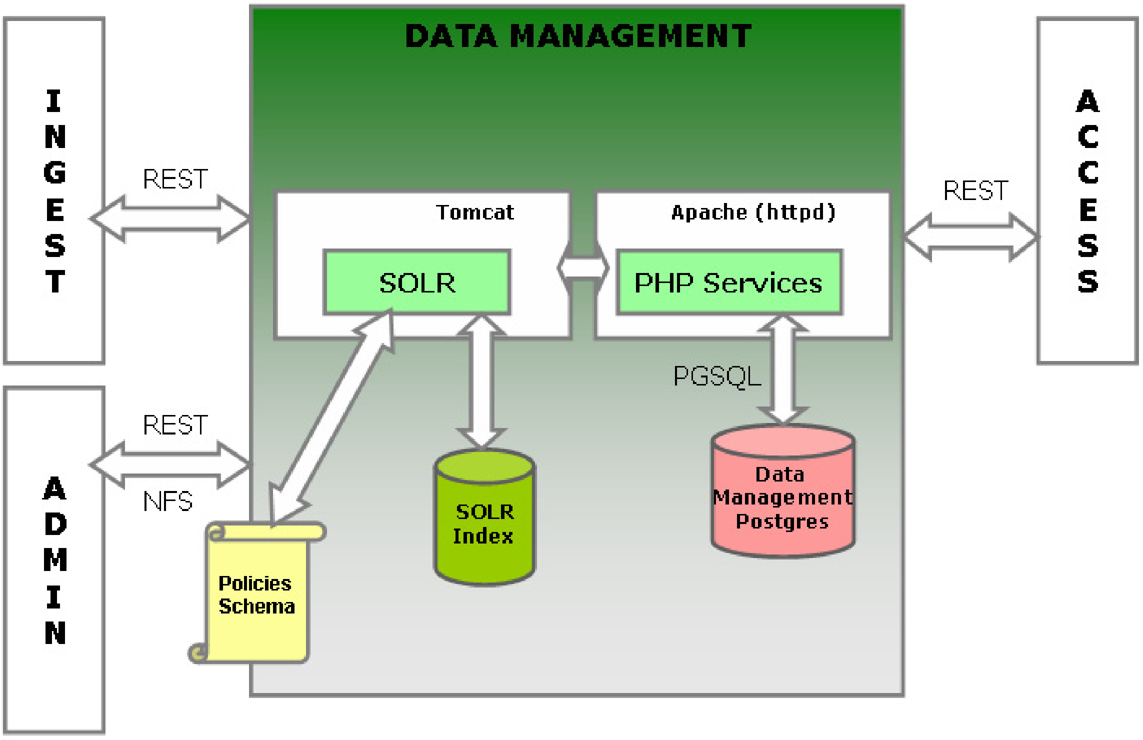


Figure 4-18: Data Management Architecture

4.5.3.1 SOLR Search services

Search services will be operated by SOLR Search Engine which uses an Open Source Indexer called Apache Lucene. Lucene is widely accepted, and has been successfully used in numerous web based applications. The success of Lucene has encouraged tools like Solr that add value to the Lucene framework.

Solr is written in Java and is packaged as a J2EE web archives (WAR) that will be deployed in a J2EE servlet container: Tomcat. Tomcat requires a Java Development Kit (JDK). We will use JDK 6 or higher.

Solr accepts both `HTTP GET` and `HTTP POST` messages for queries and BHL-Europe will use following features:

- A Real Data Schema, with Numeric Types, Dynamic Fields, Unique Keys for performance optimization using external configuration file via XML
- Faceted Search and Filtering results
- Processing algorithms that are run to retrieve data
- Highly Configurable and User Extensible Caching
- Exportable Report via XML file
- Fast Incremental Updates and Index Replication
- XML, CSV/delimited-text, and binary update formats
- REST Approach to pull in data from databases and XML files from local disk and HTTP sources
- Multiple search indices

More about: http://lucene.apache.org/solr/features.html

4.5.3.2 Search Query Language

Data management search query language will be the SOLR query Syntax. Solr supports multiple query syntaxes through its query parser plug-in framework. So that for the Access part interoperation with Data Management is simple.

Features of the Solr query syntax are listed bellow:

- Sorting by any number of fields
- Advanced DisMax query parser for high relevancy results from user-entered queries
- Highlighted context snippets
- Faceted Searching based on unique field values, explicit queries, or date ranges
- Multi-Select Faceting by tagging and selectively excluding filters
- Spelling suggestions for user queries
- More Like This suggestions for given document
- Functions Query - influence the score by user specified complex functions of numeric fields or query relevancy scores.
- Range filter over Function Query results
- Date Math - specify dates relative to "NOW" in queries and updates

- Numeric field statistics such as min, max, average, standard deviation
- Combine queries derived from different syntaxes
- Auto-suggest functionality
- Allow configuration of top results for a query, overriding normal scoring and sorting

More about: http://wiki.apache.org/solr/SolrQuerySyntax

4.5.3.3 PostgreSql

Data management will store data in the standard relational database PostgreSql which will be managed by Php services.

4.5.3.4 Php services

Data Management services to maintain database and search index will be developed in Php. Php services will be running on an Apache Server and they will be custom developed to fit BHL-Europe system requirements. Communication between Php services and Solr Search engine services will be operated via JSON. Services will be responsible for the Search index maintenance and for performing some queries such as:

- Database / Search Index Maintenance
- Loading new descriptive information
- Updating status and statistics of objects usage
- Regenerating Search Index due to a Schema or a Policy update.
- Security control

Php Services will also manage deduplication using BDS fields (BHL Deduplication Subset Fields) based on descriptive information. After ingest a report will be sent to the provider which records have been identified as duplicates and not ingested.

4.5.3.5 Descriptive Information for Object

Regarding to the Dublin Core standards, BHL-Europe will use fields for descriptive Information presented bellow. Those are not exhaustive and may change for optimization reason.

BMS - BHL Metadata Scheme	Obligation	Occurrence	Description
Category	Required	Once	Monograph OR Serial component part
Creator	Required	Repeatable	People or organizations responsible for creation of title.

Date Created	Required	Once	Date digital object was created or scanned.
Date Last Modified	Required	Once	Date digital object was last updated.
Publisher	Required	Once	The person or organization responsible for Publishing the original object. A concatenation of relevant MARC 260 subfields.
Title	Required	Once	The title of the title digitized. Either plain text or a concatenation of relevant MARC 245 subfields.
Date Indicated on Title Page	Required (monographs)	Once	Date indicated on title page
Edition	Required (monographs)	Once	The edition of the book being scanned
Date Created	Required	Once	Date item scanned.
Date Last Modified	Required	Once	Date item modified.
Local Library Identifier	Required	Once	Unique identifier for item scanned, from scanning library catalogue. Note: Use same as local library identifier if this information is not easily accessible
Scanning Contributor	Required	Repeatable	Institution(s) assisting with scanning.
Scanning Institution	Required	Once	Institution responsible for scanning.
Sequence	Required	Once	If a multivolume monograph or serial, the sort order for this item relative to the other objects scanned in this title.
Embedded Rights	Required only if they differ from title rights.	Once	Only required where relating to a specific item.
Start Date	Required - Serials (unless Start Volume present) Optional — monographs	Once	Beginning year of bound object.
Start Volume	Required - Serials (unless Start Date present) Optional - Monographs	Once	Beginning volume number of bound object.
Date Created	Required	Once	Date page scanned.
Date Last Modified	Required	Once	Date page modified.
FileName	Required	Once	File name of associated image file page.
Sequence	Required	Once	Sort order for given page, relative to other pages in item.
Embedded rights	Required only if they differ from title's rights.	Once	Only required where relating to a specific page.
Name	Required	Once	Name of person or organization responsible for creation of title. Last Name, First Name. An alternative is to enter (if relevant) "[Anonymous]" or [S. n.]
Role	Required	Repeatable	One of: Author Publisher Illustrator Added Author
More than one rights holder	Required	Once	Default: 1
Organisation issuing licence	Required	Once	Default: "Rights owner"
Rights owner	Required	Repeatable	
Rights Status	Required	Once	One of "in Copyright", "out of Copyright", "public domain"
End Date Published	Optional (serials)	Once	Year title ended publication.
Start Date Published	Optional (serials)	Once	Year title began publication.
Abbreviation	Optional	Repeatable	Standard abbreviation of the title.
Call Number	Optional	Repeatable	Shelf location of book in scanning library.
Description	Optional	Once	A description of the title.
Foreign Keys (TropicosID, IPNI PubID, TL2, BPH)	Optional	Repeatable	Identifiers for titles as recognized by popular indices.

ISBN	Optional	Repeatable	ISBN(s) for original title, used for monographs.
ISSN	Optional	Repeatable	ISSN(s) for original title, used for serials/journals.
Language	Optional	Repeatable	3 letter language code taken from bits 35-37 in MARC control field 008 or see list at http://www.loc.gov/standards/iso639-2/ISO-639-2 utf-8.txt
Local Library Identifier	Optional	Once	Unique identifier for title in scanning library's catalog. (Allows for later resolution with ILS record)
OCLC Number	Optional	Repeatable	OCLC Number(s) for original title.
Sub-category	Optional	Repeatable	Free text e.g. Atlas, Biography, Encyclopedia, Scientific Journal, popular Journal
Subject	Optional	Repeatable	Subjects assigned to work, as indicated in subfields within MARC 650.
Barcode	Optional	Once	The barcode affixed to the physical object scanned. Note: A library may have more than one copy of an item and this is where the correct copy is identified
End Date	Optional	Once	Ending year of bound object.
End Volume	Optional	Once	Ending volume number of bound object.
Language	Optional	Repeatable	3 letter language code, if different frorn Language of associated Title.
Scanning Sponsor	Optional	Repeatable	Institution Sponsoring scanning.
Issue	Optional	Once	Numeral in which page was published.
IssuePrefix	Optional	Once	Description of subdivision below Volume, if present in item. One of: • Issue • Part • Number • Heft
Page number	Optional	Repeatable	Physical printed page number of scanned page.
Page Type	Optional	Repeatable	A description of the contents of a page. One of: • Appendix • Blank • Cover • Illustration • Index • Map • Table of Contents • Text • Title Page
Volume	Optional	Once	Volume in which page was published.
Year	Optional	Once	Year page was published.
Biography	Optional	Once	Free text biography of creator.
DOB	Optional	Once	Date of Birth
DOD	Optional	Once	Date of Death
Licence terms	Optional	Once	e.g. CC3.0 or CCO CCPD etc

4.6 Access

This component provides the services and functions that support Consumers in determining the existence, description, location and availability of information stored in the OAIS, and allowing Consumers to request and receive information products. Access functions include communicating with Consumers to receive requests, applying controls to limit access to specially protected information, coordinating the execution of requests to successful completion, generating responses (Dissemination Information Packages, result sets, reports) and delivering the responses to Consumers.

4.6.1 Processes

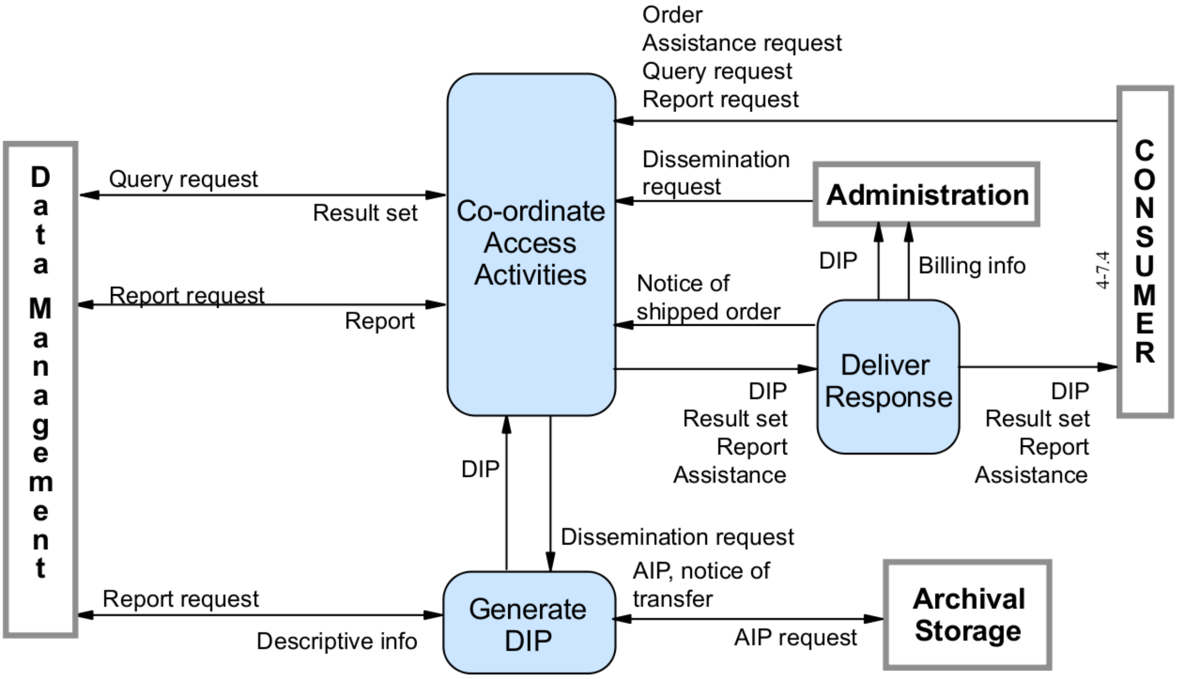


Figure 4-19 Access processes

The **Coordinate Access Activities** function provides a single user interface to the information holdings of the archive. This interface will be implemented as REST and SOAP services and the payload will be encoded in XML and JSON. Three categories of Consumer requests are distinguished:

- query requests, which are executed in Data Management and return immediate result sets for presentation to the user;
- report requests, which may require a number of queries and produce formatted reports for delivery to the Consumer;
- and orders, which may access either or both Data Management and Archival Storage to prepare a formal Dissemination Information Package (DIP) for on- or off-line

delivery. An order may be an Adhoc Order that is executed only once, or an Event Based Order that will be maintained by the Activate Requests function in Administration, and initiated by a dissemination request that may result in periodic deliveries of requested items. The Archival Information Update function in Administration also submits dissemination requests to obtain DIPs needed to perform its update functions.

- Other special request types are listed below.

As naming convention in BHL-Europe, DIP is used for all responses and will be prefixed by its purpose.

This function will determine if resources are available to perform a request, assure that the user is authorized to access and receive the requested items, and notify the Consumer that a request has been accepted or rejected (possibly with an estimate of request cost and an option to cancel the request). It will then transfer the request to Data Management or to the Generate DIP function for execution. This function also provides assistance to OAIS Consumers including providing status of orders and other Consumer support activities in response to an assistance request.

The **Generate DIP** function accepts a dissemination request, retrieves the AIP from Archival Storage, and moves a copy of the data to a staging area and caches these packages for further processing. This function also transmits a report request to Data Management to obtain Descriptive Information needed for the DIP. If special processing is required, the Generate DIP function accesses data objects in staging storage and applies the requested processes. The types of operations, which may be carried out, include statistical functions, sub-sampling in temporal or spatial dimensions, conversions between different data types or output formats, and other specialized processing (e.g., image processing). This function places the completed DIP response in the staging area and notifies the Coordinate Access Activities function that the DIP is ready for delivery.

The **Deliver Response** function handles deliveries of responses (DIPs, result sets, reports and assistance) to Consumers. The on-line delivery accepts a response from Coordinate Access Activities and prepares it for on-line distribution in real time via communication links. It identifies the intended recipient, determines the transmission procedure requested, places the response in the staging area to be transmitted, and supports the on-line transmission of the response.

4.6.2 Standards

Access will provide a range of request types and DIPs for the consumer. As the consumer varies, standards have will be used to supply information to the consumer interfaces. Next to the official standards, community based de facto standards will be provided as well.

4.6.2.1 OAI-PMH [standard]

Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)[9] is a protocol developed by the Open Archive Initiative. It is used to provide metadata description of items which will later be harvested through the simple XML over HTTP interface. The metadata will be presented in Dublin Core and additionally defined formats. For more formats see the DIP's standards. Search functionality provided by this interface reduces to collections, time frames and formats. This interface will be used by Europeana[10] to get all records in their desired format.

4.6.2.2 SRU/SRW [standard]

Search/Retrieve via URL (SRU)[11] is a standard for libraries created within the Z39.50 International Next Generation (ZING). It is a modern equivalent to the Z39.50 protocol based on HTTP and XML. Therefore it is a candidate for cloud services. Search/Retrieve via Web (SRW) additionally provides a SOAP Interface. Both standards use CQL as query language.

4.6.2.3 CQL [standard]

The Contextual Query Language (CQL)[12] is a formal language for representing queries to information retrieval systems such as web indexes, bibliographic catalogues and museum collection information. The design objective is that queries be human readable and writable, and that the language be intuitive while maintaining the expressiveness of more complex languages.

Contextual Queries are built using an Index, a Relation and a Search Term. These elements can be used multiple times connected with Boolean operators and embraced by parentheses. Due to Relation Modifiers a wide range of queries can be built covering proximity searches as well as relevance searches.

The query commands are case insensitive. An example query might look like

```
dc.title any fish or (dc.creator any sanderson and dc.identifier =
"id:1234567")
```

[9] http://www.openarchives.org/pmh/ [30.10.2009]

[10] http://www.europeana.eu/portal/ [30.10.2009]

[11] http://www.loc.gov/standards/sru/specs/search-retrieve.html [30.10.2009]

[12] http://www.loc.gov/standards/sru/specs/cql.html [30.10.2009]

4.6.2.4 RSS/ATOM feeds [standard]

Web feeds like Really Simple Syndication (RSS)[13] and Atom Syndication Format (ATOM)[14] are data formats used to provide information about frequently updated content. For BHL-Europe, new items added or changed within the BHL-Europe network will be placed onto the feed. The feed will be customizable via GET parameters in CQL/OpenSearch or the BHL-Europe Data API and will aggregate the items suitable for the query.

4.6.2.5 OpenSearch [standard]

OpenSearch[15] is a collection of simple formats for the sharing of search results. The OpenSearch formats helps people to discover and use the search engine and to syndicate search results across the web. Search engines that crawl the surface of the web are picking up only a small fraction of the great content that is available within BHL-Europe. OpenSearch provides a deeper insight on the BHL-Europe content for crawlers like Google and Yahoo. OpenSearch was created by Amazon.

4.6.2.6 OpenUrl [standard]

The OpenURL standard (Z39.88)[16] is designed to support mediated linking from information resources, such as abstracting and indexing databases (sources), to library services (targets), such as academic journals, whether online or in printed or other formats. A "link resolver", or "link-server", parses the elements of an OpenURL and provides links to appropriate targets available through a library. A source is generally a bibliographic citation or bibliographic record used to generate an OpenURL. A target is a resource or service that helps satisfy user's information needs. Examples of targets include full-text repositories, online journals, online library catalogues and other Web resources and services. A source is typically a database that indexes various types of information resources often found in libraries, such as articles, books, patents, etc. Some examples of such databases include Web of Science, Scifinder, Modern Languages Association Bibliography and Google Scholar.

[13] http://en.wikipedia.org/wiki/RSS [30.10.2009]

[14] http://en.wikipedia.org/wiki/ATOM [30.10.2009]

[15] http://www.opensearch.org/Home [30.10.2009]

[16] http://www.niso.org/kst/reports/standards?step=2&gid=None&project_key=
d5320409c5160be4697dc046613f71b9a773cd9e [30.10.2009]

4.6.2.7 BHL-Europe Data API [BHL-Europe Development]

This interface will provide search and retrieval functionality especially designed for the BHL-Europe Portal component. It will just serve specific items needed for high performance portal usage without transformation processes at Portal side.

4.6.2.8 Google Booksearch Data API [proprietary]

The Google Booksearch API[17] may be an additional option for Google My Library users. This interface is based on AtomPub and JSON and can process various queries, authentication, and batch requests. Therefore, this interface my be moved to Portal and will be implemented there for better user experience as users are not stored in Access. Base functionality may still be provided by Access.

4.6.2.9 Amazon SOAP [proprietary]

As Amazon is the leading book selling institution, their interface is well known. BHL-Europe will implement the same interface for a broad user community.

4.6.3 Implementation

The Access component coordinates the search and retrieval of information. Figure 4-3 shows the custom made modules within this component.

[17] http://code.google.com/intl/de-DE/apis/books/docs/gdata/developers_guide_protocol.html [30.10.2009]

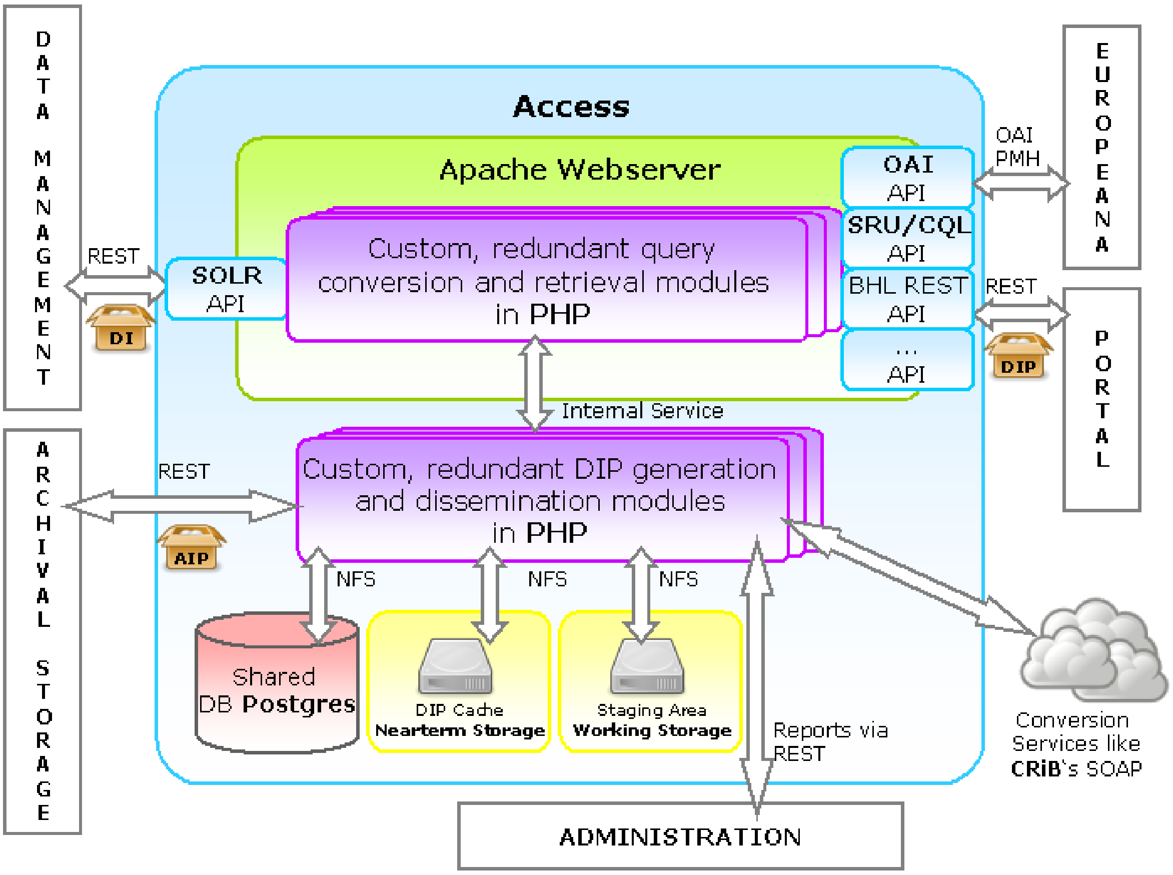

DATA MANAGEMENT
ARCHIVAL STORAGE
Access
Apache Webserver
Custom, redundant query conversion and retrieval modules in PHP
SOLR API
REST
DI
OAI API
SRU/CQL API
BHL REST API
... API
OAI PMH
REST
EUROPEANA
PORTAL
DIP
Internal Service
REST
AIP
Custom, redundant DIP generation and dissemination modules in PHP
NFS
NFS
NFS
Shared DB Postgres
DIP Cache Nearterm Storage
Staging Area Working Storage
Reports via REST
Conversion Services like CRiB's SOAP
ADMINISTRATION

Figure 4-20 Access Architecture

Depending on the request, two sequences are run. Figure 4-21 shows the case of a query. Queries just have to be converted in the appropriate format of DataManagement and the resultlist has to be converted into the format desired by the client. The DI might be changed to make the AIP accessible outside of the OAIS system.

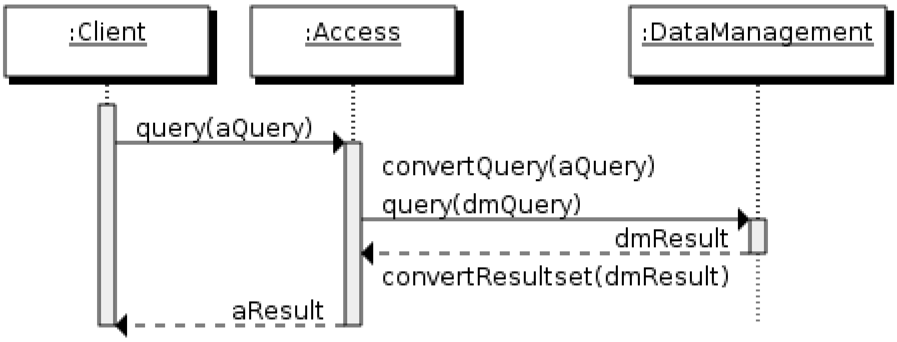


Figure 4-21 A query in access

Figure 4-22 shows the request for a DIP or scan of a page. As ArchivalStorage only provides the best quality raw data, the response must be converted for better performance at the client side. Therefore a cache will be created. If the derived image cannot be created within a few seconds, a Service Temporarily Unavailable HTTP response will be send with a estimated retry date. Then the DIP will be created and stored in Access's cache.

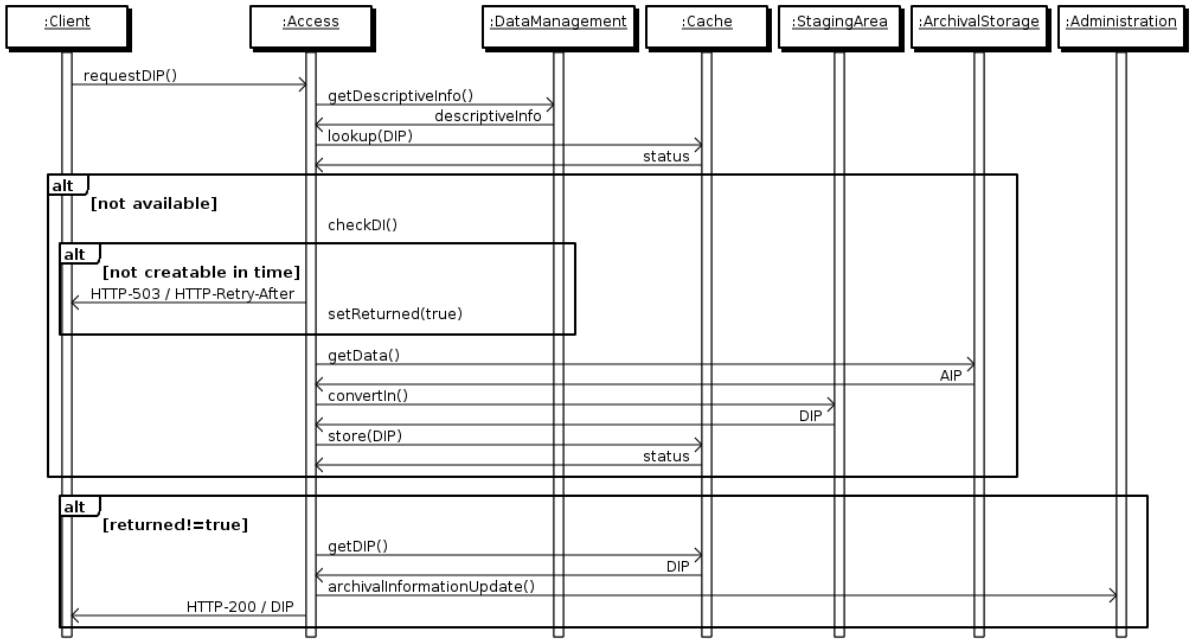


Figure 4-22 A DIP request in Access

The client will always be served from cache. Cache is a nearterm storage while the StagingArea is working storage. When new items are added to BHL-Europe, Administration will request commonly needed DIPs from Access to build up the Cache.

4.7 Administration

This component provides the services and functions for the overall operation of the archive system. Administration functions include soliciting and negotiating submission agreements with Producers, auditing submissions to ensure that they meet archive standards, and maintaining configuration management of system hardware and software. It also provides system engineering functions to monitor and improve archive operations, and to inventory, report on, and migrate/update the contents of the archive. It is also responsible for establishing and maintaining archive standards and policies, providing customer support, and activating stored requests.

4.7.1 Process

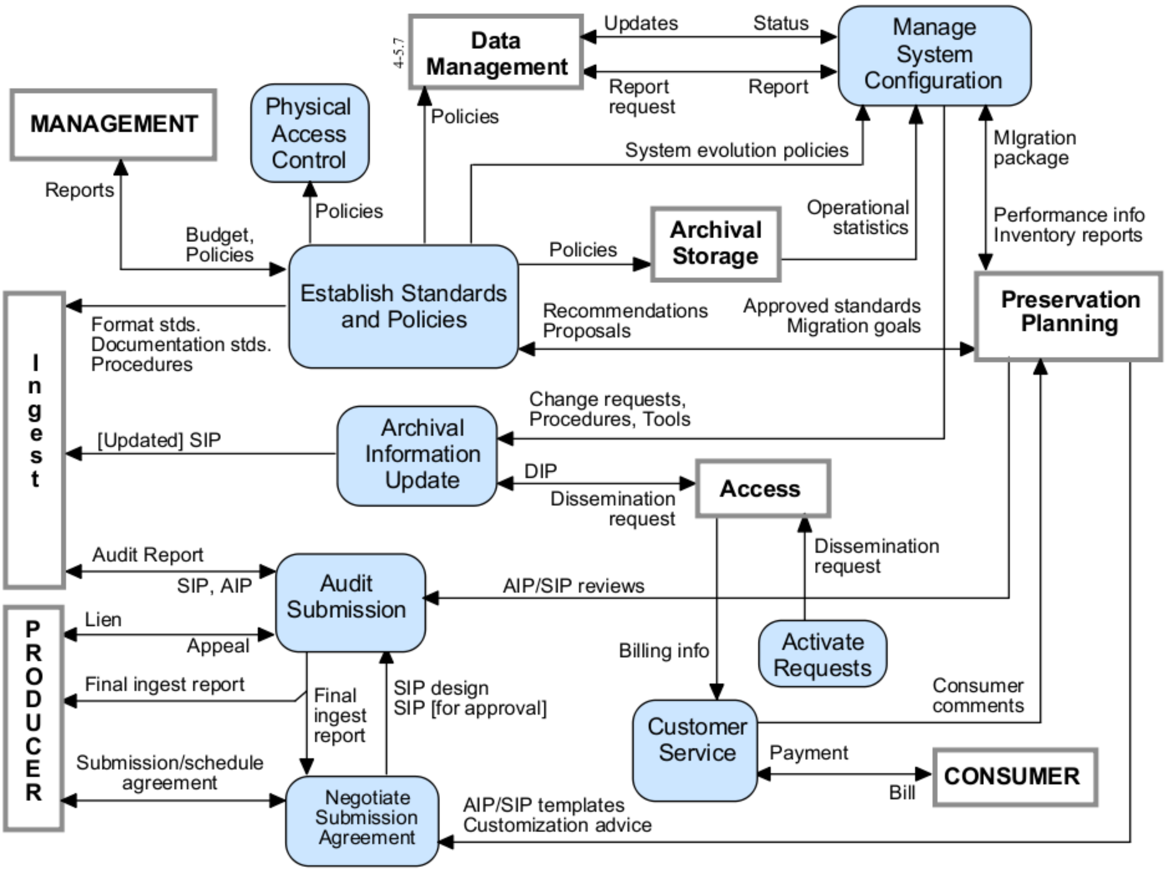


Figure 4-23 Administration processes

The **Negotiate Submission Agreement** function solicits desirable archival information for the OAIS and negotiates Submission Agreements with Producers. This function also negotiates a data submission schedule with the Producer. It maintains a calendar of expected Data Submission Sessions that will be needed to transfer one or more SIPs to the OAIS and the resource requirements to support their ingestion. This function receives AIP/SIP templates and customization advise from Preservation Planning and sends SIP designs and SIPs to the Audit Submission function as part of the submission approval process. The data submission formats and procedures must be clearly documented in the archive’s data submission policies, and the deliverables must be identified by the Producer in the Submission Agreement.

The **Manage System Configuration** function provides system engineering for the archive system to continuously monitor the functionality of the entire archive system and systematically control changes to the configuration. This function maintains integrity and tractability of the configuration during all phases of the system life cycle. It also audits system operations, system performance, and system usage. It sends report requests for system information to Data Management and receives reports; it receives operational statistics from

Archival Storage. It summarizes those reports and periodically provides OAIS performance information and archive holding inventory reports to Preservation Planning. It sends performance information to Establish Standards and Policies. It receives migration packages from Preservation Planning. It receives system evolution policies from the Establish Standards and Procedures function. Based on these inputs it develops and implements plans for system evolution. It sends change requests, procedures and tools to Archive Information Update.

The **Archival Information Update** function provides a mechanism for updating the contents of the archive. It receives change requests, procedures and tools from Manage System Configuration. It provides updates by sending a dissemination request to Access, updating the contents of the resulting DIPs and resubmitting them as SIPs to Ingest.

The **Physical Access Control** function provides mechanisms to restrict or allow physical access (doors, locks, guards) to elements of the archive, as determined by archive policies.

The **Establish Standards and Policies** function is responsible for establishing and maintaining the archive system standards and policies. It receives budget information and policies such as the OAIS charter, scope, resource utilization guidelines, and pricing policies from Management. It provides Management with periodic reports. It receives recommendations for archive system enhancement, and proposals for new archive data standards from Preservation Planning. It also receives performance information and archive holding inventories from Manage System Configuration. Based on these inputs, archive standards and policies are established and sent to other Administration functions and the other Functional Entities for implementation. The standards include format standards, documentation standards and the procedures to be followed during the Ingest process. It provides approved standards and migration goals to Preservation Planning. This function will also develop storage management policies (for the Archival Storage hierarchy), including migration policies to assure that archive storage formats do not become obsolete, and database administration policies. It will develop disaster recovery policies. It will also determine security policies for the contents of the archive, including those affecting Physical Access Control and the application of error control techniques throughout the archive.

The **Audit Submission** function will verify that submissions (SIP or AIP) meet the specifications of the Submission Agreement. This function receives AIP/SIP reviews from Preservation Planning and may also involve an outside committee (e.g., science and technical review). The audit process must verify that the quality of the data meets the requirements of the archive and the review committee. It must verify that there is adequate Representation Information and PDI to ensure that the Content Information is understandable and independently usable to the Designated Community. The formality of the review will vary depending on internal archive policies. The Audit process may determine that some portions of the SIP are not appropriate for inclusion in the archive and must be resubmitted or excluded. An audit report is provided to Ingest. After the audit process is completed, any liens are reported to the Producer, who will then resubmit the SIP to Ingest or appeal the decision to Administration. After the audit is completed, a final ingest report is prepared and provided to the Producer and to Negotiate Submission Agreement. Audit methods potentially include sampling, periodic review, and peer review.

The **Activate Requests** function maintains a record of event-driven requests and periodically compares it to the contents of the archive to determine if all needed data is available. If needed data is available, this function generates a dissemination request that is sent to Access. This function can also generate orders on a periodic basis where the length of the period is defined by the Consumers or on the occurrence of an event (e.g., a database update).

The **Customer Service** function will create, maintain and delete Consumer accounts. It will collect billing information from Access and will send bills and collect payment from Consumers for the utilization of archive system resources. It will respond to general information requests. This function will also collect and respond to feedback on Access services and products. Customer Service will summarize these comments and make them available.

4.7.2 Implementation

We are currently evaluating CRiB, Conversion and Recommendation of Digital Objects Formats (http://crib.dsi.uminho.pt/) to see if it meets part of the functional requirements of the Administration component for BHL-Europe. It is a Service Oriented Architecture (SOA) designed to assist the implementation of migration-based preservation interventions.

All other parts of Administration not implemented by CRiB will be handled by custom PHP Services. The following diagram shows the main software modules used to implement Administration.

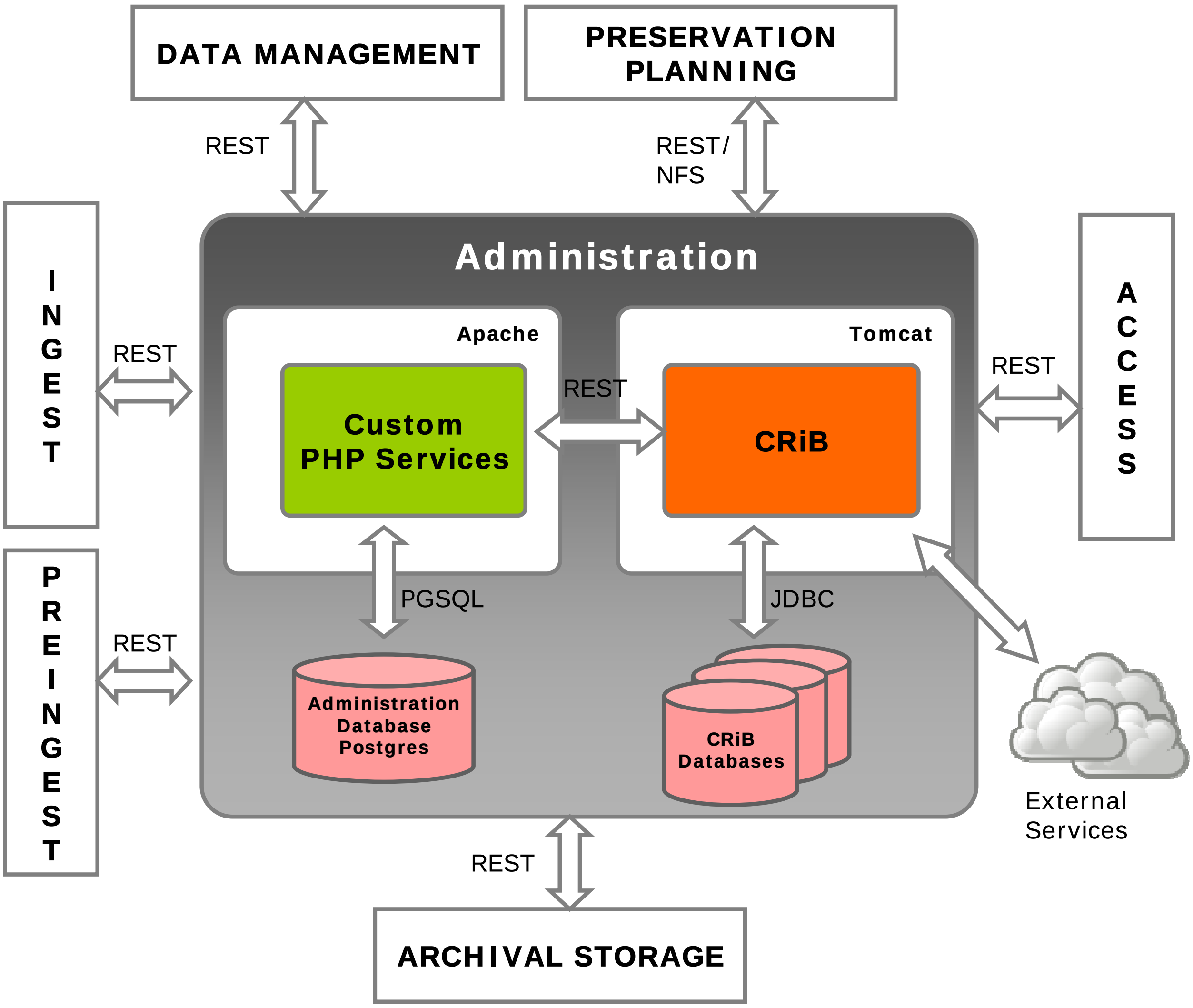


Figure 4-24 Administration Architectre

4.7.2.1 CRiB

CRiB uses web services and would perform the following activities within the Administration component :

- **Format Detection**: a service capable of identifying the underlying encoding of a digital object.
- **Service Registry**: responsible for managing information about existing conversion services. It stores metadata about its producer/developer (e.g. name, description and contact), about the service it self (e.g. name, description, the source/target formats that it is capable of handling, cost of invocation, etc.) and information on how the service should be invoked by a client application (i.e. its access point).

- **Migration Broker**: responsible for carrying out object migrations. In practice, this component is responsible for making sure that composite conversions are performed atomically from the point-of-view of the client application and the rest of the SOA components. Additionally, this component is responsible for recording the performance of each migration service. The results of these measurements are stored in the Evaluations Repository.
- **Format Evaluator**: provides information about the current status of file formats. This information enables the Migration Advisor to determine which formats are better candidates for accommodating the properties of source objects by looking at the characteristics of each pair of formats. This service is supported by a data store containing facts about formats (i.e. Format Knowledge Base), but could also exploit external sources of information such as the PRONOM registry or Google Trends, to automatically determine a format's ubiquity and use trends.
- **Object Evaluator**: in charge of judging the quality of the migration outcome. It accomplishes this by comparing objects submitted to migration with its converted counterparts. Again, these evaluations will be performed according to multiple criteria. These criteria, also known as significant properties, constitute the set of attributes of an object that should be maintained intact during a preservation intervention.
- **Migration Advisor**: produces recommendations of migration alternatives. In reality this component acts as a decision support centre for client institutions that is capable of determining the best possible option within a wide range of alternatives. It accomplishes this by confronting the preservation requirements outlined by client institutions with the accumulated knowledge about the behaviour of each accessible migration path.
- **Generation** of migration reports in appropriate forms for inclusion in the preservation metadata of migrated objects;

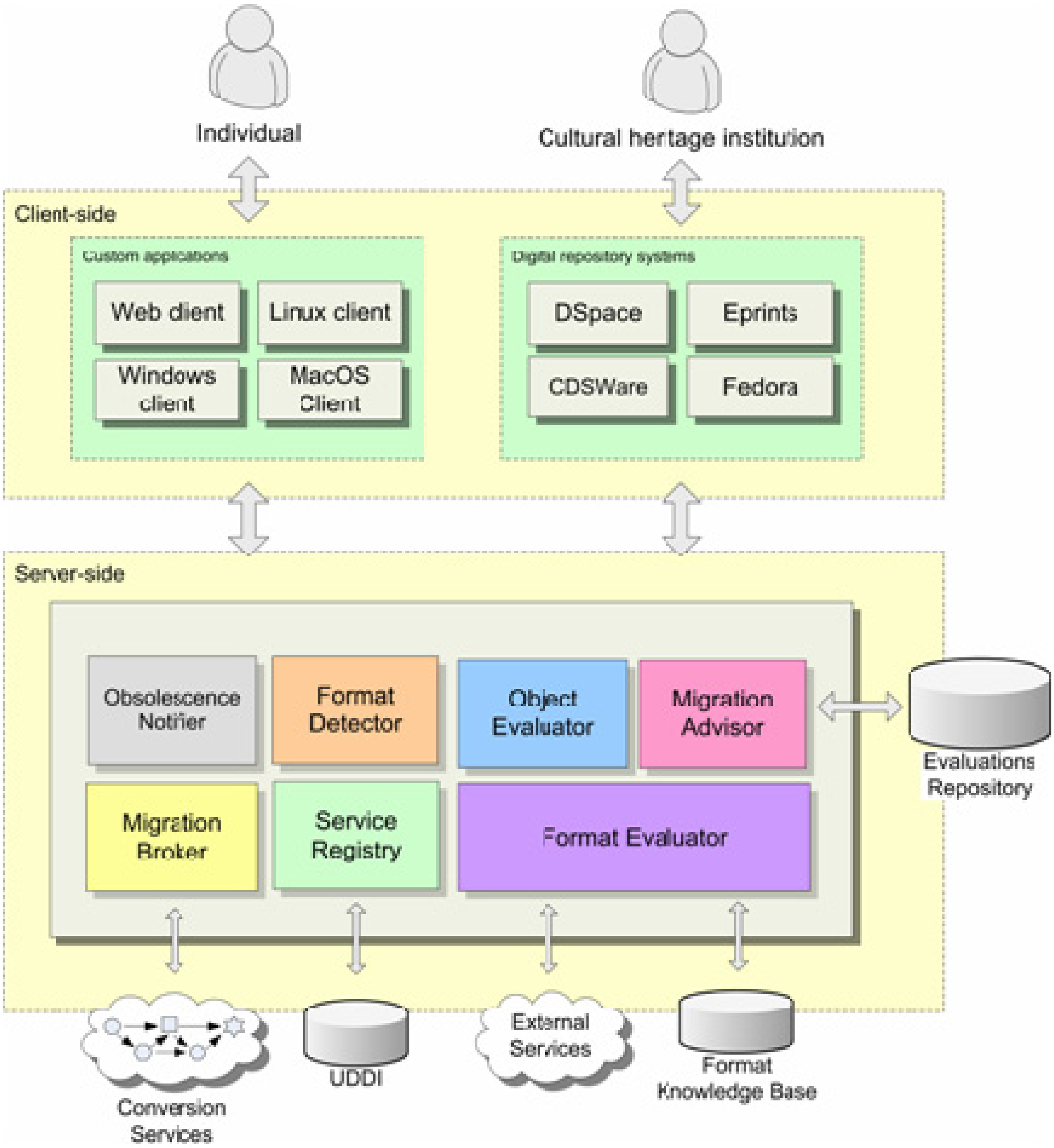


Figure 4-25 Overview of the CRiB architecture

4.8 Portal

This component represents the OAIS consumer. It will help the end user to determine the existence, description, location and availability of items. The portal provides visual aids for browsing the catalogue as well as multilingual search tools to retrieve items on various levels in various languages. In addition to the search and retrieve functionality of DIPs, the portal has to implement surveys. This derives from the OAIS consumer entity. End users will provide service requirements to OAIS preservation planning with the help of these surveys.

4.8.1 Processes

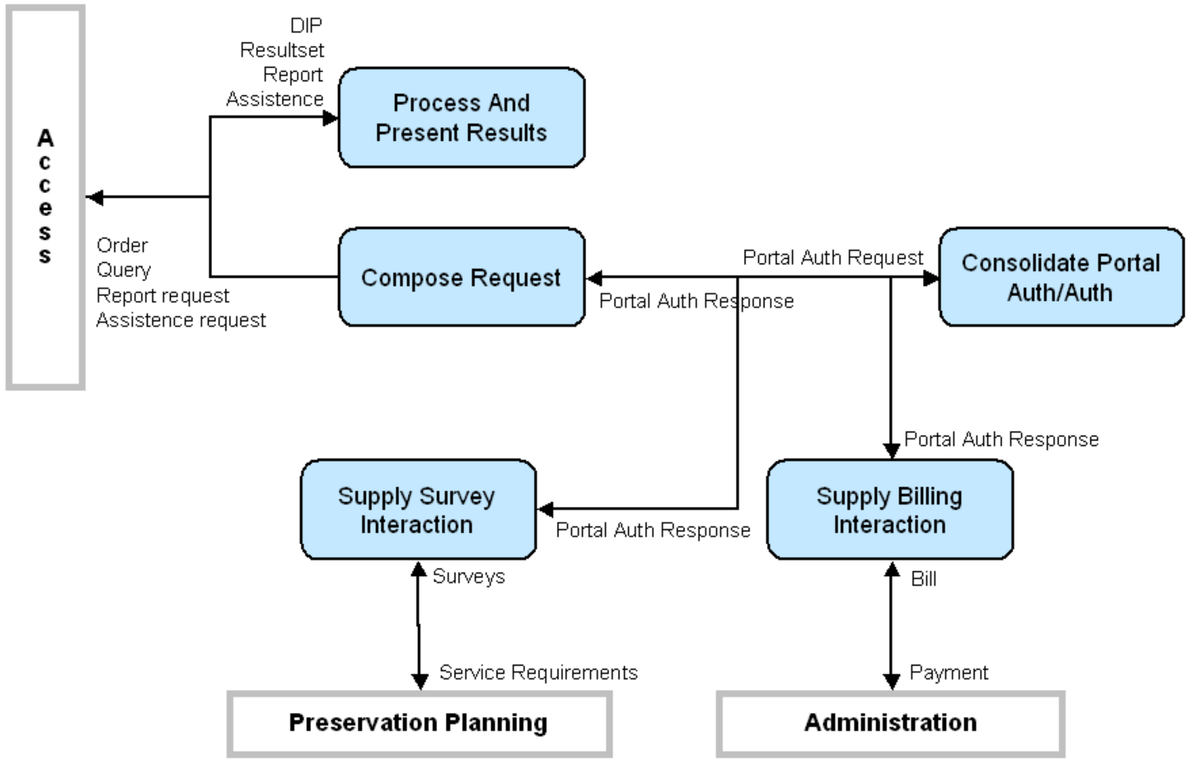


Figure 4-26 Portal processes

The **Compose Requests** function maps user requests to the appropriate Access interface to get the desired Dissemination Information Package (DIP). The portal users will use tools to build their query. These tools will be supplied by this function. Vocabulary services and multilingual search features will enhance the full-text and metadata search. Within the **Process and Present Results** function, various result sets provided by Access will be prepared for further user interaction. The results may be presented in categories and divided by facets. Additional visualizations for available digitized data are provided by this function as well. The prepared result set can be used to compose new requests if desired. To improve the user experience, the **Supply Survey Interaction** is implemented by this component. The end-user can use these functions to provide additional service requirements to the provider and the portal.

The **Supply Billing Interaction** functions lets the end-user purchase items from the OAIS. Within BHL-Europe, most items are within public domain and payment is not relevant.

As an Auth/Auth convenience, the portal will use the **Consolidate Portal Auth/Auth** functions to provide information about security required by the other OAIS components. Not all users registered at the portal need to be known by the other components which reduces administrative overhead.

4.8.2 Standards

4.8.2.1 Vocabulary Representation SKOS [W3C standard]

The Simple Knowledge Organisation Systems (SKOS)[18] is a family of formal languages designed for representation of thesauri, classification schemes, taxonomies, subject-heading systems, or any other type of structured controlled vocabulary. SKOS is built upon RDF and RDFS, and its main objective is to enable easy publication of controlled structured vocabularies for the Semantic Web. SKOS is currently developed within the W3C framework. It is used for export and import of controlled vocabulary within some thesaurus services used by the portal.

4.8.2.2 ISO 2788:1986 Guidelines for the Establishment and Development of Monolingual Thesauri [ISO standard]

An International Organization for Standardization (ISO) standard for the construction of monolingual thesauri. Includes guidelines for dealing with references, definitions, abbreviations, vocabulary control, indexing terms, compound terms, basic relationships, display, and management. Available for sale at www.iso.org

4.8.2.3 ANSI/NISO Z39.19-2005 Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies. [ANSI standard]

This American standard shows how to formulate descriptors, establish relationships among terms, and present the information in print and on a screen. Included are thesaurus maintenance procedures and recommended features for thesaurus management systems. This standard is available online.

4.8.2.4 ISO 5964 Guidelines for the Establishment and Development of Multilingual Thesauri [ISO standard]

An International Organization for Standardization (ISO) standard for the construction of multilingual thesauri. This standard is to be used as an extension of the monolingual guidelines (ISO 2788). Includes guidelines for dealing with degrees of term equivalence and non-equivalence, single-to-multiple term equivalence, etc. Also includes examples of thesaural displays. Print version available for sale.

[18] http://www.w3.org/2004/02/skos/ [30.10.2009]

4.8.3 Implementation

The Portal component acts as the presentation layer of the BHL-Europe OAIS System. The following diagram shows the modules within this component which are based on Drupal.

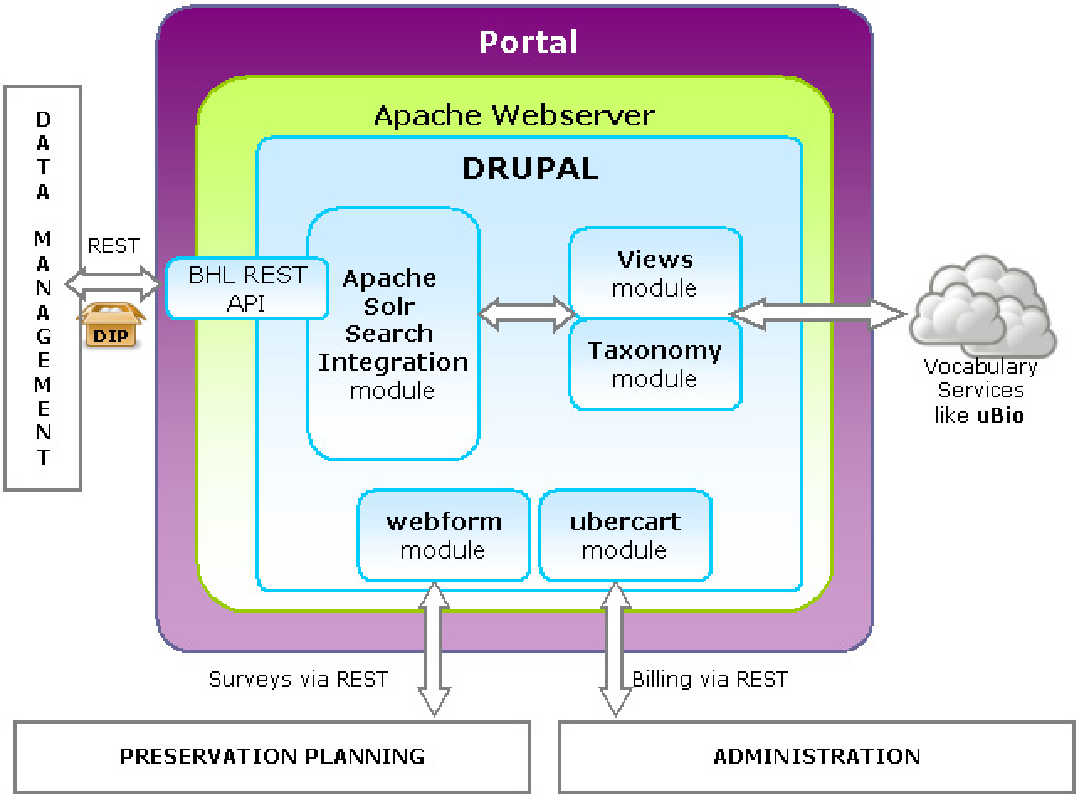


Figure 4-27 Portal Architecture

In most cases, the Portal will be used for searching as the BHL-Europe content is meant to be free. The following figure shows a possible search process within Drupal. The various modules are registered in Drupal for later use. Each module utilizes specific tasks during a request. The SearchModule gathers the information from Access and integrates it into Drupal’s resultlist. The ViewsModule controls the rendering of the result.

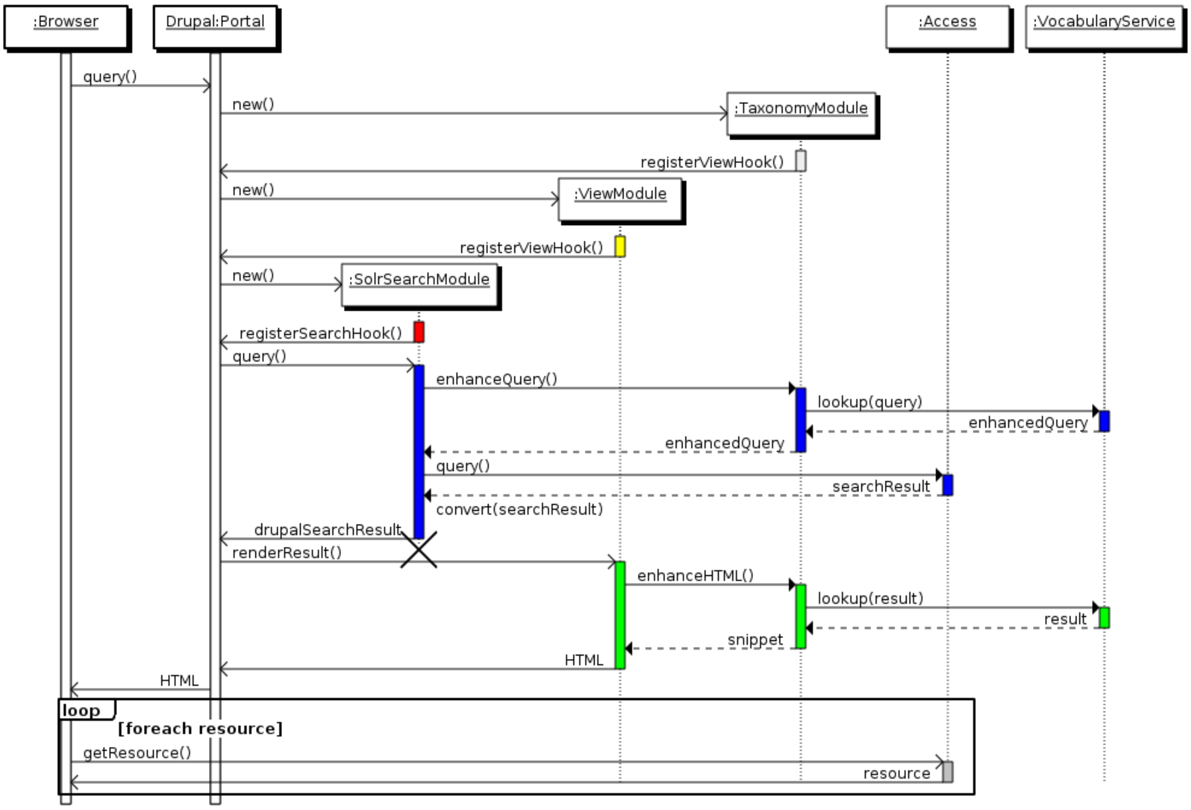


Figure 4-28 Search and retrieval in Drupal

The TaxonomyModule controls the usage of various VocabularyServices and unifies the interfaces. After the request is processed, the browser might have to load additional resources, like images, from Access.

5 External Interoperability

5.1 BHL-Europe / EDIT Interoperability

According to the outcome of the BHL-Europe Fall Meeting in Prague the linkage of BHL-Europe will at first be established with BHL International (BHL-US). Later when the final BHL-Europe infrastructure comes to life this architecture will be adopted and extended where necessary to build the linkage between BHL-Europe and the EDIT Platform.

5.1.1 Background and Vision

EDIT CDM-Stores will expose catalogue data as well as digitized literature. For this, additional services have to be implemented into the EDIT CDM service-layer allowing BHL to harvest reference-repositories belonging to CDM community stores. CDM community stores may also expose literature digitized by EDIT communities to BHL harvesters. All services should conform to a greed BHL-Europe bibliographic data standards and rights management.

Like other taxonomic software, the EDIT Platform uses its own data structures for storing and exposing literature references and digitized literature. Potentially hundreds of taxonomic databases are storing essentially the same information independently. It is obvious that re-using literature information stored in a global repository would be much more efficient and significantly ease the workload of the individual scientist. EDIT will use BHL and BHL-Europe for this purpose in the following ways:

- An extended Common Data Model will reference and cache literature-references and literature content stored by BHL, as well as requests for the scanning of further literature.
- The CDM Java-Library will support these CDM extensions for BHL-requests.
- BHL-functionality will be available in the applications of the Platform for Cybertaxonomy. For example, using BHL-search functions in the EDIT Taxonomic Editor taxonomists will be able to pick literature references from BHL rather than creating them themselves; CDM web portals will display selected BHL content (e.g. protologues); requests for digitization can be submitted while using either software.

5.1.2 BHL access to EDIT

Usually the primary purpose of CDM Stores and Servers is storing taxonomic data. The bibliographic data contained in CDM Stores are either nomenclatural citations or bibliographical references in a more general sense. In both cases it is information on the article and page level which is related to specific taxa and their related taxa like synonyms. Harvesting bibliographic data with the according taxonomic information from CDM stores will give BHL access to bibliographic data on page and article level and also information on the taxonomic topic of the citations which could be used to enrich metadata during the ingest process.

5.1.2.1 OAI-PMH

An OAI-PMH compliant web service will be implemented into the CDM library. This CDM OAI-PMH web service will support the dissemination of records from OAI_PMH items in multiple metadata formats:

- In compliance to the OAI-PMH each item will be exposed in the **DC (Dublin Core)** format.
- Each item will also expose related taxonomic information as **Simple Darwin Core** record (Simple DwC) (http://rs.tdwg.org/dwc/terms/simple/index.htm) Simple DwC allows multiple `SimpleDarwinRecords` in one `SimpleDarwinRecordSet` per file. Thus using Simple DwC it is possible to disseminate all related taxa, and names in one single record, e.g:

```
<?xml version="1.0" encoding="UTF-8"?>

<SimpleDarwinRecordSet

 xmlns="http://rs.tdwg.org/dwc/dwcrecord/"

 xmlns:dc="http://purl.org/dc/terms/"

 xmlns:dwc="http://rs.tdwg.org/dwc/terms/"

 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

 xsi:schemaLocation="http://rs.tdwg.org/dwc/dwcrecord/
http://rs.tdwg.org/dwc/xsd/tdwg_dwc_simple.xsd">

 <SimpleDarwinRecord>

  <dc:type>Taxon</dc:type>

  <dwc:scientificName>Centropyge fisheri (Snyder 1904)</dwc:scientificName>

  <dwc:acceptedNameUsage>Centropyge fisheri (Snyder 1904)</dwc:acceptedNameUsage>

  ...

  </SimpleDarwinRecord>
```

5.1.2.2 Allow user contribution of digitized literature to BHL.

In principle it would be possible to allow users or even ask them to contribute metadata or scanned media to BHL-E. But at the time this document is being written, no decision has been made whether BHL-E will allow crowd sourcing, so an according policy is not jet available. Technical requirements and specifications will have to depend on a policy and thus have to be postponed.

5.1.2.3 Expose literature digitized by EDIT communities to BHL harvesters

Communities in EDIT like for example the EDIT Cichorieae exemplar group, are digitizing literature. The digitized media which is linked from the CDM database and which thus is available from the CDM Stores could be offered to BHL-E. Digitization undertakings like this are not an institutional contribution to BHL-E, but are in fact user contribution and have to be treated like this, therefore technical specifications only can be defined when a policy on the user contribution has been defined.

In principle a CDM Server could expose information on and access to digitized literature via the OAI-PMH. An additional appropriate metadata format would be offered, so that each OAI-PMH item would also expose this kind of data.

5.1.2.4 Linking CDM Entities to the GRIB (Global References Index to Biodiversity)

The GRIB (Global References Index to Biodiversity) will serve as a central index in the BHL-E infrastructure. Each bibliographic record stored in this index will be assigned with a Globally Unique Identifier (GUID). As a consequence of the interoperability between EDIT and BHL bibliographic records derived from the GRIB can be incorporated into CDM Stores. These BHL derived data records must be stored in the CDM together with the according GUID. It is important to preserve this GUID in order to link to the original data since the authority of this data still is the BHL-E. Storing the GUID in the CDM also allows BHL to identify its own records during the harvesting process. Harvesting can be more accurate and efficient if GUIDs are used as identifier for an OAI-PMH item.

The CDM internally creates a GUID for all identifiable entities like references which are not derived from a foreign source together with an existing GUID. BHL may use this GUID internally to link persistently to the original source of the bibliographic data derived from CDM Stores.

5.1.3 EDIT *access* to BHL

5.1.3.1 Prospective and available technology to access BHL-Europe and BHL-US

The following table gives a quick overview:

Standard	BHL-US	BHL-Europe
SRU/SRW	NO	Planned
OpenSearch	NO	Planned
OpenUrl	Beta version: http://www.biodiversitylibrary.org/openurl.	Planned
BHL NameService	http://www.biodiversitylibrary.org/services/name/NameService.asmx	To be determined

5.1.3.1.1 Services planned for BHL-Europe

Prospective data access services for BHL-Europe are described in BHL-Europe D3.4 Chapter 4.6.2.

The SRU/SRW and Opensearch services are intended to be implemented in the access component of the OAIS infrastructure, but from the integration point of view it is desirable to implement these services also into the GRIB (Global References Index to Biodiversity), see also the chapter “Integration of BHL services in user interface elements of EDIT Platform applications” below.

5.1.3.1.1.1 SRU/SRW

See BHL-Europe D3.4 Chapter 4.6.2.2

The SRU/SRW service implemented in BHL-Europe should at least support Dublin Core [http://www.loc.gov/standards/sru/resources/dc-schema.xsd] as record schema.

5.1.3.1.1.2 OpenSearch

See BHL-Europe D3.4 Chapter 4.6.2.5

5.1.3.1.1.3 OpenUrl

See BHL-Europe D3.4 Chapter 4.6.2.6

5.1.3.1.2 Services offered by BHL-US

5.1.3.1.2.1 OpenURL

BHL has released a beta version of its OpenURL Resolver API for testing in September 2009. A full description of the service is available at http://www.biodiversitylibrary.org/openurlhelp.aspx.

Any repository containing citations to biodiversity literature can use this API to determine whether a given book, volume, article, and/or page is available online through BHL. The service supports both OpenURL 0.1 and OpenURL 1.0 query formats, and can return its response in JSON, XML, or HTML format, providing flexibility for data exchange.

By default, the query interface will (if possible) redirect to the Biodiversity Heritage Library page containing the citation described by the query arguments. If more than one possible citation is found, the query interface redirects to a page from which the appropriate citation can be selected.

OpenURL supports wildcard, so it is for example possible to search for all titles containing the word like Dipte*a (title=%Dipte%a). The the OpenURL service implementation of BHL-US returns one citation record per matching page. Requesting only for higher level data records like data only on volumes or titles is not supported.

5.1.3.1.2.2 BHL Name Services

BHL uses TaxonFinder, a taxonomic intelligence tool developed by collaborators at uBio.org, to locate and identify scientific names within the text of digitized books. This names-based index can be searched using the BHL Name Services.

These services are XML-based web services that can be invoked via SOAP or HTTP GET/POST requests. Responses can be received in one of three formats: XML wrapped in a SOAP envelope, XML, or JSON.

Documentation at: http://docs.google.com/Doc?id=dgvjvvkz_1x5qbm3

Interesting in terms of the interoperability between EDIT and BHL are following services: The **NameSearch** service allows searching for names whereas the percent character ‘%’ can be used as wildcard:

http://www.biodiversitylibrary.org/services/name/NameService.ashx?op=NameSearch&name=ze%+mays&format=xml

It returns a result list of matching names together with the NameBankID. Using this identifier the **NameGetDetail** Service can be requested to retrieve a list of publication in which the found name occurred.

5.1.3.2 Additional Services

5.1.3.2.1 Bid-List and Scan-Request Service

In order to successfully establish the interoperability of CDM applications like the CDM DataPortal and the Taxonomic Editor with BHL-Europe these applications will be able to get information from the **Bid-List** and from the **Scan-Request-List**. Both are integral part of the GRIB (Global References Index to Biodiversity). Thus the GRIB should expose the required functionality through web services which preferably are implemented as lightweight REST services.

As a centralized index on references the GRIB will assign a GUID (Globally Unique Identifier) for each reference record whereas a GUID may be a Life Sciences Identifier (LSID), a Digital Object Identifier (DOI®), or a Persistent Uniform Resource Locator (PURL). In web service requests this GUID will be used as the primary key to address a reference as a resource, so this GUID will be the first path element on top of the root of the REST service. Information from the Bid-List or from the Scan-Request-List is subordinate resources of each root resources. The following table gives an overview on the different resources.

Path	HTTP methods	Notes
/{GUID}	GET: Reference Item preferably in Dublin Core format	Reference items may be Concepts of references, Titles or even concrete Items
/{GUID}/bid	GET: Pager or List of bids for the specified resource	Each element is identified by a unique id called the {bid-id}
/{GUID}/bid/{bid-id}	GET: A specific Bid Record identified by the {bid-id}	
/{GUID}/request	GET: Information on existing requests POST: create a new request for this reference item	

In addition to these RESTful web services it is necessary that the GRIB allows find resources for which the GUID is not jet known to the client. This search functionality will be provided by implementing OpenUrl, OpenSearch, or SRU/SRW as described above. Data returned by any of these services will always reference the GUID.

5.1.3.2.2 Crowd Contribution Services

Further interoperability like crowd contribution of scans is not jet covered by this document and will be specified elsewhere, please also refer to chapter “Allow user contribution of digitized literature to BHL”

5.1.3.3 Integration of BHL services in user interface elements of EDIT Platform applications

It is desirable to be able to use BHL metadata to enrich user interfaces of EDIT Platform applications with features like pick lists, autosuggestion, suggesting search, etc. In order to allow using BHL-Europe metadata to enrich user interfaces a fast and suitable search service is needed. OpenSearch or SRU/SRW are suitable service standards for this functionality. It is planned to implement these in to the access component of the BHL-Europe OAIS infrastructure.

However it is questionable if this infrastructure can support that amount of traffic, caused by highly frequent search requests. Therefore it is beneficial to implement OpenSearch or SRU/SRW into the GRIB (Global References Index to Biodiversity).

A further point for the implementation of these services into the GRIB is that this index will not only contain bibliographic data related to literature already scanned, but the de-duplicated union of all collaborating catalogues. So the GRIB is capable to deliver the most complete result sets on search requests.

5.1.3.4 Supporting BHL within the CDM Library

The CDM Library offers service methods to get, find, create, update, and delete CDM entities as well as higher level operations on those. In order to allow using BHL-Europe services from the CDM in an easy way, wrapper services will be implemented into the CDM Library. These wrappers will accept or return CDM entities. The CDM entities are converted internally into the data formats or objects required by BHL-Europe services or vice versa. So from the CDM side all BHL functionality will completely encapsulated and hidden by these wrapper services. This avoids erroneously wrong usage of the BHL-Europe services and thus increases the reliability of the interoperability between both sides.

6 Prototype requirements and implementation plan

This chapter summarizes the implementation plan for the prototype and its requirements until month 18.

The prototype will not provide enhanced security features and will be deployed as VM Ware Image at NHM UK. A test users group has to be created.

Planned releases:

- July 2010: v1.Alpha
- September 2010: v1.Beta release
- October, 31 2010: v1.Final release

Each component has its own requirements and planned features until month 18.

6.1 Pre-Ingest

- 14 Partners will give links to the data for BHL-Europe
- No security for the prototype
- No need of storage/archiving of data in infrastructure for the moment
- Links will be made to each storage areas of partners

6.2 Portal

- Incorporate user survey feedback : D5.8 (~ end of June)
- English and German version (Multi-lingual portal)
- Running in the NHM Servers
- biodiversity.eu domain name
- Connected to data management - and archival storage
- Multi-lingual search
- The design has to be the BHL-Europe one : http://bhl.ait.co.at/

6.3 Ingest

- Survey about taxonomic tools we have to integrate

6.4 Access

- EUROPEANA will access to BHL-Europe components
    - OAI-PMH Provider
- No derivative generation

6.5 Data Management

- SOLR implementation of the Search Engine
- METS Draft as Schema (BHLE METS SIP Profile (1.Draft - September 09))

6.6 Archival Storage

- Using fedora (NHM Servers)
- ~1000 Documents (German or other)

6.7 Preservation planning

- None

6.8 Administration

- None

6.9 GRIB

- None

6.10 GUID Mint

- None
- No Global one, but temporary from AIT, for the prototype

7 Acronyms and Abbreviations

AIP	Archival Information Package
API	Application Programming Interface
APP	Atom Publishing Protocol
BDS	BHL Europe Deduplication Subset (of Metadata)
BHL	Biodiversity Heritage Library
CQL	Contextual Query Language
CRC	Cyclical Redundancy Check
DI	Descriptive Information
DIP	Dissemination Information Package
DISMARC	DIScovering Music ARChives
EOL	Encyclopedia of Life
ESE	Europeana Semantic Elements
ETL	Extract, Transform, Load
Europeana	European Digital Library
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
MARC	MAchine-Readable Cataloging
METS	Metadata Encoding and Transmission Standard
MIME type	Internet Media Type
MODS	Metadata Object Description Scheme
OAI	Open Archives Initiative
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
OAIS	Open Archival Information System
OCR	Optical Character Recognition
ODBC	Open Database Connectivity

PDI	Preservation Description Information
RDF	Resource Description Framework
REST	Representational state transfer
RSS	Really Simple Syndication (RSS) is a lightweight XML format designed for sharing headlines and other Web content.
SIP	Submission Information Package
SOAP	Simple Object Access Protocol providing a simple and lightweight mechanism for exchanging structured and typed information between peers in a decentralised, distributed environment using XML proposed under the W3C.
SQL	Structured Query Language: ISO, ANSI standard user front end to a relational database management system.
SRU	SRU, the Search and Retrieve URL Service, is a companion service to SRW, the Search and Retrieve Web Service. Its primary difference is its access mechanism: SRU is a simple HTTP GET form of the service
SRW	SRW defines a web service for searching databases containing metadata and objects, both text and non-text. The SRW Initiative builds on Z39.50 along with web technologies. Building on Z39.50 semantics enables the creation of gateways to existing Z39.50 systems; web technologies reduce the barriers to new information providers allowing them to make their resources available via a standard search and retrieve service.
SWORD	Simple Webservice Offering Repository Deposit
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WAR	Web archive
WDL	World Digital Library
XML	EXtensible Markup Language
XSLT	XSL Transformation, kurz XSLT, ist eine Programmiersprache zur Transformation von XML-Dokumenten
Z39.50	National Information Standards Organization Z39.50 Information Retrieval Protocol (Z39.50/ISO 23950), a computer protocol that can be implemented on any platform, defines a standard way for two computers to communicate for the purpose of information retrieval. Standard (= ISO

	23950)

8 Figures