Abstract: The EUDAT project is developing a collaborative data infrastructure (CDI) for European research communities. These user communities play a prominent role in defining the CDI. They are actively involved in gathering the requirements for this CDI, and in building and shaping the common data services (which will be deployed and operated at the service provider partner sites). This requirements analysis and development process is the backbone of EUDAT. It is vital as it ensures that EUDAT delivers services which are needed and which will be actively used by the current user communities and beyond.

This document describes the service building process and the current status of the first generation of services that will be offered by EUDAT.
<table>
<thead>
<tr>
<th>Document identifier: EUDAT-DEL-WP5-D5.2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deliverable lead</td>
</tr>
<tr>
<td>Related work package</td>
</tr>
<tr>
<td>Author(s)</td>
</tr>
<tr>
<td>Contributor(s)</td>
</tr>
<tr>
<td>Due date of deliverable</td>
</tr>
<tr>
<td>Actual submission date</td>
</tr>
<tr>
<td>Reviewed by</td>
</tr>
<tr>
<td>Approved by</td>
</tr>
<tr>
<td>Dissemination level</td>
</tr>
<tr>
<td>Website</td>
</tr>
<tr>
<td>Call</td>
</tr>
<tr>
<td>Project number</td>
</tr>
<tr>
<td>Instrument</td>
</tr>
<tr>
<td>Start date of project</td>
</tr>
<tr>
<td>Duration</td>
</tr>
</tbody>
</table>

*Disclaimer:* The content of the document herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the document is believed to be accurate, the author(s) or any other participant in the EUDAT Consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the EUDAT Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the EUDAT Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.
TABLE OF CONTENTS

EXECUTIVE SUMMARY .............................................................................................................6

1 INTRODUCTION .....................................................................................................................8

2 SERVICE BUILDING PROCESS...............................................................................................10
  2.1 Iterative design ..................................................................................................................10
    2.1.1 Service building and integration ...............................................................................10
    2.1.2 Testing and validation ...............................................................................................11
    2.1.3 Productization ...........................................................................................................12
  2.2 Phased delivery ..................................................................................................................12
  2.3 Task forces .......................................................................................................................12
    2.3.1 Task force definition .................................................................................................12
    2.3.2 Roles and responsibilities .......................................................................................13
    2.3.3 Use of resources ......................................................................................................13
    2.3.4 Current task forces ..................................................................................................13
  2.4 Service handover procedure .............................................................................................13

3 EARLY CANDIDATE SERVICES ............................................................................................15
  3.1 Safe Replication ...............................................................................................................15
    3.1.1 Concrete reference architecture ..............................................................................16
    3.1.2 Implementation plan ...............................................................................................17
    3.1.3 iRODS .....................................................................................................................18
    3.1.4 Building the PID micro-service ..............................................................................23
  3.2 Data Staging .....................................................................................................................28
    3.2.1 Implementation plan ...............................................................................................29
    3.2.2 Staging data using third-party services ..................................................................31
    3.2.3 Staging of data using iRODS tools .........................................................................35
  3.3 Persistent Identifier service ..............................................................................................37
    3.3.1 The Handle service .................................................................................................37
    3.3.2 EPIC Handle Service ..............................................................................................39
    3.3.3 Memorandum of understanding between EUDAT and EPIC .................................40
  3.4 Federated AAI ...................................................................................................................40
    3.4.1 Implementation plan ...............................................................................................41
    3.4.2 Architecture Overview ............................................................................................41
    3.4.3 Detailed community requirements evaluations .......................................................44
    3.4.4 Overview of EUDAT services ..................................................................................44
    3.4.5 Pilot services and integration – mid-term view (Phase 2) ...........................................46
    3.4.6 AAI services and integration – long-term view (Phase 2 & 3) .................................47
    3.4.7 Implementation plan ...............................................................................................48
    3.4.8 Technical evaluations ...............................................................................................48
  3.5 Joint Metadata Domain ......................................................................................................49
    3.5.1 Architecture Overview .........................................................................................49
    3.5.2 Implementation Plan .............................................................................................51
  3.6 Simple Store .....................................................................................................................51
    3.6.1 Work plan ...............................................................................................................52
    3.6.2 Status and Initial results .........................................................................................53
    3.6.3 Evaluation process .................................................................................................54
4 CONCLUSIONS ........................................................................................................... 55
4.1 Task forces and the service-building process ............................................................... 55
4.2 Service building status and next steps ........................................................................ 55
  4.2.1 Safe Replication .................................................................................................... 55
  4.2.2 Data Staging ......................................................................................................... 56
  4.2.3 Federated AAI ....................................................................................................... 57
  4.2.4 Joint Metadata Domain ........................................................................................ 57
  4.2.5 Simple Store ......................................................................................................... 57
4.3 Handover and task force termination ............................................................................ 58

ANNEX A. EVALUATION SIMPLE STORE TECHNOLOGIES ........................................... 59
ANNEX B. GLOSSARY ......................................................................................................... 61

LIST OF FIGURES
Figure 1: Technology appraisal and service building process ............................................. 8
Figure 2: Iterative design process ..................................................................................... 10
Figure 3: Service building phases (orange boxes) ............................................................... 11
Figure 4: Phased service delivery .................................................................................... 12
Figure 5: Concrete derived architecture for the safe replication service ......................... 16
Figure 6: The safe replication service key functionality is the inter-working of iRODS and EPIC .................................................................................................................. 16
Figure 7: Basic overview iRODS .................................................................................... 18
Figure 8: Mass Storage System universal driver to dCache ............................................. 21
Figure 9: iRODS Rules Flow diagram ............................................................................ 22
Figure 10: Example rule to safe copy data ...................................................................... 22
Figure 11: PID management scenario ............................................................................. 24
Figure 12: Example Python code to create a handle in the EPIC handle service ............ 27
Figure 13: Example code for creating an iRODS rule .................................................... 27
Figure 14: Dynamic replication utilization scenario steps ................................................. 29
Figure 15: Example of a Griffin configuration file to access iRODS resources ............... 32
Figure 16: Accessing the service using any GridFTP client (third-party model) ............. 33
Figure 17: Accessing the service using Globus On-line (third-party model) .................. 33
Figure 18: Data stager script utilisation ......................................................................... 34
Figure 19: Globus On-line web interface ...................................................................... 35
Figure 20: XSEDE File Manager Interface .................................................................... 35
Figure 21: Staging data using iRODS icommands ............................................................. 36
Figure 22: iRODS icommands utilisation example ......................................................... 36
Figure 23: Life cycle of referable data .......................................................................... 37
Figure 24: Global overview Handle service .................................................................... 38
Figure 25: Overview EPIC Local Handle Service ............................................................ 39
Figure 26: EPIC REST web based service ..................................................................... 40
Figure 27: AAI Architecture ......................................................................................... 42
Figure 28: Global overview metadata architecture ........................................................ 50
Figure 29: The long tail of small data ............................................................................ 53
LIST OF TABLES

Table 1: Community Service Matrix ................................................................. 15
Table 2: Overview of the three different islands that implement the “Safe Replication” service .......... 17
Table 3: Overview of the planned time-frame of the implementation of the “Safe Replication” service . 17
Table 4: PID record description ......................................................................... 24
Table 5: Data staging pilot islands ....................................................................... 30
Table 6: Overview of the planned time-frame of the implementation of the “Data Staging” service ...... 31
Table 7: Overview provided prefixes for testing purposes ..................................... 40
Table 8: Overview AAI requirements ................................................................... 44
Table 9: Timeline Branch A: the harvester and catalogue ..................................... 52
Table 10: Timeline Branch B: Assess metadata schemas and their semantics .......... 52
Table 11: Timeline Branch C: Technology appraisal ............................................. 52
EXECUTIVE SUMMARY

The EUDAT project aims to develop a pan-European collaborative data infrastructure (CDI) for research. The success of the project relies on the user communities actively participating in the process of defining the requirements for the CDI. It is also vital that the research communities are involved in building and shaping the common data services (which will be deployed and operated at the service provider partner sites). This requirements analysis and development process (in collaboration with the communities) is the backbone of EUDAT – it ensures that EUDAT delivers services which are needed and actively used by the current core of user communities and that will benefit additional research communities beyond.

The assessment of the data landscape and the needs of the communities were conducted during first 6 months of the project. The result of these investigations showed that there were five common services that were needed. These are referred to as: Safe Replication, Data Staging, Joint Metadata Domain, Simple Store and Persistent Identifiers. The Safe Replication, Data Staging and Persistent Identifiers are described in detail in this document. At the time of writing the Joint Metadata Domain and Simple Store services are in the preparation (or technology appraisal) phase, therefore only an update on the current status and on the next steps to be taken in their development.

It is important to note that, although authentication and authorization (AAI) is not regarded as a separate service, all the communities that were interviewed mentioned specific AAI requirements. Therefore AAI is essential for the common data services and constitutes an integral part of the CDI. Consequently an overview of the work on AAI is included in this document.

Having identified the common services that were needed, a technology appraisal was undertaken for the different services. These appraisal processes resulted in recommendations for the technologies appropriate for building the EUDAT data services.

To build the required services, EUDAT is using an incremental and iterative approach to service development (generically known as Rapid Application Development). In the EUDAT context, the iterative service deployment process is divided into “service building and integration”, “testing and validation” and “productization”. The plans for implementing the required services reflect these phases of the service building process. In order to organize all the work related to building and deploying the services, the necessary tasks have been divided between different task forces and work packages. During the first year of the EUDAT project, five task forces have been set up to design the five common services and the AAI. Note that there are only five task forces (rather than six) because the Persistent Identifiers (PID) service is included of the Safe Replication task force.

By collaborating closely with technical experts from the core research communities (who are fully aware of the existing community data services, AAI and policies), EUDAT has ensured that the data services provided by the project will fulfil the needs of the communities to a large extent and that they will integrate with the community infrastructures. Thus the EUDAT data services are guaranteed to be of benefit to the communities.

“Safe Replication of Data” is a service that enables communities to easily create replicas of their datasets in multiple data centres. The Safe Replication service will be based on the policy engine provided by iRODS and on the use of micro-services. Three community islands (e.g., CLARIN, EPOS and ENES) were created across seven partner sites (MPI-PL, INGV, DKRZ, FZJ, SARA, RZG, and CINECA) to set up pilot iRODS instances to test cross-site replication. One of the core functions of the Safe Replication service is the automatic registration of persistent identifiers – which includes keeping track of replicas. To implement this function, a PID micro-service has been developed. Part of the development process
for this has involved a discussion on replication policies. Basic tests on cross-site replication and PID registration, including replicas, have been conducted and were successful.

“Dynamic Data Replication” (or Data Staging) is a service that makes it possible to move datasets close to HPC facilities, and then ingest the results of the analysis back into the EUDAT infrastructure. Like the Safe Replication task force, the Data Staging task force has been working in community islands. The actual data staging tool needs to work “on top” of the data service (e.g., iRODS) that is being built by the Safe Replication service task force. The main transfer protocol that is being used for data staging is GridFTP. For all of this to work, iRODS needed to be extended with a GridFTP server (such as Griffin). Different technologies (such as GridFTP, Globus On-Line, XSEDE and basic iRODS) are being evaluated within the community islands and are successfully demonstrated.

Federated AAI is an enabling service which makes it possible access to EUDAT services while upholding user identities, access rights, trust and privacy. The AAI task force has developed an implementation plan consisting of three phases: short-, mid- and long-term. The short-term phase has been completed. It involved enabling Single Sign-on (SSO) to the project members for the use of the project collaborative tools provided by three partners across three countries. The tools included the EUDAT wiki, the JIRA ticket system, and the EUDAT central monitoring and Trouble Ticket System (TTS). The mid-term phase of the plan has mainly involved collecting detailed information about the requirements of the communities (which included discussing topics such as the overall approach to be taken and the use of credential conversion technologies), along with evaluating and piloting different technologies and assessing the community portals. The long term goal is to provide a full and scalable federated identity management – integrated with core EUDAT services, and with a framework for extending to other services.

Federated AAI is a topic that is currently discussed a great deal in the scientific domain. It is considered to be a very complex area, not only from the technical point of view, but also from the perspective of the different levels of policies, for example, attributes release. EUDAT is actively engaged with other initiatives in the area of federated identity management. It is important for EUDAT to continue this participation, both in Europe and globally.

Overall, the task forces have made good progress in building the first candidate services. The first services are expected to be handed over to the operations team in the upcoming months so they can be used by the communities in a pre-production mode.
1 INTRODUCTION

EUDAT is a project where the user communities play a prominent role in defining the collaborative data infrastructure that is being developed. The research communities are actively involved in the requirements gathering process, and in building and shaping the common data services which will be deployed and operated at the service provider partner sites. This collaborative process is the backbone of EUDAT. It is important because it makes sure that EUDAT delivers services that really are needed and that will be actively used by the user communities. This development process is illustrated in Figure 1, and will result in the deployment of the EUDAT CDI envisioned in the Description of Work.

![Figure 1: Technology appraisal and service building process](image)

The overall purpose of the EUDAT services and technology appraisal work package is to identify and utilize available technologies and tools for implementing the required data services, and to identify technology gaps that should be addressed by the joint research activities within EUDAT. The assessment of the data landscape and the communities’ needs has been conducted and the initial results were reported in D4.1.1\(^1\). That report discusses the five common data services that were found to be necessary for all the communities, and gave further details in so-called “researchers’ use cases”. This information was used as input to the technology appraisal process – the results of which are reported in D5.1.1\(^2\). The results of the technology appraisal process (see Figure 1) were recommendations for technologies appropriate for building the EUDAT services.

From the initial assessment, the following common services were identified as necessary: services to easily create and keep track of multiple copies of datasets across multiple data centres so as to improve preservation and persistency (“safe replication”), staging tools to easily move data to and from HPC computing facilities (“data staging”), a joint metadata domain to facilitate and stimulate cross-disciplinary research, and a “simple store” service that makes it possible to store and share the vast

---

\(^1\) D4.1.1 Management Landscape Characteristics and Community Requirements
\(^2\) D5.1.1 Technology appraisal report
quantity of “small” data that exists. Here, “small” data refers to data that is not part of official data sets or collections, but that is as important for the advancement of research as “official” data.

This document describes the service-building process and the current status for the first generation of services that EUDAT is working on: “safe replication” including persistent identifiers, and “data staging”. The “joint metadata domain” and “simple store” services are currently under preparation and hence this document just contains a brief overview of the next steps that will be taken in their development. AAI is also described in this document. Although AAI is not a separate use case, all the communities that were interviewed had specific AAI requirements. AAI is therefore essential for the common services and constitutes an integral part of the CDI.

Chapter 2 discusses the building process, the use of task forces, and the handover procedure for taking services into production. Chapter 3 gives a detailed description of the first generation EUDAT candidate services, and Chapter 4 presents overall conclusions and time plans.
2 SERVICE BUILDING PROCESS

2.1 Iterative design

EUDAT uses an incremental and iterative approach for service development (generically known as Rapid Application Development). The iterative approach in general consists of requirements gathering, design, implementation and re-assessment of a service, as depicted in Figure 2. The user communities are actively involved during all these processes from the definition of requirements to the evaluation. This builds a sense of mutual trust, as well as developing a shared (technical) vocabulary and understanding of the relevant constraints and opportunities. This leads to the technology and service providers having a better understanding of the users’ (or stakeholders’) requirements. It ensures that EUDAT will provide services that fulfil the needs of the research communities and that will actually be used by the users from the different communities.

![Figure 2: Iterative design process](image)

The “design”, “implementation” and “continuous evaluation” steps of the iterative design process are translated to “service building and integration”, “testing and validation” and “productization” in the service building process as outlined in Figure 3.

2.1.1 Service building and integration

The integration sub-task is the initial step of the process. First the specific candidate service is designed according to the input received from the communities, then the different software tools, and the implementations of different technologies are coupled together according to the design. Often the integration is not straightforward, but requires some changes in the interfaces or the development of a small extension. Therefore the final results usually need to be tested and evaluated to ensure that the various components work harmoniously. Typically the changes

- can be related to the adoption of standard interfaces and protocols (and more generally to the implementation of functional requirements), or
- can be intended to improve the software quality, or satisfy non-functional requirements such as robustness, redundancy, performance, and user-friendliness.
For example, the Safe Data Replication service is based on iRODS and integrates the PID (Persistent Identifier) management functionality of the EPIC server (see section 4.2, D5.1.1). In order to make these work together, it has been necessary to extend iRODS (see the following chapters for details) to support the needed functionalities.

The initial candidates are considered to be core services. This means that these services are generic enough for the EUDAT consortium to offer them to all the communities. Consequently community services, which are more specialized, have not been involved in the integration so far. However, specific instances of the core services interact with specific communities, and hence different configurations can be defined – based on the needs of each individual community. The definition of such configurations is part of the integration sub-task too. If it is possible to identify a common approach underlying the various configurations, it is shared among the EUDAT partners as a best practice, which is later included in the documentation associated with the particular service.

2.1.2 Testing and validation

After the completion of the integration step, the results are tested and validated. The tests are performed using an incremental approach: starting from the basic functional requirements and progressing to the more complex non-functional ones. The respective test beds are extended progressively too. Some of the tests involve direct human interactions; therefore they take a significant amount of time. In fact these are integration tests (as explained in the previous paragraph) and the complexity of an infrastructure with distributed components requires careful supervision. Another important outcome of the test phase is a good estimation of the resource consumption of the tested service. This information is then provided as input for the operational team. The service’s abilities to recover from hardware failures are also evaluated during the testing activity.
Once the tests are successful, the tested service is validated by the EUDAT core communities. Obviously the whole test process does not proceed in a completely linear manner as in the previous description. There is an ongoing exchange of information between the core communities and the service building task force. For example, if an important vulnerability is discovered during the tests by a pilot user, that is reported back to the group working on the integration part. They will evaluate whether it is possible to create a patch, or whether it is necessary to update the design by changing one or more components. In this way, the evaluation runs in parallel with the other steps of the process. The final evaluation determines whether to promote the candidate service to a fully supported EUDAT service. However if the evaluation is positive, a further step is needed before handing the service over to the operational team, namely the creation of a complete product, as explained in the following section.

2.1.3 Productization
After testing and successful evaluation, each of the candidate services must be transferred so as to become EUDAT services. This productization process is responsible for the release management, along with the documentation for the installation and configuration of the services. A properly-functioning productization process ensures a smooth roll-out of EUDAT services to operations. This process is covered in the handover procedure which is described from the perspective of the service building process in section 2.4.

2.2 Phased delivery
EUDAT is a three year project that is broken down into three separate year-long phases as shown in Figure 4. The diagram shows the specific developmental goal for each phase of the service activity, as well the overall goal to be achieved by the end of the project. The phased delivery fits perfectly into the iterative design cycle.

2.3 Task forces
Within EUDAT, the concept of taskforces has been adopted, mainly for organizational purposes and so as to maintain focus in the process of building services. The following sections give the definition of an EUDAT task force, and describe the roles and responsibilities associated with such task forces, and the use of resources from work packages.

2.3.1 Task force definition
An EUDAT project task force (TF) is a collaborative activity involving more than one work package (WP). As such, task forces are different from WP tasks – which are internal to WPs – even though WP tasks might require and build on input and output from other WPs. A project task force is a vehicle for making progress in a concrete field of work that requires continued collaboration across work packages. The
work of the task forces is monitored by the Services and Architecture Forum (SAF). Task forces can be initiated by the Executive Board (EB) alone, or upon a recommendation from the SAF.

In addition to the elements described above, a task force can be defined by the following characteristics:

- It is used to drive the process of building and deployment in a specific service domain with known building blocks
- It has a short time horizon (usually less than 12 months).
- Its scope and objectives are documented by an implementation plan

2.3.2 Roles and responsibilities

The formal “owner” of a TF (that is, the person primarily responsible for the work done within the TF) shall be a WP leader. Typically this will be the person who can bring the most relevant resources, or the person who is primarily concerned with the work to be achieved, according to the Description of Work (DoW). TF “owners” are agreed upon by the SAF.

TF “owners” can appoint a TF “chair” whose role is to steer and organize the TF work. The chair should be an expert in the field of the TF, and also have some leadership experience. The chair proposes the work plan for the TF, and conducts the TF meetings, unless decided otherwise.

The TF owner and the TF chair are responsible for selecting the TF participants. Any discussions and decisions about the use of resources should involve the other WP leaders affected by the TF work and the PMO. This is to ensure that the requested resources match the resources allocated to the partners in the project, and to make sure that a balance of effort is maintained between the partners in the project.

2.3.3 Use of resources

People who participate in a TF should report their work as part of the work they are doing either in the WP that is responsible for the TF or in another WP that is involved in the TF activity. If a task force participant does not have Person Months (PM) in any of the WP involved with the TF in question (or if the person has too little compared to what they are actually supposed to do for the TF), this should be discussed with the Project Management Office (PMO) and a solution should be found that reallocates the effort required to match the assigned work (e.g. by moving some PMs from a WP that is not related to the TF to one of the WP that is involved with that TF).

2.3.4 Current task forces

At the time of writing of this deliverable, five task forces have been formed for the following services:

- Safe Replication,
- Dynamic Replication (or Data Staging),
- AAI,
- Joint metadata domain, and
- Simple store (previously known as “Researchers Data Store”).

The work undertaken by these task forces is included in this document.

2.4 Service handover procedure

The operational teams need all necessary technical and operational details to deploy and manage the services. Providing this technical and administrative information is part of the work done by the service building teams. The service building teams are responsible for describing the generic configuration of
the service (e.g. of a reference installation), while the WP6 service teams are responsible for the rollout including the adaptation and of the concrete configuration of the service and service components to the specific environments at the sites. These site and community specific documentations have to be provided by WP6. The main pieces of information and service documents that are required are listed below.

1) Technical contact:
   This specifies the person who will directly interact with (and instruct) the operational team contact to setup and consolidate the service in production. This is also the person who is in charge of producing documentation to promote the service, and providing that to the dissemination team.

2) Description of the test instance of the service:
   a) infrastructure (hardware/system resources) – the actual list of resources on which the test instance of the service was deployed.
   b) logical (software component) – a description of all software component of the service and how they interact with each other (with a schema/diagram, if possible).

3) Clear list of the resources needed so that the operational team can:
   a) define which partner is suitable for hosting the service (if possible, including information about whether there are plans to extend the service in terms of available resources, such as more disk space, or if more computing is needed), and
   b) properly evaluate the monitoring system needed (which features should be monitored and which checks should be implemented).

4) Expected/Requested Quality of service (resiliency, performance, and so on):
   This is necessary to eventually deploy service fail-over or load-balancing features, if they are available. This is also used to tune monitoring parameters (for example, how often monitoring checks should be run and how notifications should be configured).

5) Installation/Deployment manual:
   This is useful for the operational team, who are responsible for coordinating the service installation autonomously.

6) Description of basic operations:
   a) Service management procedures (start/stop, reconfigure, etc)
   b) User management procedures (enable/disable users, management of privileges)

Like the services, the technical documentation is also subject to future updates and refinements as far as the generic service is concerned.
3 EARLY CANDIDATE SERVICES

From the cases described for each community, the following services were identified as relevant for realizing the CDI.

- Safe Replication of Data (SR) is a service allowing communities to easily create replicas of their datasets in multiple data centres.
- Dynamic Data Replication (DR) is a service that is responsible for moving datasets close to HPC facilities and ingesting the analysis results back into the EUDAT infrastructure.
- EUDAT Metadata Domain (MD) is a service for aggregating metadata from multiple communities, and providing search facilities with results that link back to the actual resources.
- The Simple Store Service (SSS), previously called the Researcher Data Store (RDS), will provide an easy way for researchers to upload resources with minimal effort. All data uploaded through the SSS should be open and publicly available, and have proper metadata associated with it.
- A Persistent Identifier (PID) service to manage and facilitate the registration of persistent identifiers for the communities.

These services, listed by community, are summarized in Table 1.

Table 1: Community Service Matrix

<table>
<thead>
<tr>
<th>Community</th>
<th>SR</th>
<th>DR</th>
<th>MD</th>
<th>SSS</th>
<th>PID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLARIN</td>
<td>X</td>
<td>+</td>
<td>X</td>
<td>X</td>
<td>+</td>
</tr>
<tr>
<td>ENES</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>EPOS</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>VPH</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>LifeWatch</td>
<td>X</td>
<td>+</td>
<td>X</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

"X" indicates that the service is relevant to the community, whereas "+" means that the community has interest in this service but at a later stage, or that the community already has a similar service running in production.

Although there is no separate AAI use case, all the interviewed communities mentioned having specific AAI requirements, so it is considered to be an enabling service that is an integral part of the CDI and defined common services.

Work on the first generation of the “Safe Replication”, “Data Staging”, “PID”, “Federated AAI”, “Joint Metadata Domain” and “Simple Store” services is described in the next sections.

3.1 Safe Replication

The “Safe Replication” service enables EUDAT communities to easily create replicas of their scientific datasets in multiple data centres. This service can be considered as the fundamental service in EUDAT for storing data reliably, accessible and persistently in an environment of distributed repositories in different administrative domains. The degree of reliability, accessibility and persistency depends on the variety of storage technologies used, the number of replicas and the quality of assurance that the centres are offering. While deliverable D4.1.1 provides more information about the requirements associated with such a service, deliverable D5.1.1 outlines several technology areas that are required, along with potential candidates. This section is based on these two documents and describes the implementation of the early candidate services in order to realize the “Safe Replication” service.
3.1.1 Concrete reference architecture

![Concrete derived architecture for the safe replication service](image)

The conceptual blueprint of the service is given in deliverable D4.1.1. Figure 5 illustrates the derived concrete architecture for the service using technologies that are introduced further in this document. Figure 6 depicts the deployment locations in the context of the employed technologies (i.e. EPIC PID service, iRODS, etc.). Each user community and service provider infrastructure will be heterogeneous (for example, the firewalls, and accounting systems will differ). The architecture in Figure 5 helps us to understand the problems faced in a distributed system such as the federation domain established by EUDAT, which includes different organizations and administrative domains. It needs to be understood that, while the work moves from ‘test’ phases to ‘production’ phases, many aspects within the different data replication islands are different (such as user accounts, firewall setups, and DMZ setups). These differences need to be observed, and solutions need to be explored in order to ensure a production status towards the end of the TD lifetime. A more comprehensive reference model and architecture is part of the EUDAT work plan. The concrete architecture in Figure 6 provides a ‘frame of reference’ for stakeholders, while there will be concrete derived architectures for each island with concrete technologies deployed on real existing servers. A key functionality is the inter-working of iRODS and EPIC that is conceptually illustrated in Figure 6. Subsequent sections provide more implementation details (for example, about micro-services and rules).

![The safe replication service key functionality is the inter-working of iRODS and EPIC](image)
3.1.2 Implementation plan

The “Safe Replication” service will be built from tests within a number of ‘community and service provider islands’ to a production level service which spans all the prime user communities (with the exception of LifeWatch). The islands with their stakeholders are summarized in Table 2. This includes the initial EUDAT data centres and service providers for running services. The islands work in parallel and can achieve their milestones independently of each other. Once significant progress is made on the work on different segments with the initial communities and service providers, a second implementation period will include more data centres and service providers towards the end of the TF lifetime.

Table 2: Overview of the three different islands that implement the “Safe Replication” service

<table>
<thead>
<tr>
<th>Island</th>
<th>Communities/Community Centers</th>
<th>Service Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CLARIN/MPI-PL</td>
<td>RZG, SARA</td>
</tr>
<tr>
<td>2</td>
<td>ENES/DKRZ</td>
<td>FZI, CSC (second period)</td>
</tr>
<tr>
<td>3</td>
<td>EPOS/INGV, VPH/UCL</td>
<td>CINECA, SARA (second period), PSNC (second period)</td>
</tr>
</tbody>
</table>

The work within the task force and across the three islands is organized into three major segments, each including one or more phases. The highlights of the work in each of the segments are as follows.

- **Segment 1 - Service Building (pilot and pre-production phases):**
  Start implementing storage services based on iRODS; integrate iRODS in the local Long-Term Archives; implement initial replication policies; pilot replications within and across the different islands.

- **Segment 2 - Test and Evaluation (test and evaluation phases):**
  Test replication according to defined policies (test data set); test robustness and availability of the services; evaluate the registration of the multiple file locations in the EPIC/Handle system.

- **Segment 3 - Production (production and TF termination phase):**
  Hand-over a full storage service based on iRODS to production; user communities start replicating data to more than one EUDAT data centre; distribute user manual of task force to new user communities.

The implementation time-frame of the segments with the different phases is illustrated in Table 3.

Table 3: Overview of the planned time-frame of the implementation of the “Safe Replication” service

<table>
<thead>
<tr>
<th>Segment 1: Service Building</th>
<th>Segment 2: Test and Evaluation</th>
<th>Segment 3: Production</th>
<th>TF Termination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot</td>
<td>Test</td>
<td>Simple Production</td>
<td></td>
</tr>
<tr>
<td>Pre-Production</td>
<td>Evaluation</td>
<td>Extended Production</td>
<td></td>
</tr>
<tr>
<td>Extended Pre-Production</td>
<td></td>
<td>Full Production</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
<th>M7</th>
<th>M8</th>
<th>M9</th>
<th>M10</th>
<th>M11</th>
<th>M12</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/2012</td>
<td>02/2012</td>
<td>03/2012</td>
<td>04/2012</td>
<td>05/2012</td>
<td>06/2012</td>
<td>07/2012</td>
<td>08/2012</td>
<td>09/2012</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.1.3 iRODS

The iRODS technology plays a central role in the Safe Replication service. This section provides an overview of this technology and describes the main components of iRODS used in the Safe Replication service.

3.1.3.1 A global overview of iRODS

iRODS\(^3\) is a community-driven, open-source, data grid software solution. It provides a means for managing large distributed collections of digital objects, maintaining metadata and applying data management policies. iRODS comes with a comprehensive set of generic capabilities — although in a concrete setup, usually only a subset of these is required. The functionality can be extended further by user-defined rules and micro-services. Hence, an iRODS-based data grid can be tailored exactly to the needs of the end-users.

The central components of each iRODS installation are the iCAT metadata store, the Rule Engine and storage resources (such as the iRODS Server), as shown in Figure 7. The iCAT database is an SQL database managed by iRODS. It contains both information needed for authentication and authorization decisions, as well as user and system metadata describing the objects being managed. Data objects that are managed by iRODS are stored on storage resources i.e. software/hardware systems able to store data. A standard iRODS installation creates a default resource (a so-called Vault), but it is possible to add further resources — for instance, to account for increasing user requirements. It is possible to create a group of iRODS servers managing distributed resources. A group of such servers is called an iRODS Zone. It is important to stress that such a zone always includes exactly one iCAT server. Thus, a zone usually represents a single administrative domain (as the iCAT server stores authentication and authorization information).

![Figure 7: Basic overview iRODS](image)

---

\(^3\) Home page iRODS: [http://www.irods.org](http://www.irods.org)
3.1.3.2 iRODS zones and federations

The data that is managed in iRODS is presented as hierarchical collections of objects and is physically stored on the storage resources. Users use logical names for the collections of objects to manipulate their contents. An example of a logical name would be:

/tempZone/home/testuser1/file.txt

These logical names are composed of the zone name (tempZone), path (/home/testuser1) and the name of the actual object (file.txt). Each data object has only one logical name, but can be replicated on multiple storage resources, thus any data object can have a number of physical locations. The replication of a data object in iRODS involves physically copying the object to a new resource within the same zone, in such a fashion that the process is transparent to the user – that means that the logical name of the replicated object does not change. When a particular data object is accessed by a user, iRODS can deliver the contents of the object from one of the replicas, rather than having to use the original version of the object – for instance, from the physical resource which is closest to the requesting user.

iRODS also provides a means for connecting distinct administrative domains together to create iRODS Federations. As explained previously, each of the zones maintains its own user database and the information needed for authorization decisions in iCAT. To move data between iRODS zones within one federation, synchronization can be used. However, data objects transferred in this way will be treated as new data objects (as opposed to when data is replicated within one zone – which is transparent for the user). The replication process as defined in the Safe Replication service is not the same as the replication that is defined in iRODS and had to be implemented by means of synchronization between iRODS zones. Therefore to indicate that data items from different zones are in fact the same digital object, external identifiers had to be introduced. This has been achieved by using Persistent Identifiers (PID). They play a similar role to the logical names used within a single iRODS zone.

During the configuration of iRODS, it is possible to define a set of data management policies in the form of so-called rules. Rules are definitions of actions that should be performed by the server when a given pre-condition (such as a data ingest) occurs. The definitions of these rules consist of micro-services (atomic actions) and calls to other rules. The iRODS server has a built-in Rule Engine that triggers the rules, and also controls their execution by calling the underlying micro-services as needed.

3.1.3.3 Integration with mass storage: Mounted collections versus Compound resource

Mounted collections

A standard iRODS installation creates one storage resource (known as a Vault), but it is also possible to manage multiple resources (e.g., multiple directories where data should be stored or multiple disks/partitions). Such resources are managed entirely by iRODS, and accessing them directly might render the data in iRODS unusable. Such native iRODS resources are perfectly suited for providing storage space for new data ingested into the system. Sometimes, however, iRODS is installed so as to provide access to existing data sets. This can be achieved by using mounted collections. iRODS does not maintain any metadata information for such collections – in particular, it has no information about the subdirectories and files in the collection. When using this option, iRODS works as a "proxy": each time a request is issued for a data object from a mounted collection, the request is simply passed on to the underlying file system. The advantages of this are twofold. Already existing data can be easily and quickly made available via iRODS (without labour-intensive and time-consuming re-ingests). Secondly, it is possible to modify the content of the mounted collections by using tools other than iRODS without the danger of creating inconsistencies. For a normal collection of data, any low-level direct access and
modification (such as the physical removal of a file on the resource) would lead to an inconsistency: iRODS would keep the record describing the data object in the iCAT database, although the object would not be present anymore. The major disadvantage of mounted collections is the fact that metadata functionality is not available.

**Compound resource, Universal MSS driver**

As explained previously, data objects in iRODS are stored on storage resources. Each native iRODS storage resource (that is, the resources that are not mounted collections) must have a unique name, a type and a resource class. The resource can be located on any iRODS server within a given zone (unambiguously identified by the hostname and path).

Currently the list of available resource types comprises of the following:

- Unix file system,
- HPSS file system,
- Windows file system,
- S3,
- MSS universal driver,
- database, and
- Micro Service Objects (MSO).

The two main resources classes are:

- cache (short access times e.g., file systems), and
- archival (longer access times e.g., tape systems).

There are also special resource classes which are **compound resources**. These can be used to integrate HPSS, Amazon S3, FTP and ADS type resources into iRODS. A compound resource is a class where the POSIX data access type I/O calls (such as open, read, write, lseek and close) are not readily available. Instead, put and get calls are used to transfer entire files. The compound resource implementation in iRODS requires a cache class resource to be configured in the same resource group as the compound resource. Data stored in the compound resource cannot be accessed directly – instead it is accessed through the cache resource with staging and synchronizing using the put/get driver functions. When a data object is ingested into a compound resource, it is first written into the cache, and then transferred from the cache to the archive. When a user requests an existing data object, it is moved from the archive into the cache front-end resource.

Apart from the standard resource classes available in iRODS (like file system or HPSS), it is also possible to integrate special storage solutions. iRODS provides a **Universal Mass Storage System driver** which can be used to implement access to such special storage solutions. However, they can only be used as compound resources. iRODS installations provide a scaffold for a storage driver, which can be found in iRODS/server/bin/cmd/univMSSInterface.sh. The following (self-explanatory) methods must be implemented to integrate a storage solution into iRODS:

- syncToArch ()
- stageToCache ()
- mkdir ()
- chmod ()
- rm ()
- mv ()
- stat ()
Figure 8 shows an example of the implementation of the Mass Storage System (MSS) universal driver to a dCache storage system.

```bash
grep "()" /opt/iRODS/iRODS/server/bin/cmd/univMSSInterface.sh
syncToArch () {
    # <your command or script to copy from cache to MSS> $1 $2
    # e.g.: /usr/local/bin/rfcp $1 rfioServerFoo:$2
    /bin/cp $1 $2
    return
}
--
stageToCache () {
    # <your command to stage from MSS to cache> $1 $2
    # e.g.: /usr/local/bin/rfcpscp rfioServerFoo:$1 $2
    /bin/cp $1 $2
    return
}
--
mkdir () {
    # <your command to make a directory in the MSS> $1
    # e.g.: /usr/local/bin/rfmkdir -p rfioServerFoo:$1
    /bin/mkdir -p $1
    return
}
--
chmod () {
    # <your command to modify ACL> $1 $2
    # e.g.: /usr/local/bin/rfcchmod $2 rfioServerFoo:$1
    /bin/chmod $2 $1
    return
}
--
rm () {
    # <your command to remove a file from the MSS> $1
    # e.g.: /usr/local/bin/rfrm rfioServerFoo:$1
    /bin/rm $1
    return
}
--
mv () {
    # <your command to rename a file in the MSS> $1 $2
    # e.g.: /usr/local/bin/rfrename rfioServerFoo:$1 rfioServerFoo:$2
    /bin/mv $1 $2
    return
}
```

**Figure 8: Mass Storage System universal driver to dCache**

### 3.1.3.4 The rule engine

iRODS makes it possible to define data management policies in the form of rules. The rules can be compared to a workflow, or chains of actions composed of calls to other rules and/or micro-services. A micro-service is a "unit of work" in iRODS: each micro-service represents an atomic operation. iRODS provides a comprehensive set of such micro-services, for instance, to calculate a checksum or replicate files.

---

4 www.dcache.org
data. From the micro-services that are provided, quite complex workflows can be built to reflect complex data management policies. Furthermore, it is possible to extend iRODS by defining new micro-services. This document contains an example of how some new micro-services for PID manipulation were implemented.

The execution of the rules is illustrated in Figure 9. The execution process is controlled by the internal iRODS policy engine. The system rules are defined by the administrator and can be found in the \texttt{server/config/reConfigs/core.re} file. There are already a number of useful examples which can be adjusted for concrete setups.

![Figure 9: iRODS Rules Flow diagram](image)

Figure 10 shows an example of a rule which is called each time a file is ingested into a given collection (/tempZone/home/rods/monitored). Each time this happens, the collection will be replicated to /targetZone/home/rods/safe-copy by the means of synchronization, and a message will be written into the server log. This is a very simple policy which ensures that the ingested data is available at two distinct locations.

```plaintext
acPostProcForPut{
  ON($objPath like "/tempZone/home/rods/monitored/*") {
    msiSplitPath($objPath,*collection,*fileName);
    msiColIrsync(*collection,
      "/targetZone/home/rods/safe-copy",
      "demoResc",
      "IRODS_TO_IRODS",
      *Status);
    writeLine("serverLog", 
      "rsync of *collection to its safe-copy done \ (status=*Status). Triggered by creation of $objPath");
  }
}
```

Figure 10: Example rule to safe copy data
3.1.4 Building the PID micro-service

According to the specifications for the Safe Replication use case, data ingested into the service should be assigned a Persistent Identifier (PID) and subsequently be replicated to a defined number of EUDAT data centres. Information about the repositories where the data object is located (replicas) should also be included in the PID record. As mentioned previously, the policy-based replication in the EUDAT Safe Replication Service is performed with iRODS. In iRODS, data management policies are defined by so-called system rules, that is, workflows defined by the data administrators and triggered by the iRODS system, for example, when a data ingest occurs.

In the early stages of the project, it was decided that the PIDs should be registered with the EPIC system (technically based on the Handle standard (footnote 255 on page 25). This service is described later in this document (section 3.3.1 on page 37). For the remainder of this section, it is enough to know that the Handle implementation provided by SARA (which was used in the Safe Replication service) offers a simple HTTP-based RESTful interface to manipulate handles. It offers a basic set of operations that can be subsumed under the term CRUD (Create/Read/Update/Delete). Unfortunately iRODS does not provide micro-services to interact with Handle system, so these had to be developed.

Technically, the interaction with any REST-based service boils down to exchanging representations of objects known to the service domain. This is also true for the Handle system implementation provided by SARA where representations of handles are exchanged. In particular, there are three possible representations:

- HTML,
- JSON, and
- text.

However the text representation only contains partial information from the handle record, and thus was discarded as an implementation option. The HTML representation is more convenient for human users as it can be interpreted by a web browser and presented in a readable form. JSON representations also contain all the information from the handle record, and they have the advantage of being much easier to use in machine-to-service interactions.

To provide a means for manipulating PIDs in iRODS, EUDAT needed to develop a set of micro-services to interact with the Handle system. Consequently the micro-services need to be able to digest and manipulate JSON documents and transfer them through HTTPS.

A small EUDAT team was formed and charged with the task of integrating the EPIC system into iRODS by implementing a set of PID manipulation micro-services that could be used in iRODS rules.

The PID Integration team consisted of the following representatives:

- Willem Elbers (MPI-PL), Leader,
- Javier Bertolome (BSC),
- Claudio Cacciari (CINECA),
- Elena Erastova (RZG), and
- Jedrzej Rybicki (FZI).

The idea was to keep the team small, and thus make it possible to use agile development for this crucial component.
### 3.1.4.1 EUDAT Persistent Identifier (PID) record description

The EUDAT Safe Replication (SR) task force agreed to use the approach shown in Figure 11 for the first service delivery. This approach allows for replica management, discovery and verification within the EUDAT infrastructure in such a way that each centre operates in an independent administrative domain.

![Figure 11: PID management scenario](image)

This means there are several things that the PID record needs to do:

1. store multiple URIs,
2. store a repository of record (RoR) reference, and
3. store a checksum to be able to verify the replicas.

This results in a PID record as shown in Table 4.

#### Table 4: PID record description

<table>
<thead>
<tr>
<th>PID identifier or Handle</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10320/LOC</td>
<td><code>&lt;locations&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;location&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;location&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;location&gt;</code></td>
</tr>
<tr>
<td>ROR</td>
<td><code>&lt;URI to RoR PID record&gt;</code></td>
</tr>
<tr>
<td>CHECKSUM</td>
<td><code>&lt;checksum value, e.g. SHA256&gt;</code></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Handle record field description:

- **PID identifier or Handle**
  - The PID identifier consists of two fields separated by a “/”. The fields are as follows.
    a) `<NamingAuthority>` (or prefix) identifies an authority (such as a company or institute) that is able to create handles. Within EUDAT, all communities and service providers are individual naming authorities.
    b) `<LocalName>` reflects the name of the Digital Object (DO) within the local naming scheme used at the NamingAuthority.
- **10320/LOC**
  - Each community centre or data centre will store a number of URIs in the PID record, in the “10320/loc” field and format, with two distinct purposes:
    a) pointing to the location where the DO can be accessed via the specified protocol, and
    b) pointing to child PID records that are each associated with a replica that resulted from a replication from the centre in question to another centre.
- **RoR (Repository of Record)**
  - All data centres store RoR references – each of which is a URI pointing to the RoR PID record. (Typically the community centre that is effectively the owner of the digital object does not store a RoR field).
- **CHECKSUM**
  - A checksum should be stored in order to easily verify the validity of each replica. The specific checksum algorithm is not imposed by EUDAT – that is left up to each community.

There are no other restrictions or requirements imposed on the PID record.

### 3.1.4.2 Implementation

The process of implementing the micro-services started by assessing the ways that external programs could be integrated into iRODS. There were at least three possible options:

1. *native iRODS modules* written in C,
2. *embedPython* iRODS modules that would allow micro-services to be defined in Python, and
3. external scripts that could be called by *msiExecCmd* micro-services.

*Native iRODS modules* have to be written in C. They are composed of sources, headers, a *Makefile* and a special file (*microservices.table*) that assigns micro-service names to C functions defined in the module. Micro-services written in this fashion can access iRODS functionality by using the iRODS C API and are the most “natural” way of extending iRODS (as iRODS is also written in C).

*EmbedPython* is an external project that provides a set of micro-services to run external Python scripts. This makes it possible to implement any required functionality in Python, and then integrate the Python program into iRODS. By using *pyrods* (which is the Python iRODS API) it is easy to access iRODS functionality. To run such micro-services it is, of course, necessary to install both the Python interpreter, pyrods, and the *embedPython* iRODS module.

When using external scripts, it is theoretically possible to use any language to implement missing functionality, and to then call such a program using the *msiExecCmd* micro-services, passing arguments to it as required. However, this approach makes it harder to access iRODS functionality from external sources.

---

5 [http://www.handle.net/rfc/rfc3651.html](http://www.handle.net/rfc/rfc3651.html)
6 [http://www.handle.net/overviews/handle_type_10320_loc.html](http://www.handle.net/overviews/handle_type_10320_loc.html)
programs, especially if the program is written in a language that does not have an iRODS API. Therefore this option was rejected.

3.1.4.3 Pilot implementations

In order to compare the two remaining options, pilot implementations of the PID micro-services were created using the two acceptable options. The final decision about which option to use was based on the results and experiences from these practical exercises.

The proof-of-concept implementation of the PID manipulation micro-services in C was provided by Jedrzej Rybick (FZI). The results\(^8\) of this exercise are available in the EUDAT SVN repository. The implementation facilitates external libraries for JSON manipulation (jansson) and HTTP communication (libcurl). In particular, it was noted that manipulating JSON was much harder than in Python, and that there was a lack of C-programming expertise amongst the partners. Hence there was a possibility that the production-ready implementation would not be ready in time, and it was likely to be harder to maintain the module.

The proof-of-concept implementation of a Python-based EPIC client was provided by Javier Bartolome (BSC) and the integration into iRODS was done by Willem Elbers (MPI-PL) and subsequently by other partners. It was found that the JSON manipulation was much easier when using the Python external library (simplejson). For the HTTP interactions, the Python library (httplib2) was used. The implementation\(^9\) was made available in the project SVN repository.

This was tested by all the partners and was proven to work properly. A further advantage of this approach was that there was much more Python expertise available among the project partners. Hence quicker development, testing and integration were possible. Maintenance was also expected to be easier.

**Recommendation**

After internal discussions, the Safe Replication Task Force made the following recommendations:

- For the pilot implementations, the Python module should be used.
- Because native iRODS modules are written in the C programming language it is advisable when the service reaches production-ready status and is handed-over to operations to revive the effort on a C implementation of the micro service.

3.1.4.4 Usage examples

The Python code to create a Handle in EPIC service offered by SARA looks like Figure 12 (many details omitted):

---

\(^8\) https://svn.eudat.eu/EUDAT/Services/DataManagement/PID/iRODS-Integration/C
\(^9\) https://svn.eudat.eu/EUDAT/Services/DataManagement/PID/iRODS-Integration/Python
The workflow is pretty simple: first a URL is created. This is composed of the service URL, a \texttt{prefix} (which is unique for each partner) and a \texttt{suffix} (which is unique for each object). Then a JSON document is prepared from the parameter array \texttt{keyValues}. Before the HTTP request PUT is sent, the HTTP headers are extended to include the information that the JSON representation will be using. Finally the whole HTTP message, composed of the header and body, is sent to the previously-prepared URL, and the service response is interpreted and returned.

The method \texttt{createHandle} from the file \texttt{epicclient.py} can be included in an iRODS rule as shown in Figure 13.

Firstly the Python interpreter is initialized, then the method \texttt{"createHandle"} (from the file \texttt{epicclient.py}) is called with following parameters: \texttt{*prefix}, \texttt{*suffix}, and JSON contents (as a string). The result of
the operation can be accessed via the variable *res*. If no more Python-based micro-services will be called, the Python interpreter can be finalized.

### 3.1.4.5 Problems identified

The Python micro-services were successfully tested by all the partners involved in the pilot implementations of the Safe Replication Service. The goal of integrating the Handle PID service into iRODS was achieved. However, one of the partners (FZI) identified a severe security issue in `embedPython`. In short, this means that code can be remotely executed on an iRODS server by unauthorized users with administrating privileges. The issue was reported\(^{10}\) to the `embedPython` developers and confirmed by them. At the time of writing, there is no security fix available, and thus `embedPython` is not to be used in the production environment.

The existing EPIC client can still be used if it is called by the `msiExecCmd` iRODS micro-service. However, this can only be done safely when using iRODS version 3.1 or higher. The reason for this is another security problem identified by FZI in iRODS version 3.0, which has also been confirmed by the iRODS developers.

Given these security issues, the long-term recommendation for operating the Safe Replication Service is to extend and use a C-based module for PID manipulation.

### 3.2 Data Staging

When scientific communities use data-intensive applications, their data inputs are typically staged onto computational resources prior to job execution. Following execution, output data sets are generally staged out of compute nodes and onto storage systems. Additional data transfer operations may be required to move intermediate data products from nodes where they are produced to nodes where they will be consumed by other jobs. If the storage space on a node is limited, these stage-out operations need to be performed before other jobs can run. In distributed computing, where resources are geographically dispersed, efficient data staging is essential. Starting with this assumption, the objective of the data staging service is to enable research communities to stage data from EUDAT repositories to external HPC facilities for further processing. The “Data Staging” service can be considered as the fundamental service for exchanging data between the EUDAT domain of globally registered data and the domain of data consumers and producers (e.g. scientific instruments, HPC or HTC facilities). Three EUDAT core communities (VPH, ENES and EPOS) explicitly requested this service as they need to perform statistical model analyses on stored data.

This service area differs from the safe replication scenario, which was discussed in the previous section, on three main points.

- The replicated data are discarded when the analysis application ends.
- The PID references are not applied to data replicated into scratch workspaces.
- The users initiate the process of replicating data, whereas in the safe replication scenario, data are replicated automatically on a policy basis.

A typical utilization scenario is depicted in Figure 14.

1. The desired data are replicated from community storage to an EUDAT node using the “safe replication” solution.
2. Next the data are replicated between two EUDAT nodes, since the destination node is close to the target HPC facility. The corresponding PID record contains all the relevant URLs for the copies.
3. The data are then staged into a HPC facility that is either close to the EUDAT node or available elsewhere, for example, as part of the PRACE infrastructure.\textsuperscript{11}
4. Finally the replicated data are staged into the local HPC facility. Afterwards, the analysis results will be staged out to the original source, and the results can be copied back to the community storage.

### 3.2.1 Implementation plan

The Data Staging service is being developed using a two-pronged approach that consists of identifying and addressing the major requirements of user communities, and also developing collaborative relationships with leading European e-Infrastructures, such as PRACE\textsuperscript{12}. Several technologies – including iRODS Clients\textsuperscript{13}, GridFTP\textsuperscript{14}, Griffin\textsuperscript{15}, gTransfer\textsuperscript{16}, Parrot\textsuperscript{17}, UNICORE FTP\textsuperscript{18}, Globus On-line\textsuperscript{19} and the XSEDE File Manager\textsuperscript{20} – were evaluated for this purpose. They were assessed on their ability to meet the requirements of the communities, their code maturity, fault tolerance and reliability, and their facility for interaction with PRACE data services.

\textsuperscript{11} PRACE infrastructure, http://www.prace-ri.eu/
\textsuperscript{12} A collaboration plan is currently under discussion between PRACE and EUDAT
\textsuperscript{13} iRODS, https://www.irods.org/
\textsuperscript{14} GridFTP, http://globus.org/toolkit/docs/latest-stable/data/gridftp/
\textsuperscript{15} Griffin, http://code.google.com/p/datafabric-griffin/
\textsuperscript{16} gTransfer, https://github.com/fr4nk5ch31n3r/gtransfer/
\textsuperscript{17} Parrot User’s Manual, http://www.cse.nd.edu/~ccl/software/manuals/parrot.html
\textsuperscript{18} UNICORE, http://www.unicore.eu
\textsuperscript{19} Globus On-line, http://www.globusonline.org
\textsuperscript{20} XSEDE File Manager, http://www.xsede.org/web/guest/data-transfers
The implementation plan began with pilot tests in a number of community service provider islands. These were then developed further into a production level service that spanned all the interested communities and also incorporated the initial EUDAT service providers for running services.

The work was carried out in four different phases: scouting, design/implementation, pre-production, and production.

- The **scouting** phase was based on four separate activities that were designed to check which components/technologies could be used to implement the final service. During this phase, service providers and communities were also assessed to gain a better understanding of their internal organization, including their infrastructure configurations and policies (i.e. their level of security).

- The **design/implementation** phase consisted of two different pilots, each of which focused on a different scenario. The pilot studies were carried out by two organizational islands (see Table 5). The design phase started in parallel with the technology scouting phase, and provided input for the implementation of the final service. The design phase is still active and is working integrating the service with the upcoming AA Infrastructure; this step is mandatory before moving the service into production.

- The **pre-production** phase is concerned with the execution of final tests and the evaluation of service quality factors such as usability, reliability and availability. All the stakeholders have been involved with the final tests.

- The **production** phase involves rolling the service onto the production infrastructure and promoting it to the user communities.

Note that the Data Staging service will be built from tests within a number of “community and data centre islands” to a production level service which spans all the prime user communities (e.g. EPOS and VPH). The islands with their stakeholders are summarized in Table 5. The islands work in parallel and, depending on the specific user requirements, evaluate different solutions on the basis of common services.

### Table 5: Data staging pilot islands

<table>
<thead>
<tr>
<th>Island</th>
<th>Community</th>
<th>Service Providers</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VPH/UCL</td>
<td>CSC, PSNC (second period)</td>
<td>Development of a solution for staging data between resources belonging to the same site using iRODS low level commands</td>
</tr>
<tr>
<td>2</td>
<td>EPOS/INGV</td>
<td>CINECA, SARA</td>
<td>Development of a solution for the remote staging of data from an EUDAT node onto an external HPC facility using third-party services</td>
</tr>
</tbody>
</table>

The two different scenarios are described in the next sections. The implementation time-frame of the different phases is illustrated in Table 6. Blue boxes represent the work that remains to be done for the integration of the service with the AA Infrastructure.
3.2.2 Staging data using third-party services

This scenario addresses the case where data needs to be staged onto a remote HPC facility which is located outside the EUDAT domain. The first version of this service – which will be offered in production in the coming months – extends the iRODS system (which is at the base of the EUDAT storage offering) with a GridFTP interface. Thus, by accessing EUDAT endpoints, data can be placed using any GridFTP client. GridFTP is a mature and widely-adopted technology that, by extending the standard FTP protocol, allows high-performance data transfers to be executed in a very efficient and reliable way between two remote sites. It also offers other valuable functionalities such as executing parallel data transfers for improving network utilisation, third-party (direct server-to-server) transfers, and pipelining for optimizing the transfer of many small files. GridFTP has the flexibility to make it possible to tune some network parameters (such as the TCP buffer size, or number of streams) so as to optimize the data transfer performance. A further advantage is that the GridFTP protocol is already used to move data within the PRACE Infrastructure.

In order to enable iRODS to support the GridFTP protocol, the Griffin component was successfully tested and integrated into the service. Further details on Griffin are presented in the following section.

Alternative technologies, such as FTS\textsuperscript{21} from the XSEDE File Manager\textsuperscript{22}, are currently under evaluation and could be employed in the near future to extend the service to support further functionalities. There are also plans to integrate the service with the upcoming EUDAT AA Infrastructure (see Section 3.4).

3.2.2.1 Griffin

Griffin is a GridFTP server, entirely written in JAVA, that can access iRODS resources as well as plain file systems. Leveraging the Globus Toolkit\textsuperscript{23} libraries makes it possible to access and stage data from iRODS to any external GridFTP server or client. Thanks to a powerful bash script, the installation is relatively simple, although the configuration may require more effort and expertise (as the official documentation lacks descriptions of some fundamental steps). Consequently, it took a long time to set up the test-bed properly; server X.509 credentials needed to be included in the JAVA runtime trust-store and that was not described adequately in the official documentation. Like a traditional GridFTP server, Griffin makes use of X.509\textsuperscript{24} proxies for authenticating users and supports most of the GridFTP features – including

\begin{footnotesize}
\begin{itemize}
\item File Transfer Service, http://egee-jra1-dm.web.cern.ch/egee-jra1-dm/FTS/
\item XSEDE File Manager, http://www.xsede.org/web/guest/data-transfers
\item Globus Toolkit, http://www.globus.org
\end{itemize}
\end{footnotesize}
multiple streams, tcp-buffer size tuning and pipelining files. The only unsupported feature is the multiple-stripe (that is, having multiple GridFTP nodes transferring parts of the same data set on different network paths). From the security point of view, Griffin behaves like a normal GridFTP server, thus requiring a control-channel port and a range of data-transfer ports to be made available to external clients.

Griffin is not widely used – there are probably only tens or hundreds of installations around the world – but its development team is actively working to maintain the code, and they are also following the iRODS evolution roadmap. The willingness of Griffin developers to promote their technology culminated in a prompt reaction where they fixed important bugs which were discovered during our tests using the Globus On-line service. Several tests were performed at CINECA and SARA where large data sets (containing hundreds of GBs of data) were transferred back and forth between the two sites without experiencing any instability problems or misuse of the machine memory.

Griffin is an open-source project, now hosted at Google Code\(^{25}\), and being released under the Apache License, version 2.0\(^{26}\).

![Example of a Griffin configuration file to access iRODS resources](image)

To permit Griffin to have access to iRODS resources, the GSI authentication mechanism needs to be enabled. A user who intends to stage a data set needs to be authorized to access the Griffin module: the user’s DN must be included in the proper authorization file (e.g., gridmap-file using the GridFTP terminology) and included in the database of iRODS users.

Enabling iRODS to communicate via the GridFTP protocol has several advantages, including the availability of several interfaces ranging from APIs to advanced web portals. The following section presents three options for accessing the iRODS-Griffin endpoint.

### 3.2.2.2 Accessing the service using any GridFTP client

As mentioned previously, data transfers can be executed using any GridFTP client (such as globus-url-copy, gTransfer, UberFTP, or Eclipse GridFTP Client) through accessing the “iRODS-Griffin” endpoint (Step 1 in Figure 16). In a client-server scenario, the user’s workstation plays an important role as it controls any such connection and must remain active for the whole duration of the transfer. Thus, the overall reliability of any transfer is determined by the capability of the user’s workstation to remain operative during the transfer.

---

26 Apache License 2.0, http://www.apache.org/licenses/LICENSE-2.0.html
In addition, most of the GridFTP clients are only able to manage one transfer at a time and rely on the user’s experience for tuning the connection parameters (which are vital for achieving good performance). Practically, stand-alone clients do not support auto-tuning.

### 3.2.2.3 Accessing the service using the Globus On-line service

Globus On-line is a web service that is hosted at the Amazon Elastic Compute Cloud. The aim of the service is to facilitate the transfer of huge data sets between two GridFTP endpoints (see Figure 17). Globus On-line makes it easy to execute any data transfer by helping the user to select data sets to stage and with the configuration of network parameters. It also offers an intuitive web interface that simplifies the control of running transfers.
With respect to basic GridFTP clients, the Globus On-line service offers three big advantages as follows.

- **Ease of use** – Users just need a web browser to execute, manage, and control their data transfers. Furthermore, once initiated, transfers can progress autonomously without relying on the availability of the user’s workstation.
- **Reliable data transfers** – The service offers a higher level of reliability as it incorporates logic designed to optimize end-to-end performance and to recover from transient failures. The Storage-as-a-Service (SaaS) approach also permits rapid (and transparent) software upgrades and expert operator diagnosis of persistent failures.
- **Auto-tuning of network parameters** – The overall performance of a transfer is strongly related to the correct tuning of network parameters, such as the number of parallel streams or the buffer-size of the TCP protocol. Choosing the right values for these parameters might not be an easy task for non-technical users. In order to overcome this limitation, Globus On-line automatically tunes network parameters based on a continuous analysis of connection characteristics and historical transfer information.

There is one drawback with Globus On-line – it might introduce potential security risks since user credentials need to be transferred from each user’s workstation to the central service. For this reason, PRACE has decided to not promote the adoption of this service to its users until the possible security risks have been mitigated27.

3.2.2.4 The data staging script

To support scientific communities in integrating the data staging service with their existing tools, a sample Python script was developed to demonstrate how to instrument remote data transfers using available APIs. The script is composed of two different modules: the first, called selector, is responsible for the selection of the data sets to be staged, while the second, called transfer, executes the real transfer using the preferred protocol. The modules can be replaced to support alternative transfer protocols or other methods for selecting data.

The current implementation (which is based on EPOS requirements but can be extended easily) utilizes an iRODS micro-service for the selection of the data sets and the Globus On-line API for their transfer.

```
prompt$ ./datastager.py -p /home/irods/data/archive -y 2004 -n MN -s AQU -c BHE -u cin0641a --ss ingv --ds GSI-PLX --dd /shared/data/userprace/tmp
```

**Figure 18: Data stager script utilisation**

The script takes various parameters as input, including the path of the archive (\texttt{-p}), the criteria for the selection of the data sets (\texttt{-year, -network, -station, -channel}), the username (\texttt{-u}), the source and destination sites (\texttt{--ss, --ds}), and the destination directory onto which the data is staged (\texttt{--dd}).

Once executed, the progress of the transfer can be monitored using the Globus On-line web interface (see Figure 19).

27The PRACE Security Team, in collaboration with the Globus team, is currently working to address this point.
3.2.2.5 The XSEDE File Manager

The data staging service can be also exploited via the XSEDE\(^\text{28}\) (Extreme Science and Engineering Discovery Environment) File Manager which offers a powerful interface to ship data either through the GridFTP protocol of the native iRODS one, based on FTP.

The XSEDE File Manager, shown in Figure 20, is being developed within the XSEDE project, provides users with an easy-to-use, drag-and-drop interface for managing data transfers across remote servers. Its evaluation was carried on in collaboration with the developers’ team being located at the Texas Advanced Computing Center\(^\text{29}\) (TACC) which also helped us in extending the client to provide further functionalities, such as the capability to directly interact with iRODS. Given the good results experienced during the testing phase, it is our intention to promote the XSEDE File Manager as preferred standalone client to access the data staging service.

![Figure 19: Globus On-line web interface](image)

![Figure 20: XSEDE File Manager Interface](image)

3.2.3 Staging of data using iRODS tools

In addition to the scenario presented in the previous section, it might be necessary to stage data between two endpoints within the same site using low-level data movement tools. This scenario is very

\(^{28}\) XSEDE, http://www.xsede.org

\(^{29}\) TACC, www.tacc.utexas.edu
common among sites offering both storage and computational resources – like the vast majority of the EUDAT sites which offer both types of resources under the same organisational umbrella. The adoption of alternative tools is necessary to address those scenarios where remote data staging tools cannot be used due to internal policy or configuration limitations, such as firewall rules that block inbound connections. Basically, the idea behind this service is to offer a solution whereby data can be staged by an HPC facility, so the data transfers are initiated by the HPC facility which effectively pulls the required data from the EUDAT resources.

The solution that has been chosen for this service is based on the iRODS command line interface (icommands)\(^{30}\). The main reasons behind the choice of the iRODS command line interface are as follows.

- It is reliable.
- It is scriptable.
- It provides high performance.
- It is easy to set up and is portable between many systems.

Figure 21 illustrates the process of staging data using icommands.

The iRODS command line interface is composed of various low-level tools that can be used to manage data stored within iRODS resources. For instance, to stage data into the HPC working directory (which is usually called the **SCRATCH** area), the `iget` command can be used, while `iput` can be used to ingest computation results back onto EUDAT resources (see Figure 22).

```
prompt$ iget -N 4 /home/irods/data/archive /shared/data/userprace/tmp
prompt$ iput -N 4 /shared/data/userprace/tmp /home/irods/data/archive
```

Figure 22: iRODS icommands utilisation example

Note that the transfer of data can be improved using parallel channels, each controlled by a separate thread (via the `-N` option).

\(^{30}\) iRODS icommands, https://www.irods.org/index.php/icommands
3.3 Persistent Identifier service

In the scientific data domain, many different levels of data can be distinguished. Raw data come from scientific machines (e.g. particle accelerators, MRI, microscopes, and telescopes) or from large scale simulations (e.g. climate or seismology applications). There are also processed datasets, analyzed datasets, and data used in publications. Even final publications are forms of data.

One of the main objectives of EUDAT is to deploy a common and persistent data e-Infrastructure. A core element of this EUDAT e-Infrastructure is the persistent data storage facility based on iRODS described in section 3.1.3. The automatic registration of persistent identifiers is an essential aspect of the persistent data storage facility. For storing these persistent identifiers, the handle services provided by EPIC will be used.

3.3.1 The Handle service

The Handle System was developed by CNRI and is an infrastructure upon which applications that serve many different purposes have been built. The main function of the Handle System is to manage references to objects. These references are called handles. Among the objects that are identified by handles are journal articles, technical reports, books, theses and dissertations, government documents, metadata and data sets. In EUDAT, the handles are mainly used as persistent identifiers pointing to digital objects, which consist of referable data (Figure 23).

3.3.1.1 Handle architecture

The Handle system (Figure 24) is a two-level hierarchical service model which consists of a top-level service (known as the Global Handle Registry or GHR) and all the other handle services (which are known as local handle services or LHS). An LHS can be a single site running a one-handle server, or an LHS could consist of multiple sites running multiple-handle services in which one of the handle services is designated as the primary – this means that particular service is responsible for handling all the requests to create, modify or delete handles. Any changes are then propagated from the master to the other sites. In this way, each site will have a complete set of handles managed within a single local handle service domain.

---

31 Home page the Handle System: http://handle.net/index.html
32 Corporation for National Research Initiatives: http://www.cnri.reston.va.us/
3.3.1.2 Handle System scalability

The GHR is the service that knows about each local handle service (LHS) and stores the prefix for each service. The handle for a naming authority can be resolved to obtain the service information that clients and servers need in order to locate the LHS. The GHR has a setup that is similar to an LHS, and can consist of multiple sites. At present, the GHR consists of three sites: a primary site, and two secondary sites running five top-level registry servers to which the “root information” is replicated. In principal, there is no logical limit to number of local handle services whose information can be managed by the GHR. The GHR currently stores information for more than a thousand handle services (located in 67 countries on six continents) and manages more than 200,000 prefixes.

In a logical sense, the scope of a local handle service is unlimited, and can range from a single site (running a one-handle service) to many sites (on which the prefixes and handles are managed by more than one handle server). The handle system has the flexibility to accommodate differences in local configurations, for example, it can deal with sites running fat handle servers (that manage all the prefixes and handles) or with a number of small servers (that manage some of the prefixes and handles), as shown in Figure 24.

3.3.1.3 Registered and non-registered prefixes

While it is possible to download, install and use the Handle software, that does not mean that the handles stored locally will be automatically available publically or searchable from the top-level. To make it possible to find handles from the global handle resolver, prefixes must be registered at the GHR. For this, a service-level agreement must be signed and an annual service fee must be paid. After registration, CNRI provides a registered prefix. After that, it is necessary to obtain an account at a local handle service provider. In the case of EUDAT, SARA will function as the master site for the EPIC consortium.
3.3.2 EPIC Handle Service

On basis of the technology appraisal, EUDAT has decided to use the EPIC Handle Service for registering persistent identifiers for digital objects. The EPIC Handle Service is provided by the EPIC consortium. The EPIC Handle Service (shown in Figure 25) acts as a Local Handle Service (LHS) which is spread across three sites (namely, GWDG, CSC and SARA), where SARA acts as the master site for managing handles (for example, creating, modifying or deleting).

![Figure 25: Overview EPIC Local Handle Service](image)

3.3.2.1 EPIC REST-based web service

For easy and automated management of handles, and in order to process more handles, EPIC has developed a REST-based web interface. This runs alongside the basic Handle service, storing handles in a central handle database, as shown in Figure 26.

At the time of writing, EPIC provides two API versions of the REST-based interface (version 1 and version 2). The current EPIC production service run at SARA provides version 1 of the API interface. Therefore, for the implementation of the PID micro-service, EUDAT is using version 1 of the API.

33 Home page EPIC consortium: [http://www.pidconsortium.eu](http://www.pidconsortium.eu)
34 EPIC API version 1 description: [https://docs.google.com/Doc?docid=0AV1__dd00fuZGNtcnZueDNfNdyODU2NXFjeg](https://docs.google.com/Doc?docid=0AV1__dd00fuZGNtcnZueDNfNdyODU2NXFjeg)
3.3.2.2 PID test bed

For development and testing purposes, and also to avoid unnecessary costs for registering tens or maybe hundreds of prefixes, SARA provided a separate test instance of the EPIC handle service. According to the needs of the PID micro service development team, SARA provided a total of eight prefixes and accounts (as shown in Table 7). These prefixes are not officially registered at CNRI. Therefore they cannot be found via the global handle resolver. The only way to find them is via the local resolver\(^ {36} \).

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Site</th>
<th>Prefix</th>
<th>Site</th>
</tr>
</thead>
<tbody>
<tr>
<td>841</td>
<td>SARA</td>
<td>845</td>
<td>DKRZ</td>
</tr>
<tr>
<td>842</td>
<td>MPI-PL</td>
<td>846</td>
<td>INGV</td>
</tr>
<tr>
<td>843</td>
<td>RZG</td>
<td>847</td>
<td>CINECA</td>
</tr>
<tr>
<td>844</td>
<td>FZJ</td>
<td>848</td>
<td>BSC</td>
</tr>
</tbody>
</table>

3.3.3 Memorandum of understanding between EUDAT and EPIC

One of the main objectives of EUDAT is the automatic handling of registered data objects. This distinguishes EUDAT from many of the other projects that are focused on data. The current EUDAT choice for registering and managing identifiers for data objects is the EPIC Handle system. Because the EPIC Handle Service is important for the success of the EUDAT project, EUDAT is working with the EPIC consortium and partners to formalize this relationship in a Memorandum of Understanding (MoU).

3.4 Federated AAI

The AAI aims to provide authentication and authorization infrastructures for EUDAT. The goal is to build on existing infrastructures and work as much as possible, while integrating with existing community practices (and improving on them, if necessary).

---

\(^ {36} \) EPIC test local resolver: [http://epic.sara.nl/](http://epic.sara.nl/)
3.4.1 Implementation plan

Briefly, the EUDAT AAI work is divided into three phases:

1. SSO for members of the project – using a single system to authenticate to collaborative services within the EUDAT project,
2. Federated identity management for pilot user communities – initially CLARIN and ENES (selected partly because they use Shibboleth and OpenID respectively), and
3. A full and scalable federated identity management – integrated with core EUDAT services, and with a framework for extending to other services.

In phase 1, CROWD (which is already used by CSC for their Atlassian Confluence service) was integrated with the request tracker (JIRA) by CSC. Therefore the JIRA service was extended to consume SAML identity assertions issued by the CROWD system in order to map these to usernames.

Other technologies were evaluated, and the results of the evaluations were documented in deliverable D5.1.1 section 3.5.5. Here the focus is on the main candidates for the Phase 2 AAI. This evaluation was based on both technical merit and early requirements specified by the communities. The next step was to evaluate the candidates against a more detailed set of user requirements, which are summarized in Table 8.

3.4.2 Architecture Overview

The overall architecture of the AAI is illustrated in Figure 27. The diagram shows a set of external Identity Providers (IdPs), and external Attribute Providers (AtPs), where, in the case of Shibboleth, the IdP itself is also an AtP 37.

At the boundary of the EUDAT core and community services (represented by the large circle) is a front end (or gateway), which accepts tokens and attributes from the existing AAI. The gateway converts an external credential to an internal credential. Note that it may be more efficient to use a single internal credential, rather than the many different credentials provided by the AAI, as otherwise every single service in EUDAT would have to be able to understand every type of credential. Instead, it is better to convert the external token into a single credential. One consequence of this approach is that the gateway now holds a credential with which it can act on behalf of the user. This credential has to be user-specific, as the service providers must know the individual users of the services (or be able to trace them in cases of misuse). The alternative is to use a single internal credential and then track very carefully who is doing what at which time, but this will not scale securely to a large multi-user multi-service infrastructure like EUDAT with multiple gateways 38.

---

37 Of course, every IdP is an AtP: an attribute which says “I have authenticated this person” (e.g. ePTID) is an attribute; an attribute like commonName (e.g. “Joe Bloggs”) is an attribute. The distinction that is being made here refers mainly to the use of the attribute: IdPs issue attributes which are used to identify the person, AtPs issue attributes used for authorizations to the entity identified by the identity attributes. The only way an IdP could authenticate a person without issuing an attribute is by generating only a session id.

38 The approach of VPH is essentially the same; an individual RFC3820 proxy certificate is generated for each user, but they all represent the same identity. Using different proxies for each user allows for giving them different (authorisation) attributes, but they still appear to have the same identity (viz., the Distinguished Name.)
Within EUDAT there is a range of services, represented as cylinders in Figure 27. These services all accept the EUDAT-internal federation credential. For authorization purposes, it is best to have each service perform its own access control. It is helpful to use the XACML\textsuperscript{39} language where the architecture\textsuperscript{40} comprises a Policy Decision Point (PDP), a Policy Enforcement Point (PEP), and a Policy Administration Point (PAP) – and may also contain policy repositories, as well as attribute authorities (or, more generally, Policy Information Points, PIP). In this terminology, the service is responsible for the PEP: only the service is able to enforce any decision. However, the service does not have to take the decision. The decision could be a purely automated by processing attributes against defined relevant policies, so it is possible to “factor out” the PDP. This would help services where the decision process is complex, or where additional information is required (which the service itself may not be authorized to access, but which could be accessed by a trusted PDP\textsuperscript{41}). As far as the management of attributes, the policies themselves are defined by the PAPs, although these would normally be defined by the communities. It follows that EUDAT should run services for the conversion of AAI tokens, as well as the authorization infrastructure, but suitable individuals from the communities should have access to the PAPs.

\textsuperscript{39} http://www.oasis-open.org/ XACML working group
\textsuperscript{40} OASIS XACML working group: access_control-xacml-2.0-saml-profile-spec-os
\textsuperscript{41} It also potentially enables the service to be run at a slightly lower security level, as only the PDP itself needs to run at a high security level.
3.4.2.1 Non-web clients

Simple store should be relatively straightforward: if the plan is to have a bespoke client run by the user, it will be necessary to integrate this client with the federation AAI. However, some care should be taken here, because such a client\textsuperscript{42} will not necessarily be able to follow the HTTP redirection steps required for implementing the client side of most types of AAI (notably, OpenID and Shibboleth). Interacting successfully requires both that the client can act as an HTTP client, including caching cookies and following redirections, but also that it presents HTML information to the user, e.g. a WAYF’s IdP selector. In other words, to successfully log in, a client will have to contain a web browser! It will also need interactions from the user to instruct the client to act upon the redirections, as well as to type the relevant IdP passwords.

This is clearly not desirable. In the cloud computing world, alternatives are being sought\textsuperscript{43}, but they are currently non-standardized and there is not yet consistency across providers of cloud services.

It follows then, that one of the following must be implemented.

\begin{itemize}
  \item Users must log into a portal, and by some alternative means, the client must obtain the credential the user has generated by logging into the portal – this is clearly not desirable as a client should be truly stand-alone.
  \item Clients could have restricted IdP selection services, e.g. a configured IdP, which redirects the user straight to the IdP, bypassing the normal AAI IdP selections (such as WAYF).
  \item Alternatively, clients could be configured with a more machine readable list of IdPs, generated from (Shibboleth) federation metadata, which can be refreshed regularly from known and fixed endpoints (albeit one per Shibboleth federation).
\end{itemize}

3.4.2.2 Data staging

The question that remains is how to integrate the architecture with the outside world. We have already seen that we make use of external infrastructures for AAI.

For the moment, we assume data is being transferred with GridFTP. In this case, three entities must have X.509 certificates (and private keys, and credentials): the source, the destination, and the controller (namely the user client). One or more of these could be the same. But, more importantly, the certificates must be signed by CAs that is trusted by each entity. This means that EUDAT cannot necessarily run its own CA and expect PRACE (say) to accept it\textsuperscript{44}. The proposal of the AAI TF is to secure the infrastructure itself with known and trusted certificates from NRENs, IGTF, and commercial CAs (particularly for browser-facing services) – these are all (usually) trusted by e- and cyber-infrastructures across the world. However, earlier we proposed to generate federation credentials on behalf of users, to be used internally in EUDAT. Such a service, if X.509 based, would most likely not be trusted outside of EUDAT (see footnote 44 on page 43). Consequently, data staging would have to be done by an automated service with a trusted certificate. However, it would also need to preserve, or at least respect, permissions and rights associated with the user.

\textsuperscript{42} The case is equivalent in general for command line login.

\textsuperscript{43} See e.g. the IETF SCIM working group. Moonshot (ABFAB) will also be a potential alternative.

\textsuperscript{44} That is not to say this is impossible: it can be done but it requires a lot of work (defining a CP/CPS, see RFC3647), buying a Hardware Signing Module for protection of an online CA; checking and validating the CA against the IGTF SLS profile, and seeking accreditation – the latter process can take 2-3 years – which is not compatible with the timescale of this project.
3.4.3 Detailed community requirements evaluations

From time to time these requirements may be extended and the priorities may be adjusted. Note that requirements may also be deleted, but the numbering will be preserved.

We list here a brief summary of the evaluations, without going into further details. Here, “ok?” means that the requirement is supported conditionally upon features outside the control of EUDAT (e.g., the types of attributes published by the identity providers.)

Table 8: Overview AAI requirements

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Description</th>
<th>Pri</th>
<th>Opt. 1</th>
<th>Opt. 2</th>
<th>Opt. 3</th>
<th>Opt. 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.a</td>
<td>Same username/password across all services</td>
<td>4</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>ok?</td>
</tr>
<tr>
<td>1.b</td>
<td>Single login in EUDAT</td>
<td>3</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>1.c</td>
<td>Single login (desktop)</td>
<td>3</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>No</td>
</tr>
<tr>
<td>2.a</td>
<td>Single registration</td>
<td>3</td>
<td>no</td>
<td>ok?</td>
<td>ok?</td>
<td>ok?</td>
</tr>
<tr>
<td>2.b</td>
<td>Single account (EUDAT/community)</td>
<td>3</td>
<td>no</td>
<td>ok?</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>2.c</td>
<td>Transparent registration</td>
<td>3</td>
<td>??</td>
<td>ok?</td>
<td>ok?</td>
<td>ok?</td>
</tr>
<tr>
<td>2.d</td>
<td>Deregistration of idle accounts</td>
<td>3</td>
<td>??</td>
<td>??</td>
<td>??</td>
<td>??</td>
</tr>
<tr>
<td>3.a</td>
<td>EUDAT view in community portal</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>Depends on community portal tech.</td>
</tr>
<tr>
<td>3.b</td>
<td>Command line login supported</td>
<td>3</td>
<td>no</td>
<td>ok?</td>
<td>no</td>
<td>No</td>
</tr>
<tr>
<td>4.a</td>
<td>Login with (trusted) OpenID</td>
<td>3</td>
<td>no</td>
<td>OK</td>
<td>OK</td>
<td>No</td>
</tr>
<tr>
<td>4.b</td>
<td>Log in with given SAML IdP</td>
<td>3</td>
<td>no</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>4.c</td>
<td>Support for “homeless” users</td>
<td>3</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>No</td>
</tr>
<tr>
<td>5.a.i</td>
<td>Use external attributes</td>
<td>3</td>
<td>no</td>
<td>ok?</td>
<td>no</td>
<td>OK</td>
</tr>
<tr>
<td>5.a.ii</td>
<td>Supplement missing attributes from internal AA</td>
<td>2</td>
<td>N/A</td>
<td>ok?</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>5.b.i</td>
<td>Attribute management</td>
<td>3</td>
<td>ok?</td>
<td>OK</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>5.b.ii</td>
<td>Attributes used for access control by services</td>
<td>4</td>
<td>OK</td>
<td>OK</td>
<td>no</td>
<td>OK</td>
</tr>
<tr>
<td>6.a</td>
<td>Distinguish level of assurance in authentication</td>
<td>3</td>
<td>N/A</td>
<td>ok?</td>
<td>ok?</td>
<td>no</td>
</tr>
<tr>
<td>7.a</td>
<td>Delegation of identity</td>
<td>3</td>
<td>no</td>
<td>OK</td>
<td>ok?</td>
<td>no</td>
</tr>
<tr>
<td>7.b</td>
<td>Delegation of authority</td>
<td>2</td>
<td>no</td>
<td>ok?</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

3.4.4 Overview of EUDAT services

Information in this section is still subject to evaluation and discussion, partly because of the need for further technical evaluations, and partly due to the ongoing nature of the work in other task forces. As such, the view is still relatively high level.

3.4.4.1 Web services

Support for RESTful web services generally does not work well with SAML. Authorization credentials need to be carried in the HTTP header (a GET has no body), and SAML assertions can exceed the limits on the length of the attribute values in the header. When using SAML for web services, one will normally use SOAP which can carry SAML assertions (message level).

RESTful web services work well with X.509 credentials (and indeed so does SOAP), but the mode is different as the authentication and authorization is done at the transport level (socket) and requires
access on the client side to the private key, to prove possession of the key. Socket level security is extremely well understood and very mature.

3.4.4.2 Data Staging

Data staging imposes the following requirements on the AAI.

- It needs to have EUDAT-federation credentials accepted across community (or non-EUDAT) resources as well as EUDAT. Identity mappings, and in particular permissions, need to be consistent.
- The EUDAT federated AAI must provide access to the replicated copies of the DOs.
- It should work with target data interfaces at non-EUDAT resources.
- It needs support for data transfer protocols (e.g. GridFTP, UFTP, or similar). GridFTP is currently known to be working with X.509 certificates. UFTP, as used by Unicore, should work with certificates. Support for other types of credentials (e.g. SAML assertions) is highly desirable.
- Data staging requires command line support.
- GridFTP, being the preferred protocol, suggests the use of X.509 certificates. Indeed, third party copying will normally require RFC3820 “proxies” (or the older versions, GT2 or GT3 – some implementations are sensitive to the version of the proxy).

3.4.4.3 Safe Replication

Data replication imposes the following requirements on the AAI.

- Consistent (federated) identities are needed.
- Delegation is necessary.
- Support for automated services (services which can authenticate themselves to other services without human intervention) is required.
- Consistent access control management across all replicas (as well as potentially other security attributes) is necessary.
- Community managers need to manage access control permissions.
- Data centres must be authenticated (host/service certificates).
- Access is logged – so persistent, unique, and non-reusable user ids are required. Traceability may be required.
- The technology must work with iRODS (also for automated services).

3.4.4.4 Simple Store

Simple Store imposes the following requirements on the AAI.

- Simple Store services must be authenticated.
- Each identity must be persistent, unique, and non-reusable. That ensures that data ownership is consistent across all services.
- Users must be able to set own permissions and access control to easily share data. Sharing may require user-discoverable identities, in particular with meaningful names45.
- Authentication must be user friendly (avoiding user-side X.509 key management).
- Simple Store may require traceability options.

45 Compare calling someone on Skype (say), where you do not have the person’s Skype username.
3.4.4.5 Metadata
Metadata service imposes the following requirements on the AAI.

- Community managers control attributes (that are used for authorization) of users that are members of communities (or who request such membership).

Metadata is thought to be openly accessible, so there are no confidentiality requirements, nor are there any access control requirements. However, there may be restrictions on modifying metadata and/or writing it.

3.4.4.6 Data security
In general, data security requirements (if there are any) may impose requirements on the AAI. For example, if integrity of data is very important, it may make sense to restrict who can update the data, which in turn may require a high level of assurance of the authentication. This is expected to be addressed in future work.

3.4.5 Pilot services and integration – mid-term view (Phase 2)
The AAI TF now needs to progress to integrating the pilot AAIs with the services. The choice of technology has not been settled. The AAI TF is currently restricted to making recommendations; the choice will be made in consultation with the SAF (in September 2012) and the pilot user communities\(^46\) (in October 2012). As portals are crucial to the *modus operandi* of the communities, the following points will have to be considered carefully in the integration and selection processes.

- The portal integration can be done in several ways, depending on the choice of technology. There is a trade-off, as the more complex solutions meet more requirements but depend on features that are still under development, or that are tailored to specific portals. The AAI TF would, of course, like to deliver the solution that meets most of the requirements, but also has to consider the risks associated with doing so.

- The communities themselves have already developed their own approaches to AAI, all of them different. In migrating communities to an EUDAT AAI, we have two extremes: have them start over and learn a new, and better, AAI, or work bug-for-bug with their existing infrastructure. The optimal approach is likely to be somewhere in-between. Having community users retain their current AAI *provided* that it is an external one (which implies that the chosen technology will have to support both OpenID (used by ENES) and Shibboleth (used by CLARIN). If the current community AAI is a community-specific username/password, it seems best not to allow them to retain it, as it means EUDAT would have to rely on a bespoke identity provider and/or manage passwords within EUDAT for users other than the homeless ones. Thus, we have the following recommendations.
  - If users use a federated identity (OpenID, or Shibboleth), they should be able to retain the use of this with EUDAT (with the usual caveat if their IdP only publishes ePTID).
  - If users do not use a federated AAI, they should be encouraged to migrate to one; and EUDAT will provide them with one (while not all users have Shibboleth, everyone can have an OpenID account – albeit with a lower level of assurance).
  - In addition, EUDAT may run an IdP for “homeless” users. Managing passwords for users should be the exception, though.

\(^46\) The two first pilot communities were CLARIN and ENES; this was later (August 2012) expanded to include EPOS and VPH.
It will be necessary for users to have a federation-unique identity, which again suggests that the use of OpenID and Shibboleth is a good idea (subject to Shibboleth providing the required attributes, ideally ePPN).

- From a technological point of view, it seems clear that X.509 certificates offer the best approach to federation-internal AAI. This is because certificates are very mature technology, almost everything we work with supports X.509 authentication, and we need host and service certificates anyway to secure and establish trust in the infrastructure itself.

- Managing authorizations is potentially more challenging for the following reasons.
  - We need to obtain attributes from users’ current attribute providers (which may or may not be the same as their IdPs).
  - Attributes have to be passed on to EUDAT resources which need to enforce the very same access control requirements that the users would have seen in their own communities.
  - It is clear that we should not re-implement the communities’ authorization schemes, otherwise we would have several different implementations for authorizations: one for each community (more or less). It seems more sensible to provide something which (a) satisfies the community requirements, (b) is functionally equivalent or better than their current approach where possible, and (c) fits, whenever possible, with established community practices. For example, CLARIN currently has the access control decision taken by the portal but enforced by the resource. This is not optimal, since it requires that users re-request authorization for other resources whenever they need to access something else. It would be better to have the portal perform the authentication and gather the relevant attributes for the user, and have the services make the access control decisions. Following an XACML model\(^47\), the access control decision workflow can be centralized (federation) or decentralized (with the site/service/community administrators) as needed.
  - Whenever communities share data or otherwise collaborate, a common model for authorization will have to be developed anyway. It is possible that EUDAT will need to provide attributes to enable authorization for sharing data between communities.
  - Attribute naming is an issue; we may need a translation service which renames attributes from those understood in one community to those of another\(^48\). Another approach is to make attributes unique by prefixing them with a namespace (like a URN, say). Another method is to look at the different roles within each community and identify commonalities, i.e., attributes which genuinely reflect the same role in each community, and where all the communities participating in EUDAT can agree on the interpretation of the attribute. Having agreement between only two communities for bilateral collaborations is insufficient unless they agree on a namespace (URN) to make it unique to their collaboration.

### 3.4.6 AAI services and integration – long-term view (Phase 2 & 3)

The long-term integration depends, of course, on the results of the work in the medium-term – unless we were to migrate fully to a new infrastructure. This would only be the case if a wholly revolutionary type of technology arrived and matured within the lifetime of the project – which is unlikely, but not impossible, so we are tracking developments in the area.

---

\(^47\) XACML authorisation workflow contains entities known as PAP, PIP, PDP, PEP, each with particular roles. As these are notionally distinct entities, they can be run at federation level or devolved as required.

\(^48\) This has been a problem in ENES (or rather in ESGF).
Work being tracked in this area includes the European Federated Identity Management collaboration (FIM), Open Grid Forum (FEDSEC, IDEL, and VOMSPROC), and related cloud security and identity management activities (IETF or OGF).

The short-term problem is partly the variable maturity of implementations, and partly that no AAI satisfies all our requirements. Hence EUDAT will probably need to invest work in addressing the lack of maturity and/or features in the AAI, and in integrating the AAI with the services.

A lack of support for features could also be combated by mixing and matching technologies, although if this approach were adopted, then one should be careful that it does not result in even more work. For example, if every service must understand two (or more) different technologies, a lot more work on adaption of existing services may have to be done. A better approach may be credential conversion (as in Options 2 and 3) where a credential is “converted” into another by a trusted service. In this case, one should be careful that the trust issues with the core conversion services (at federation level) are well understood.

3.4.7 Implementation plan

The AAI task force has defined the following implementation plan and timeline to provide a federated AAI solution to enable access to the EUDAT services.

- Current: Pilot AAIs running. Evaluations completed. May need to upgrade some services as they develop.
- 2012-09: Presentation of AAI planning and progress to SAF.
- 2012-10: Integration of one or more pilot AAI with community portals. Demonstration of access to portal with Shibboleth or OpenID (or, ideally, both). Support for at least CLARIN and ENES (login).
- 2012-12: Support for EPOS and VPH (login).
- 2012-12: Demonstrate data staging.
- 2013-02: EUDAT/community integration in portals (account, attributes).
- 2013-02: Integration of community attributes with EUDAT.
- 2013-04: Toolkits/instructions for all communities.
- 2013-05: Safe replication and simple store integrated with AAI for all communities.

3.4.8 Technical evaluations

In brief, there are four distinct possibilities, all of which are currently available for use as pilot services.

1. CROWD was used for EUDAT staff, to ensure that an easy-to-use single sign-on mechanism was available as soon as possible to access all collaborative services. The IT support at CSC worked on the integration of CROWD with JIRA, Confluence, SVN, etc. in what the AAI TF termed phase 1. Like some of the other AAIs mentioned later, CROWD is based on SAML2.

The CROWD service is being run in production by CSC. It has two interesting features which make it useful as a pilot SSO: first, the login page offers a selection between Haka, the Finnish identity management federation, and a CSC-internal SAML2-based identity provider. This means that either a user provides a federated identity (of interest only to the Finnish collaborators in EUDAT), or users that are “homeless” from the point of view of Haka are offered logins (via the second provider). Secondly, the service is based on SAML2. However, it currently supports only collaborators in projects supported by CSC, e.g., the EUDAT collaborators.

2. Another option is to reuse the AAI from another EU-funded project, Contrail – this would solve most of our problems as it should be able to consume Shibboleth and OpenID logins, as well as
provide “homeless” access. Attributes can be provided by an internal attribute authority. There is provision for attribute-based access control. The AAI supports both portal and command line login, but currently command line is available only with internal identities. The federated identity is based on generating X.509 certificates internally (but in a way that is hidden from the user). While this option meets nearly all of our requirements on paper, the work on developing all the features is still in progress.

FZJ runs a pilot installation of this service and have experimented with customizations of the portal, which will be essential for future integration work.

3. Use SimpleSAMLPhp (in option 1) as a front end to a portal, which then uses a portal certificate to control access to services. While the portal certificate cannot be individual to the user, as in option 1, it can be tailored to each community.

STFC runs a pilot installation of this service. It is useful also for evaluating identity provider selectors, e.g. DiscoJuice. Excepting the pilot, there has been no direct use of SimpleSAMLPhp for portal development. However, communities have developed portals themselves (outside of EUDAT) which have used either OpenID or Shibboleth to authenticate to the portals, and then used a fixed portal certificate to access community services. Adapting such a portal to use SimpleSAMLPhp would most likely be relatively easy.

4. Use Shibboleth only. Shibboleth is quite mature technology, and having Shibboleth access to services generally requires joining each service into the existing Shibboleth federations. This would be a major task in itself, but the work being done by eduGain would simplify this.

A pilot installation of Shibboleth should not be necessary, as we should be able to use existing IdPs (provided we integrate the service with the federation). However, there are test IdPs we can also use, for example the Feide test IdP.

5. Shibboleth is widely used with portals. BSC worked with Shibboleth authentication for iRODS, and CSC already has it working (and STFC had it in the past). However, these Shibboleth implementations generally require that users use web browsers to access the services, to enable the federation redirections and the user interactions (IdP selection).

3.5 Joint Metadata Domain

The “Joint Metadata Domain”, a service case within the EUDAT project, is based on a metadata task force position paper, summarized in section 2.5.3 of the D5.1.1 deliverable (see footnote 2 on page 8) followed by the metadata task force work plan described in this section. This service is meant as an opportunity for EUDAT to provide metadata (MD) services for cross-disciplinary use of resources. This may provide a good alternative for young or small communities that cannot afford to create their own metadata service providers.

3.5.1 Architecture Overview

As in the EUDAT communities the currently most frequently used infrastructure to provide metadata services is the harvesting model, harvesting metadata according to the OAI-PMH protocol will be a main feature of the architecture of the Joint Metadata Domain. In this model every community repository has one (or a community central) metadata provider and allows its metadata to be harvested

49 http://www.openarchives.org/OAI/openarchivesprotocol.html
by one or more central metadata service providers. The EUDAT metadata service will offer basic metadata search and browsing services to researchers looking for or exploring the resources from other disciplines.

With respect to the type of metadata and the involvement of the communities we will harvest metadata from the following types of communities:

1. Core communities providing XML type metadata through an OAI-PMH component.
2. Non-core communities providing XML type metadata through an OAI-PMH component.
3. Core communities providing other type of metadata that has to be harvested by other means.

![Figure 28: Global overview metadata architecture](image)

The metadata from all three methods will be inserted into the EUDAT Metadata Xml database (EMX). As far as the communities are able to provide their MD already in xml format, OAI-PMH servers will be installed. For the remaining communities, the proceeding depends on the structures of their meta databases. In these cases, EUDAT will offer MD ingestion services to get the metadata into the EMX.

However, priority will be given implementing a prototype using the OAI-PMH metadata providers that are available and then expanding with non-core communities that offer usable metadata of good quality. The reason is that the technology for harvesting with OAI-PMH is well known and many EUDAT partners have experience with this method. An OAI-PMH harvester will be installed close to the EMX database.

For the implementation of the EUDAT metadata catalog and search functionality we will also use existing familiar technology. It could be that this will not scale in the long term, but we will monitor this and investigate using more scalable technologies where needed. We also encourage other partners to test/use different technocnologies in parallel where feasible. Presently, priority is given to deploying usable services in the short term. Considering the above we propose to test using in combination to the EMX database Lucene\textsuperscript{50} indexes and a SOLR\textsuperscript{51} faceted browser as the first step.

\textsuperscript{50} http://lucene.apache.org/core/
\textsuperscript{51} http://lucene.apache.org/solr/
In a later stage of the project, the collected xml MD of very inhomogeneous structures will be prepared for more homogeneous access by implementing of a natural mapping for the most common concepts. The aim is to allow for search and browsing on equalized MD of different communities at the same time.

3.5.2 Implementation Plan

The implementation workload in the EUDAT MD-TF is subdivided into the following three branches A, B, and C.

Work branch A – harvesting infrastructure, XML DB (EMX), search:

1. basic harvesting of metadata from ENES, CLARIN storing it in a BaseX XML database providing keyword search.
2. Including simple mapping of metadata elements from different schemas (minimal 10, including geographic and time coverage) and offer a SOLR/Lucene faceted browser
3. Add a commenting function to the metadata catalog
4. Complete the user Interface (UI) and provide a flexible mapping mechanism between metadata concepts that will allow easy inclusion of new schema. Include non-OAI-PMH metadata harvesting

Next to the work on the harvesting side, there will be two tasks working at the metadata provider side.

Work branch B – MD schemas, quality checks, community feedback, interdisciplinarity:

1. Make an inventory and registry for the metadata schemas in our (core) communities starting with ENES and CLARIN include metadata concept descriptions or ontologies.
2. Use existing simple OAI-PMH harvesters and service providers to check the quality of the metadata providers and give feedback. CLARIN has a solution, the Virtual Language Observatory\(^5\) (VLO). Setup a feedback and reporting infrastructure with the communities.
3. Contact and support emerging communities to set-up OAI metadata providers.
4. Investigate the usability of existing national initiatives to come to national joint metadata domains for interdisciplinary metadata.

There will also be further technology appraisals of existing metadata catalogue and search systems, especially those in use by the EUDAT communities with regular sanity checks of the systems used in Branch A. Although this work is not critical to the delivery of a first prototype, it is critical for further development and is also necessary to give advance warning if any prototype can be scaled up to larger amounts of data.

Work branch C – harvesting strategies, performance evaluation:

1. appraisal of metadata services technology
   a. consider the different harvesting strategies:
      (1) flat: one aggregator harvesting individual metadata providers.
      (2) hierarchical: aggregators harvesting other aggregators. Points to consider are (a) curation strategies at different aggregator levels and (b) possibile scalability problems with these two models.
   b. suggest & test possible technologies to use for the prototypes
2. evaluating performance of the prototypes

The following timeline is envisioned to implement the full implementation plan.

\(^5\) http://www.clarin.eu/vlo/
Table 9: Timeline Branch A: the harvester and catalogue

<table>
<thead>
<tr>
<th>Phase</th>
<th>Work Items</th>
<th>Milestone</th>
<th>WP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prototype 1</td>
<td>• Metadata from ENES &amp; CLARIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Simple keyword search</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Inspection metadata record</td>
<td>M16</td>
<td></td>
</tr>
<tr>
<td>Prototype 2</td>
<td>• Include flexible semantic mapping facility</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Social commenting function</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Include more communities metadata</td>
<td>M24</td>
<td></td>
</tr>
<tr>
<td>Prototype 3</td>
<td>• Open linked data export</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Error reporting</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Non-XML / non-OAI metadata inclusion</td>
<td>M30</td>
<td></td>
</tr>
</tbody>
</table>

Table 10: Timeline Branch B: Assess metadata schemas and their semantics

<table>
<thead>
<tr>
<th>Phase</th>
<th>Work Items</th>
<th>Milestone</th>
<th>WP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core communities</td>
<td>• Inventarize metadata providers/aggregators</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Inventarize metadata schema</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Analyse schema and register core concepts</td>
<td>M15</td>
<td></td>
</tr>
<tr>
<td>Expansion 1</td>
<td>• Include other EUDAT communities and/or willing well organized externals</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expansion 2</td>
<td>• Actively solicit new communities to join</td>
<td>M24</td>
<td></td>
</tr>
</tbody>
</table>

Table 11: Timeline Branch C: Technology appraisal

<table>
<thead>
<tr>
<th>Phase</th>
<th>Work Items</th>
<th>Milestone</th>
<th>WP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>• Investigate and review the architecture of existing projects/systems using OAI-PHM (flat, hierarchical or other strategies)</td>
<td>M16</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>• Consult with partners about the design of test for evaluating different harvesting strategy using OAI-PHM</td>
<td>M16</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>• Conduct the test if applicable</td>
<td>M18</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>• Investigate alternative harvesting technology (e.g. web/XML crawler, publish as linked data)</td>
<td>M24</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>• Investigate alternative scalable indexing and searching technology (e.g. Hadoop + Apache Solr, Solr Cloud, Sindice if in Open Linked data)</td>
<td>M30</td>
<td></td>
</tr>
</tbody>
</table>

3.6 Simple Store

The goal of the Simple Store task force is to evaluate technologies and establish a pilot service that will provide an easy way for researchers to upload data with minimal effort. All data uploaded through the service should be open, publicly available and have proper metadata associated. The Simple Store service will address the long tail of “small” data and the researchers/citizen scientists creating/manipulating them and not the short head of big data that is served by other EUDAT services.
3.6.1 Work plan

The work within this task force is organized along five different phases:

- Scouting: to investigate and evaluate potential technologies to implement defined requirements;
- Design: to design the final service, defining a reference architectural model and most important a simple user interface offering simple functionality;
- Implementation: to implement the final service integrating and developing missing features;
- Pre-production: to test and evaluate the correctness and the reliability of the service;

Experts from different work packages will be involved throughout the course of the phases.

3.6.2 Status and Initial results

Since it’s creation in May 2012 the task force has, in collaboration with interested user communities from the DASISH cluster, ECRIN, VPH, LifeWatch, and EPOS, established a set of evaluation criteria and performed a paper evaluation of a number of promising technologies against these criteria. The initial set of evaluation criteria included:

- Basic technology,
- Object store methodologies,
- WebDav frontend,
- Scalability and capability,
- AAI schemes supported,
- License model,
- Deployment status,
- Deployment help available,
- Type of objects supported,
- Object identification mechanisms,
- Right and quota management,
- Metadata scheme and support for incremental metadata,
- Support for crowd sourcing,
- Server-side triggers,
- Data export mechanisms and
- Possible integration in EUDAT’s replication scenario, support for interlinking object, archiving support.
The technologies evaluated included Scratchpads\(^{53}\), VirtualCloudDrive\(^{54}\), BeeHub\(^{55}\), Figshare\(^{56}\), iDrop\(^{57}\), myExperiment\(^{58}\), and INVENIO\(^{59}\).

The paper evaluation of these technologies resulted in the decision to perform further hands-on evaluations of INVENIO, with BeeHub and myExperiment as backup solutions. The complete evaluation can be found in Annex A.

### 3.6.3 Evaluation process

To perform the hands on evaluation of INVENIO a test instance has been installed at SNIC/KTH and made available to the task force members and interesting user communities. Users are expected to perform a simple sequence of tests including

1. Store new data objects of various types (at least .pdf, binary, movies, excel, ...);
2. Retrieve metadata;
3. Add additional metadata;
4. Retrieve data objects based on metadata search;
5. Change access control settings and test them;
6. Tag an object and have an online discussion on it; and
7. Evaluate INVENIO against the criteria mentioned above.

In parallel, further evaluations of the base technology are performed at EPCC and KTH, particularly exploring possible integration with other EUDAT technologies.

The initial evaluation phase is expected to last until the end of October 2012 when a decision on the further use of the technology will be taken. This will be followed by a thorough gap analysis, the commissioning of development efforts to close these gaps, and the development of an initial deployment and operation model for the pilot phase.

---

\(^{53}\) http://scratchpads.eu/
\(^{54}\) https://github.com/VirtualCloudDrive/CloudDrive/wiki
\(^{55}\) https://www.beehub.nl/
\(^{56}\) http://figshare.com/
\(^{57}\) https://code.renci.org/gf/project/irodsidrop/idrop-swing
\(^{58}\) http://www.myexperiment.org/
\(^{59}\) http://invenio-software.org/
4 CONCLUSIONS

The selection of the type and prioritization of services that EUDAT should provide is the result of the community interviews that were conducted very early on in the project. The results of these are reported in deliverable D4.1.1 Management Landscape Characteristics and Community Requirements. The first services that were selected for implementation are: Safe Replication, Data Staging, Joint Metadata Domain, Simple Store and Persistent Identifiers. Having decided upon these, the task of appraising appropriate technology started, which resulted in the selection of technologies upon which the candidate services are being built. This deliverable gives an overview of these first services EUDAT that has been working in the first year of the project, and provides an in-depth view of the technical aspects of these services. The Joint Metadata Domain and Simple Store services are not included in this deliverable because they are in the preliminary phases of defining the work plan or appraising suitable technologies.

4.1 Task forces and the service-building process

The use of task forces has been adopted for organizing the work to build the services. In the Safe Replication and Data Staging task forces, this includes the handover to operations. In the Joint Metadata Domain and Simple Store task forces, the work undertaken by the task forces has been extended to include determining the community requirements, performing technology appraisals and even some collaboration with JRA activities. The task forces have become the main vehicle for organizing the bulk of the work across the EUDAT work packages by focusing on specific tasks or services. An envisaged risk in this approach is that it is possible to become too focused on the work within a particular task force and to lose sight of related work within other task forces, for example between AAI and Data Staging. To alleviate this risk, a closer collaboration between the task forces is needed. In the case of the AAI task force, additional AAI aspects and requirements that are discovered within the other tasks forces are taken into account, along with the technologies that are chosen by other task forces. These cross-task force collaborations should be extended more in the future.

4.2 Service building status and next steps

All the task forces have made good progress in testing and evaluating the services. Due to close collaboration with the technical people in the communities, who are fully cognisant of the community data services, AAI and policies, EUDAT is assured that the services that are provided full fill the needs of the communities to a large extend, and that these services fit into the community infrastructures and are useful to the communities.

4.2.1 Safe Replication

The Safe Replication services are based on the use of the policy engine iRODS and various micro-services. In the different community islands, a total of seven partner sites (e.g. community and service provider data centres) have set up pilot iRODS instances to test cross-site replication. After solving some network issues (e.g. opening ports within a firewall), the basic tests that were conducted were successful. Some performance issues were found which proved to be due to some configurations and the calculation of checksums. After some optimizations, the performance has been improved. The next steps are to move the pilot iRODS instances to production systems, which are integrated both with the mass storage systems of the partners, and with local and EUDAT operations.

After setting up the basic core services, the task force has focused on building the PID micro-service to automatically register PIDs to DOs on ingest and to keep track of replicas. Part of the PID building process has included discussions on the replication policies, because these influenced both the...
registration of the replicas and linking the replicas to the original DO. This issue lead to the use of the RoR field which always points back to the original DO. The PID micro-service has been implemented and has been successfully demonstrated in the CLARIN island. The next step is to extend the tests to the other islands.

For the registration of PIDs, EUDAT is going to use EPIC handle service, and hence EUDAT is negotiating with the EPIC consortium to develop an MoU.

In its initial phase, the Safe Replication service is considered to be a data management tool that enables community data managers to define and apply the community replication policies. Because the number of users requiring direct access is limited and user level access is still provided via community specific services and portals, this phase of the Safe Replication does not depend on a federated AAI solution.

It is expected that the safe replication candidate service will be handed over to the operations team in M12 of the EUDAT project (that is, September 2012).

4.2.2 Data Staging

The Data Staging service works on top of the EUDAT storage services (e.g. iRODS) provided by the Safe Replication service. Due to the nature of the different communities requesting this type of service (namely VPH, EPOS and ENES), different solutions have been evaluated and demonstrated. These solutions ranged from basic iRODS tools and third party transfers via GridFTP, to higher level tools such as Globus On-Line and the XSEDE file manager. To make it possible to access GridFTP from iRODS, iRODS was extended with the Griffin GridFTP server.

The Data Staging service is regarded as a user-level service that allows individual users to easily manage the staging of large amounts of data (that consists of many files) and that results in many file transfers. Consequently the Data Staging service depends on the services developed by the AAI task force to provide federated access to the EUDAT data services. The time plan of the AAI TF, does not predict that a federated AAI solution will be available in the short term. Because the data staging solutions are all based on standard authentication methods (e.g. X509, username/password), it is possible to provide access in a less federated way (e.g. separate EUDAT or local credentials). As well as continuing to work on the selected technologies, the Data Staging TF will focus on developing a practical solution to enable user access during the coming months.

During the service-building process, the Data Staging task force collaborated with the following organisations.

- **PRACE**
  The communities actively participating in the Data Staging task force (VPH and EPOS) also use the PRACE HPC computing facilities. To provide an optimal solution, the EUDAT data service should be accessible from the PRACE supercomputers. At the moment, EUDAT and PRACE are holding ongoing strategic discussions for the purpose of collaborating. At time of writing, EUDAT and PRACE had agreed to conduct some pilots.

- **XSEDE**
  XSEDE is the successor of the TeraGrid project in the US, and is funded by the National Science Foundation. The XSEDE organization develops and maintains the XSEDE file manager. The XSEDE file manager is the tool to use for moving large amounts of data between XSEDE supercomputers. EUDAT has collaborated with XSEDE to adopt the XSEDE file manager to make it work with the EUDAT data services.
4.2.3 Federated AAI

Federated AAI is considered to be a service that makes it possible to access EUDAT services while maintaining user identities, roles, access rights, trust and privacy. The implementation plan of the AAI TF consists of three phases: short-, mid- and long-term. The short-term phase, which makes collaborative tools that span three countries (such as the EUDAT wiki, the JIRA ticket system, the EUDAT central-monitoring system and the trouble ticket system) available to project members via SSO, has been completed.

Most of the work has gone into the mid-term solution. This involved getting detailed requirements from the communities, developing the overall approach, using credential conversion technologies, evaluating and piloting different technologies, and assessing the community portals. The candidate technology that fulfils most of the requirements is the credential conversion software being developed in the Contrail project. However this software is complex and it is still under development, rather than being completed. There are other candidate technologies (such Shibboleth and SimpleSAMLPhp) that only target web-based access, and omit command-line access. The current status is that work in the task force is very much in progress.

Currently there is a lot of activity in the scientific domain in relation to the topic of Federated AAI, or identity management. Federated AAI is considered to be a complex issue, not only from the technical point of view but also in relation to policies on identity management, attribute release and automated registration and deregistration. As EUDAT is a multi-disciplinary scientific project, spanning 13 countries across Europe, it makes Federated AAI a very challenging and demanding for the project. EUDAT is actively engaged in other initiatives in the area of federated identity management, and has:

- Attended FIM4R\(^{60}\) workshops,
- Provided input to the AAA Study group\(^{61}\) report\(^{62}\), description of the EUDAT AAI use case, and
- Presented the EUDAT AAI use case at the Terena VAMP\(^{63}\) meeting.

It is important for EUDAT to continue to participate in the arena of federated identity management, both in Europe and globally.

4.2.4 Joint Metadata Domain

The metadata task force is in the process of defining the workplan. There have been first discussions on the task division between the MD task force members. However, there needs to be further evaluation by the MD task forcemembers and perhaps EUDAT management on this topic. The meta data task force is at the start of the service building process.

4.2.5 Simple Store

The Simple Store task force is currently in the evaluation phase. The initial evaluation phase is expected to last until the end of October 2012 when a decision on the further use of the technology will be taken. This will be followed by a thorough gap analysis, the commissioning of development efforts to close these gaps, and the development of an initial deployment and operation model for the pilot phase.

---

\(^{60}\) Federated Identity Management For Research: [https://cdsweb.cern.ch/record/1442597](https://cdsweb.cern.ch/record/1442597)

\(^{61}\) AAA Study group home page: [https://confluence.terena.org/display/aaastudy/AAA+Study+Home+Page](https://confluence.terena.org/display/aaastudy/AAA+Study+Home+Page)


4.3 Handover and task force termination

Overall, the task forces have made good progress in building the first candidate services. The expectation is that, in the coming months, the first services can be handed over to the operation team and be used by the communities in a pre-production mode, while the services are operated in production by the service providers. When a service is handed over to the operation team, the task force for that particular service will terminate. If it proves necessary, a follow-up task force may be defined to continue work on future extensions of the service. The task forces do not have any formal role in the operation of the services.
## ANNEX A. EVALUATION SIMPLE STORE TECHNOLOGIES

<table>
<thead>
<tr>
<th>Basic Technology</th>
<th>Scratchpads</th>
<th>Virtual CloudDrive</th>
<th>BeeHub</th>
<th>Figshare</th>
<th>iDrop</th>
<th>myExperiment</th>
<th>Invenio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>Based on Drupal Content Management System with additional modules.</td>
<td>Build in Java and Scala</td>
<td>Web service based on PHP, Javascript, MySQL, ExtJS</td>
<td>Unknown, closed source, APIs available in Python, PHP and Ruby</td>
<td>Java</td>
<td>Ruby, Rails 2, Java, PHP, MySQL, 4store, Linked Data, XML, RDF, RESTful API, GoogleAPS for mail server, Apache Solr Search Server, RubyForge, MediaWiki, JIRA</td>
<td>Python, MySQL</td>
</tr>
<tr>
<td>Object Store</td>
<td>Local file system</td>
<td>Local file system or Amazon S3</td>
<td>Local file system</td>
<td>Amazon S3</td>
<td>iRODS storage</td>
<td>MySQL with a cache in a local file system</td>
<td>Local file system</td>
</tr>
<tr>
<td>WebDav frontend</td>
<td>Yes, via a module</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>AAI</td>
<td>username/password, openID and Shibboleth modules available</td>
<td>SimpleSaml PHP</td>
<td>HTTP Digest, HTTPS Basic, SAML2, X.509 (extendible)</td>
<td>Username/password</td>
<td>HTTP Digest, HTTPS Basic, GSI</td>
<td>Username/password, OpenID, HTTP basic, OAuth.</td>
<td>RBAC based authorization, Hooks for supporting external login methods are present</td>
</tr>
<tr>
<td>License</td>
<td>GNU GPL</td>
<td>Open source</td>
<td>Open source</td>
<td>Closed source, free to use</td>
<td>Open source</td>
<td>Open source</td>
<td>Open source</td>
</tr>
<tr>
<td>type of objects</td>
<td>Images, the administrator can allow to upload any kind of content</td>
<td>Any</td>
<td>Any</td>
<td>Documents, images, graphs, figures, video, datasets, etc.</td>
<td>Any</td>
<td>Any, workflows are preferred (Taverna being the most supported)</td>
<td>Documents, collections and trees of collections.</td>
</tr>
<tr>
<td>Identification</td>
<td>URL</td>
<td>URL, not usable as external links</td>
<td>No, PID support is on feature request list</td>
<td>DOI</td>
<td>Metadata, Handles</td>
<td>URI</td>
<td>URL-like</td>
</tr>
<tr>
<td>Rights</td>
<td>Limited, default all data is public, private directories with extended permissions are possible</td>
<td>Limited, users can share their data (or kept them private), no sophisticated (ACL-like) rights management</td>
<td>Yes, extensive ACL capabilities. User can easily set ACLs per user or group</td>
<td>Limited, data in Figshare is limited to being public or private</td>
<td>Yes, user can easily set ACLs and meta data</td>
<td>Yes, the uploader can define which rights are applicable.</td>
<td>Yes, fine grained RBAC system.</td>
</tr>
<tr>
<td>Quota</td>
<td>Yes, define quotas per user, directory, file</td>
<td>Yes, define quotas per user</td>
<td>Yes, soft quotas have been implemented</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>metadata scheme</td>
<td>Yes, can be added through modules</td>
<td>Yes</td>
<td>Yes, free form metadata</td>
<td>No</td>
<td>Yes</td>
<td>Yes, various formats</td>
<td>Yes</td>
</tr>
<tr>
<td>Feature</td>
<td>Scratchpads</td>
<td>Virtual CloudDrive</td>
<td>BeeHub</td>
<td>Figshare</td>
<td>iDrop</td>
<td>myExperiment</td>
<td>Invenio</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------</td>
<td>--------------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------</td>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>incremental metadata</td>
<td></td>
<td>NA</td>
<td>NA</td>
<td></td>
<td></td>
<td>No, is on the roadmap</td>
<td>Yes</td>
</tr>
<tr>
<td>crowd sourcing</td>
<td>Unknown</td>
<td>Limited, only access public data</td>
<td>No</td>
<td>?</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Export</td>
<td>Limited, on from storage backend</td>
<td>No</td>
<td>Yes</td>
<td>?</td>
<td>Yes</td>
<td>Limited, only via certain formats.</td>
<td>Limited, via custom exporters</td>
</tr>
<tr>
<td>archiving</td>
<td>Yes, via storage backend</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes, via iRODS</td>
<td>No</td>
<td>Yes, archiving solution supporting the OAIS model</td>
</tr>
</tbody>
</table>
# ANNEX B. GLOSSARY

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA</td>
<td>Authentication, Authorisation and Accounting</td>
</tr>
<tr>
<td>AAI</td>
<td>Authentication and Authorization Infrastructure</td>
</tr>
<tr>
<td>ADMIRE</td>
<td>ADMIRE (Advanced Data Mining and Integration Research for Europe) is a project co-funded by EU within the FP7</td>
</tr>
<tr>
<td>APA</td>
<td>Alliance for Permanent Access</td>
</tr>
<tr>
<td>APARSEN</td>
<td>APARSEN (Alliance for Permanent Access to the Records of Science in Europe Network) is a project co-funded by EU within the FP7</td>
</tr>
<tr>
<td>API</td>
<td>Application programming interface</td>
</tr>
<tr>
<td>ARC</td>
<td>Advanced Resource Connector</td>
</tr>
<tr>
<td>AtP</td>
<td>Attribute Provider</td>
</tr>
<tr>
<td>CA</td>
<td>Certificate Authority</td>
</tr>
<tr>
<td>CASPAR</td>
<td>CASPAR (Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval) is an Integrated Project co-funded by the EU within the FP6</td>
</tr>
<tr>
<td>CDI</td>
<td>EUDAT Collaborative Data Infrastructure</td>
</tr>
<tr>
<td>CDO</td>
<td>Climate Data Operators</td>
</tr>
<tr>
<td>CERT</td>
<td>Computer Emergency Response Team</td>
</tr>
<tr>
<td>CESSDA</td>
<td>Preparatory phase project for a major upgrade of the Council of European Social Science Data Archives. An ESFRI project in the Social Sciences and Humanities domain.</td>
</tr>
<tr>
<td>CIT</td>
<td>Community Integration Toolkit</td>
</tr>
<tr>
<td>CLARIN</td>
<td>Common Language Resources and technology Initiative. An ESFRI project in the Social Sciences and Humanities domain.</td>
</tr>
<tr>
<td>CMS</td>
<td>Content Management System</td>
</tr>
<tr>
<td>CNRI</td>
<td>Corporation for National Research Initiatives</td>
</tr>
<tr>
<td>CODATA</td>
<td>International Council for Science: Committee on Data for Science and Technology</td>
</tr>
<tr>
<td>COPAL</td>
<td>Heavy Payload Long endurance Tropospheric Aircraft. An ESFRI project in the Environmental Sciences domain.</td>
</tr>
<tr>
<td>CSMD</td>
<td>Core Scientific Metadata Model</td>
</tr>
<tr>
<td>CTA</td>
<td>Cherenkov Telescope Array. An ESFRI project in the Physical Sciences and Engineering domain.</td>
</tr>
<tr>
<td>Curation</td>
<td>Provision of domain-dependent contextual support for permanent access to the meaning of data – including metadata, lexica, etc</td>
</tr>
<tr>
<td><strong>Curation and Preservation</strong></td>
<td>The process of ensuring that data can be re-used over time.</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td><strong>D4Science</strong></td>
<td>Data Infrastructures Ecosystem for Science</td>
</tr>
<tr>
<td><strong>DAITF</strong></td>
<td>Data Access and Interoperability Task Force</td>
</tr>
<tr>
<td><strong>DARIAH</strong></td>
<td>Digital Research Infrastructure for the Arts and Humanities. An ESFRI project in the Social Sciences and Humanities domain.</td>
</tr>
<tr>
<td><strong>DB</strong></td>
<td>Database</td>
</tr>
<tr>
<td><strong>DECI</strong></td>
<td>DEISA Extreme Computing Initiative</td>
</tr>
<tr>
<td><strong>DEISA</strong></td>
<td>Distributed European infrastructure for supercomputing applications</td>
</tr>
<tr>
<td><strong>DG</strong></td>
<td>Directorate General</td>
</tr>
<tr>
<td><strong>DN</strong></td>
<td>Distinguished Name</td>
</tr>
<tr>
<td><strong>DO</strong></td>
<td>Digital Object</td>
</tr>
<tr>
<td><strong>DoW</strong></td>
<td>Description of Work</td>
</tr>
<tr>
<td><strong>DR</strong></td>
<td>Data Replication service</td>
</tr>
<tr>
<td><strong>EB</strong></td>
<td>EUDAT Executive Board</td>
</tr>
<tr>
<td><strong>EBI</strong></td>
<td>European Bioinformatics Institute</td>
</tr>
<tr>
<td><strong>EBS</strong></td>
<td>Amazon Elastic Block Store</td>
</tr>
<tr>
<td><strong>EC</strong></td>
<td>European Commission</td>
</tr>
<tr>
<td><strong>ECCSEL</strong></td>
<td>European Carbon Dioxide Capture and Storage Laboratory Infrastructure. An ESFRI project in the Energy domain.</td>
</tr>
<tr>
<td><strong>ECGA</strong></td>
<td>EC Grant Agreement</td>
</tr>
<tr>
<td><strong>EEF</strong></td>
<td>The European e-Infrastructure Forum</td>
</tr>
<tr>
<td><strong>E-ELT</strong></td>
<td>European Extremely Large Telescope. An ESFRI project in the Physical Sciences and Engineering domain.</td>
</tr>
<tr>
<td><strong>EESI</strong></td>
<td>European Exascale and Software Initiative</td>
</tr>
<tr>
<td><strong>EGEE</strong></td>
<td>Enabling Grids for E-sciencE</td>
</tr>
<tr>
<td><strong>EGI</strong></td>
<td>European Grid Initiative</td>
</tr>
<tr>
<td><strong>EIDA</strong></td>
<td>European Integrated Data Archive</td>
</tr>
<tr>
<td><strong>e-IRG</strong></td>
<td>e-Infrastructure Reflection Group</td>
</tr>
<tr>
<td><strong>EISCAT_3D</strong></td>
<td>The next generation European incoherent scatter radar system. An ESFRI project in the Environmental Sciences domain.</td>
</tr>
<tr>
<td><strong>ELI</strong></td>
<td>Extreme Light Infrastructure. An ESFRI project in the Physical Sciences and Engineering domain.</td>
</tr>
<tr>
<td><strong>ELIXIR</strong></td>
<td>European Life sciences Infrastructure for Biological Information. An ESFRI project in the Biological and Medical Sciences domain.</td>
</tr>
<tr>
<td>Acronym</td>
<td>Full Form</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
</tr>
<tr>
<td>EMI</td>
<td>European Middleware Initiative</td>
</tr>
<tr>
<td>EMSO</td>
<td>European Multidisciplinary Seafloor Observatory. An ESFRI project in the Environmental Sciences domain.</td>
</tr>
<tr>
<td>EMX</td>
<td>EUDAT Metadata XML database</td>
</tr>
<tr>
<td>ENES</td>
<td>European Network for Earth System Modelling</td>
</tr>
<tr>
<td>EPIC</td>
<td>European Persistent Identifier Consortium</td>
</tr>
<tr>
<td>EPOS</td>
<td>European Plate Observing System. An ESFRI project in the Environmental Sciences domain.</td>
</tr>
<tr>
<td>ePPN</td>
<td>eduPersonPrincipalName</td>
</tr>
<tr>
<td>ePTID</td>
<td>eduPersonTargetedID</td>
</tr>
<tr>
<td>FIM</td>
<td>Federated Identity Management</td>
</tr>
<tr>
<td>FP7</td>
<td>Seventh Framework Program</td>
</tr>
<tr>
<td>FTE</td>
<td>Full Time Equivalent</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GB</td>
<td>Giga Byte</td>
</tr>
<tr>
<td>GC</td>
<td>EUDAT General Council</td>
</tr>
<tr>
<td>GÉANT</td>
<td>European multi-gigabit computer network for research and education purposes.</td>
</tr>
<tr>
<td>GHR</td>
<td>CNRI top level Global Handle Registry</td>
</tr>
<tr>
<td>GSI</td>
<td>Grid Security Infrastructure</td>
</tr>
<tr>
<td>HLEG</td>
<td>High level Expert Group</td>
</tr>
<tr>
<td>HPC</td>
<td>High Performance Computing</td>
</tr>
<tr>
<td>HPSS</td>
<td>High Performance Storage System archiving facility</td>
</tr>
<tr>
<td>HTML</td>
<td>Hyper Text Markup Language</td>
</tr>
<tr>
<td>IdP</td>
<td>Identity Provider</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IGTF</td>
<td>The International Grid Trust Federation</td>
</tr>
<tr>
<td>ILL</td>
<td>An ESFRI project in the Materials and Analytical Facilities domain.</td>
</tr>
<tr>
<td>IMPACT</td>
<td>IMPACT (IMproving Protein Annotation through Coordination and Technology) is a three-year EU-funded project</td>
</tr>
<tr>
<td>IPCC</td>
<td>The Intergovernmental Panel on Climate Change</td>
</tr>
<tr>
<td>IPR</td>
<td>Intellectual Property Right</td>
</tr>
<tr>
<td>iRODS</td>
<td>Integrated Rule-Oriented Data System</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>JAVA</td>
<td>Platform independent programming language and computing platform first released by Sun Microsystems</td>
</tr>
<tr>
<td>JIRA</td>
<td>Web-based project tracker system</td>
</tr>
<tr>
<td>JRA</td>
<td>Joint Research Activity</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation, which is a lightweight data-interchange format</td>
</tr>
<tr>
<td>KM3Net</td>
<td>Kilometre Cube Neutrino Telescope. An ESFRI project in the Physical Sciences and Engineering domain.</td>
</tr>
<tr>
<td>LHC</td>
<td>Large Hadron Collider</td>
</tr>
<tr>
<td>LHS</td>
<td>Local Handle Service</td>
</tr>
<tr>
<td>LifeWatch</td>
<td>E-Science and Technology Infrastructure for Biodiversity Data and Observatories. An ESFRI project in the Environmental Sciences domain.</td>
</tr>
<tr>
<td>LOFAR</td>
<td>LOw Frequency ARray</td>
</tr>
<tr>
<td>MD</td>
<td>Metadata</td>
</tr>
<tr>
<td>METAFOR</td>
<td>Common Metadata for Climate Modelling Digital Repositories</td>
</tr>
<tr>
<td>MoU</td>
<td>Memorandum of Understanding</td>
</tr>
<tr>
<td>MSO</td>
<td>Micro Service Objects</td>
</tr>
<tr>
<td>MSS</td>
<td>Mass Storage System</td>
</tr>
<tr>
<td>NEERI</td>
<td>Networking Event for European Research Infrastructures</td>
</tr>
<tr>
<td>NERIES</td>
<td>NERIES (Network of Research Infrastructures for European Seismology) was a project co-funded by the EU within the FP6.</td>
</tr>
<tr>
<td>NGS</td>
<td>Next generation sequencing</td>
</tr>
<tr>
<td>NREN</td>
<td>National Research and Education Network</td>
</tr>
<tr>
<td>OAIS</td>
<td>Open Archival Information System</td>
</tr>
<tr>
<td>OAI-PMH</td>
<td>Open Archives Initiative Protocol for Metadata Harvesting</td>
</tr>
<tr>
<td>OASIS</td>
<td>Organization for the Advancement of Structured Information Standards</td>
</tr>
<tr>
<td>ODE</td>
<td>Opportunities for Data Exchange</td>
</tr>
<tr>
<td>OGF</td>
<td>Open Grid Forum</td>
</tr>
<tr>
<td>OGS</td>
<td>Open Grid Service</td>
</tr>
<tr>
<td>OGSA</td>
<td>Open Grid Services Architecture</td>
</tr>
<tr>
<td>OpenAIRE</td>
<td>Open Access Infrastructure for Research in Europe</td>
</tr>
<tr>
<td>OSCT</td>
<td>Operational Security Coordination Team</td>
</tr>
<tr>
<td>PAP</td>
<td>Policy Administration Point</td>
</tr>
<tr>
<td><strong>PARADE</strong></td>
<td>Partnership for Advanced Data in Europe</td>
</tr>
<tr>
<td><strong>PDP</strong></td>
<td>Policy Decision Point</td>
</tr>
<tr>
<td><strong>PEP</strong></td>
<td>Policy Enforcement Point</td>
</tr>
<tr>
<td><strong>PESI</strong></td>
<td>PESI (A Pan-European Species directories Infrastructure) is a three-year project co-funded by the EU under the FP7</td>
</tr>
<tr>
<td><strong>PID</strong></td>
<td>Persistent Identifier</td>
</tr>
<tr>
<td><strong>PLANETS</strong></td>
<td>Planets (Preservation and Long-term Access through NETworked Services) was a project co-funded by the EU within the FP6</td>
</tr>
<tr>
<td><strong>PM</strong></td>
<td>EUDAT Project Manager</td>
</tr>
<tr>
<td><strong>PMO</strong></td>
<td>EUDAT Project Management Office</td>
</tr>
<tr>
<td><strong>PRACE</strong></td>
<td>Partnership for Advanced Computing in Europe</td>
</tr>
<tr>
<td><strong>Preservation</strong></td>
<td>Provision of generic support for permanent access to ‘physical’ data – the bits and bytes – including storage, replication, provenance, etc</td>
</tr>
<tr>
<td><strong>PRINS</strong></td>
<td>Pan-European Research Infrastructure for Nanostructures. An ESFRI project in the Physical Sciences and Engineering domain.</td>
</tr>
<tr>
<td><strong>QA</strong></td>
<td>Quality assurance</td>
</tr>
<tr>
<td><strong>QoS</strong></td>
<td>Quality of Service</td>
</tr>
<tr>
<td><strong>RAD</strong></td>
<td>Rapid Application Development</td>
</tr>
<tr>
<td><strong>RI</strong></td>
<td>Research Infrastructure</td>
</tr>
<tr>
<td><strong>RoR</strong></td>
<td>Repository of Record</td>
</tr>
<tr>
<td><strong>SaaS</strong></td>
<td>Storage-as-a-Service</td>
</tr>
<tr>
<td><strong>SAF</strong></td>
<td>EUDAT Services and Architectural Forum</td>
</tr>
<tr>
<td><strong>SAML</strong></td>
<td>Security Assertion Markup Language</td>
</tr>
<tr>
<td><strong>SCAPE</strong></td>
<td>SCAPE (SCAlable Preservation Environments) is a project co-funded by EU within the FP7</td>
</tr>
<tr>
<td><strong>SDI</strong></td>
<td>European Spatial Data Infrastructures</td>
</tr>
<tr>
<td><strong>SEALS</strong></td>
<td>SEALS (Semantic Evaluation at Large Scale) is a project co-funded by the EU within the FP7.</td>
</tr>
<tr>
<td><strong>SET-Plan</strong></td>
<td>The European Strategic Energy Technology Plan</td>
</tr>
<tr>
<td><strong>SHAMAN</strong></td>
<td>Sustaining Heritage Access through Multivalent ArchiviNg. A Large Integrated Project co-funded by the EU within the FP7.</td>
</tr>
<tr>
<td><strong>SHARE</strong></td>
<td>Survey of Health, Ageing and Retirement in Europe project. An ESFRI project in the Social Sciences and Humanities domain.</td>
</tr>
<tr>
<td><strong>SIAEOS</strong></td>
<td>The Svalbard Integrated Arctic Earth Observing System. An ESFRI project in the Environmental Sciences domain.</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SKA</td>
<td>The Square Kilometre Array. An ESFRI project in the Physical Sciences and Engineering domain.</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SME</td>
<td>Small and medium enterprises</td>
</tr>
<tr>
<td>SNIA</td>
<td>Storage Networking Industry Association</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
<tr>
<td>SOLR</td>
<td>Apache full text search engine</td>
</tr>
<tr>
<td>SPIRAL2</td>
<td>An ESFRI project in the Physical Sciences and Engineering domain.</td>
</tr>
<tr>
<td>SR</td>
<td>Safe Replication service</td>
</tr>
<tr>
<td>SRB</td>
<td>Storage Resource Broker</td>
</tr>
<tr>
<td>SSH</td>
<td>Social Sciences and Humanities</td>
</tr>
<tr>
<td>SSO</td>
<td>Single Sign-On</td>
</tr>
<tr>
<td>SSS</td>
<td>Simple Store Service</td>
</tr>
<tr>
<td>STM</td>
<td>Science, Technology and Medicine</td>
</tr>
<tr>
<td>SVN</td>
<td>Apache Subversion version control system</td>
</tr>
<tr>
<td>TACC</td>
<td>Texas Advanced Computing Center</td>
</tr>
<tr>
<td>TERENA</td>
<td>Trans-European Research and Education Networking Association</td>
</tr>
<tr>
<td>TF</td>
<td>Task Force</td>
</tr>
<tr>
<td>TRAC</td>
<td>Trustworthy Repositories Audit &amp; Certification</td>
</tr>
<tr>
<td>TTS</td>
<td>EUDAT Trouble Ticket System</td>
</tr>
<tr>
<td>UNICORE</td>
<td>Uniform Interface to Computing Resources</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>URN</td>
<td>Uniform Resource Name</td>
</tr>
<tr>
<td>VAMP</td>
<td>Vo Architectural Middleware Planning Workshop</td>
</tr>
<tr>
<td>VLO</td>
<td>CLARIN Virtual Language Observatory</td>
</tr>
<tr>
<td>VO</td>
<td>Virtual Organisation</td>
</tr>
<tr>
<td>VOMS</td>
<td>Virtual Organization Membership Service</td>
</tr>
<tr>
<td>VPH-I</td>
<td>Virtual Physiological Human Initiative</td>
</tr>
<tr>
<td>WAYF</td>
<td>Where Are You From service</td>
</tr>
<tr>
<td>Web 2.0</td>
<td>A term commonly associated with web applications that facilitate interactive information sharing, interoperability, user-centred design, and collaboration on the World Wide Web.</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>WLCG</td>
<td>Worldwide LHC Computing Grid</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
<tr>
<td>XACML</td>
<td>eXtensible Access Control Markup Language</td>
</tr>
<tr>
<td>XAML</td>
<td>eXtensible Application Markup Language</td>
</tr>
<tr>
<td>XFEL</td>
<td>The European X-Ray Laser Project. An ESFRI project in the Materials and Analytical Facilities domain.</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language</td>
</tr>
<tr>
<td>XSEDE</td>
<td>Extreme Science and Engineering Discovery Environment</td>
</tr>
</tbody>
</table>