Project Number: **RI-312579**

Project Acronym: **ER-flow**

Project Full Title:

Building an European Research Community through Interoperable Workflows and Data

Theme: **Research Infrastructures**

Call Identifier: **FP7-Infrastructures-2012-1**

Funding Scheme: **Coordination and Support Action**

Milestone MS3.1
Data interoperability requirements of the infrastructure

Johan Montagnat, Nadia Cerezo

Start date of project: 01/09/2012 Duration: 24 months

Lead Contractor: University of Westminster

Dissemination Level: PU

Version: 1.0

ER-flow is supported by the FP7 Capacities Programme under contract no RI-312579

Table of Contents

Glossary

CGI	Coarse-Grained Interoperability
CNRS	Centre National de la Recherche Scientific (French National Centre for Scientific Research)
DCI	Distributed Computing Infrastructure
EGI	European Grid Infrastructure
EMI	European Middleware Initiative
EUDAT	European Data Infrastructure project
FGI	Fine-Grained Interoperability
GEMLCA	Grid Execution Management for Legacy Code Applications
IWIR	Interoperable Workflow Intermediate Representation
jSAGA	Java implementation of the SAGA API
SAGA	Simple API for Grid Applications
SCI-BUS	Scientific gateways-Based User Support project
SRM	Storage Resource Manager
SSP	SHIWA Simulation Platform
SZTAKI	Magyar Tudomanyos Akademia Szamitastechnikai Kutato Intezete
UoW	University of Westminster
VO	Virtual Observatory
WP	Work Package

Table 1. Glossary

1 Data interoperability

Data Interoperability is a broad area arising from the need of widespread user communities to share and reuse data acquired and stored in different places. Regular use of eScience platforms to speed-up discovery increases the need for data interoperability at a large-scale. Ensuring data interoperability among different users implies **publishing data** (making it known and accessible), giving **means of interpretation** (documenting data so that it can be reused by different actors), and enabling **data transfers** across different resources. This high-level definition of data interoperability covers many different challenges (e.g. the use of interpretable data formats, standard data access protocol, possibly data access control…), some of which are addressed in this document more specifically focussed on ER-Flow objectives.
The SHIWA project[1] has set up a multi-workflow systems management environment aiming at making these systems interoperable. It operationalized its concept through the SHIWA Repository[2], which enables executable workflows sharing, and the SHIWA Simulation Platform[3] (SSP), which enables multi-workflows execution over multiple Distributed Computing Infrastructures (DCIs). However, SHIWA put little emphasis on data interoperability issues. In particular, there is no data repository, no common data specification interface nor means to transfer data across different DCIs within the SSP. As a result, exchanging data across workflow systems or underlying infrastructures remained difficult, and data interoperability proved to be a substantial showstopper towards workflow interoperability. The aim of ER-Flow Work Package 4 is therefore to analyze requirements of the supported research communities in scientific data interoperability and their technical implications. This data interoperability study is split in two documents: this document (MS3.1) focuses on technical implications of data interoperability and its impact on the SSP, while MS5.1 focuses on user-level requirements for data interoperability.

1.1 Positionning

Web technologies have been pioneering data interoperability challenges since the emergence of the worldwide web. To ease data exchanges and processing over different servers, the W3C[4] developed many standards related to data interoperability, including data representation (*e.g.* XML and RDF structuring languages), data indexing (*e.g.* URLs and URIs), data transfers (*e.g.* HTTP protocol), and data processing (*e.g.* Web Services), with particular emphasis on text-based data. Many of these technologies are highly relevant for addressing general data interoperability challenges.
In the scientific area, eScience platforms are increasingly used for sharing and processing scientific data sets. These platforms are more focussed towards the management and processing of large data repositories (often referred to as Big Data[5] nowadays, to stress both data volume and scientific data sets complexity). The majority of scientific data is binary, stored within opaque files, and it is commonly spread over multiple file sets. This led to the development of specific scientific files cataloguing and transfer tools. Metadata annotating the raw data has become mandatory in order to manipulate and interpret *Big* scientific data. In eScience platforms, metadata often includes description of raw data, complementary information on the context of raw data acquisition, and provenance information.
Web technologies nowadays also address metadata description and manipulation challenges. The evolution of the *Web of Data* towards a *Semantic Web* led to the definition of the *Linked Data* concept[6], which eases data interlinking and leverages data (re-)usability

[1] SHIWA project: http://www.shiwa-workflow.eu
[2] SHIWA repository: http://repo.shiwa-workflow.eu/
[3] SHIWA Simulation Platform: http://ssp.shiwa-workflow.eu/
[4] World Wide Web Consortium (W3C), http://www.w3.org
[5] Big Data, http://en.wikipedia.org/wiki/Big_data
[6] Linked Data, http://en.wikipedia.org/wiki/Linked_data

through unique identification, referencing means, and rich description of data objects. An important particularity of eScience platforms is that they may manipulate sensitive data for which appropriate access control and privacy preserving rules are needed, while the Web of Linked Data often targets open data sources, putting little emphasis on data protection.
The main objective of this study is the interoperability of scientific data consumed and produced by scientific workflow systems embedded in eScience platforms such as the SSP. The emphasis is therefore clearly put on scientific data (often binary data files) and associated data management tools found on Distributed Computing Infrastructures. The standards developed by the W3C are considered highly relevant, due to there very wide adoption, when applying to the type of data manipulated.

1.2 Area of study

In the context of scientific data consumed and produced by scientific workflow systems (data sets), data interoperability issues usually refer to exchanges of data over different workflow systems and different underlying Distributed Computing Infrastructure on which the data sets are stored. Data interoperability challenges may arise from:

1. Different **data representations** in use (different data types and data sets specification, different encodings).
2. Different **file formats** for a same type of data.
3. Different **data storage and indexing** means (files, databases, and even data sets defined as a result of a workflow computation in some cases).
4. Different **data exchange** means (different data transfer protocols, different I/O parameters passing modes).

The management of different file formats is too application-dependent to be considered in this study. Data files are usually manipulated as opaque objects by workflow management systems. Data storage is predominantly organized through file hierarchies in Distributed Computing Infrastructures. Hence, only data file exchanges will be considered in this study. Direct access to data stored in remote databases is possible for specialized workflow activities, but these are also considered as opaque from the workflow engine perspective. The focus is therefore put on **data representation** (objective 1) and **file exchange** means (objective 2) applicable to Distributed Computing Infrastructures in this document.

1.3 Other initiatives

The problem of data interoperability is not new [Kahn, 95] and several initiatives studying various aspects of data interoperability challenges have been conducted. The W3C, driving Web development over the past decades, is undoubtedly the largest and best recognized institution dealing with data interoperability in a very wide context. W3C only produces specifications and recommendations though. The technical implementations of W3C specifications may be heterogeneous and more or less conforming the standards established.
Other projects in the context of distributed computing have focussed more specifically on remote file storage and file transfers. The Globus toolkit[7] is a pioneer and *de facto* widely adopted middleware for distributed computing. In particular, Globus provides a foundational public key-based security infrastructure (GSI), which guarantees interoperability between different systems adopting it at the lowest level. Its wide adoption among various middleware development initiatives it the key to multiple infrastructures interoperability, especially file exchange capabilities. The Globus Toolkit also provides the GridFTP[8] high-performance data transfer protocol, which became a *de facto* standard for file transfers within and across different computing infrastructures. More recently, Globus developed the Globus online data

[7] Globus toolkit, http://www.globus.org/toolkit
[8] GridFTP, http://www.globus.org/toolkit/docs/latest-stable/gridftp/

transfer service[9], a third-party service for managing file transfers over any compliant hosts. Globus online currently accounts for more that 12 PB of data transfers over the Internet.
The European Middleware Initiative[10] borrows from the Globus toolkit its foundational security infrastructure and data transfer capabilities to build on top distributed file management services, in particular the LFC File Catalog, Storage Element interfaces complying to the SRM standard[11], and the GFAL/LCG utilities for files transfer and replication. The Java implementation of the Simple API for Grid Applications[12] (jSAGA) also proposes a plugin-based extensible middleware that includes access to various file catalogues and file transfer using various file transfer protocols in use on Distributed Computing Infrastructures (such as HTTP, SRM, FTP, GridFTP, local files…). The SCI-BUS European project[13], which develops gateway technologies to facilitate access to various distributed computing infrastructures, is currently developing a cross-infrastructures file transfer tool on top of jSAGA.
Recently, the European Data Infrastructure project[14] (EUDAT) was started to tackle the specific challenges of data management in Distributed Computing Infrastructures. It follows a preliminary requirement study identifying the need for a coherent approach to data access and preservation [Koski, 09]. The EUDAT implementation of the challenges identified in the area of distributed data management among various user communities is the delivery of 5 data management-related services, namely:

- Safe Replication: replication of data in selected data centres.
- Dynamic Replication: stage data between EUDAT resources and computing resources.
- Metadata: joint open metadata domain for all data stored by EUDAT centres.
- Simple Store: data upload, storage and sharing.
- AAI: Authentication and Authorization Infrastructure.

EUDAT services are still under specification and development.
Many tools and services mentioned above can be considered as a basis for a data interoperability solution. In particular, they tackle the problems of file indexing (through global-scale file catalogues) and cross-infrastructures file transfers (through multi-protocols file access APIs), *i.e.* **objective 2** as identified in the previous section. The problem of cross-infrastructures users authentication and authorization is a cross-concern that received little attention though. More generally, the problem of managing non file-based data sets and interfacing data with workflow systems received very little attention.

1.4 Data interoperability in ER-Flow

The SSP exploited in the ER-Flow project is a multi-workflow systems platform operating over different Distributed Computing Infrastructures. Each workflow system consumes and produces data sets using its own data I/O interface, data representation and data access protocols. In ER-Flow, the problems of data interoperability arise from the need to exchange data sets among different workflow management systems. The coarse-grained workflow interoperability technology implemented in the SSP allows for the design and execution of meta-workflows, where a master workflow embeds sub-workflows as some of its activities (see Figure 1 below). The master workflow receives input data from the SSP user interface and returns output data to the end user through this interface. Intermediate data sets also need to be exchanged between the master workflow system and the embedded workflow systems at sub-workflow input and sub-workflow output.

[9] Globus online, https://www.globusonline.org
[10] Eureopean Middleware Initiitive (EMI), http://www.eu-emi.eu
[11] Storage Resource Manager (SRM), http://en.wikipedia.org/wiki/Storage_Resource_Manager
[12] Java Simple API for Grid Applications (jSAGA), http://grid.in2p3.fr/jsaga/
[13] Scientific Gateway-Based User Suport (SCI-BUS), https://www.sci-bus.eu
[14] European Data Infrastructure project (EUDAT), http://www.eudat.eu

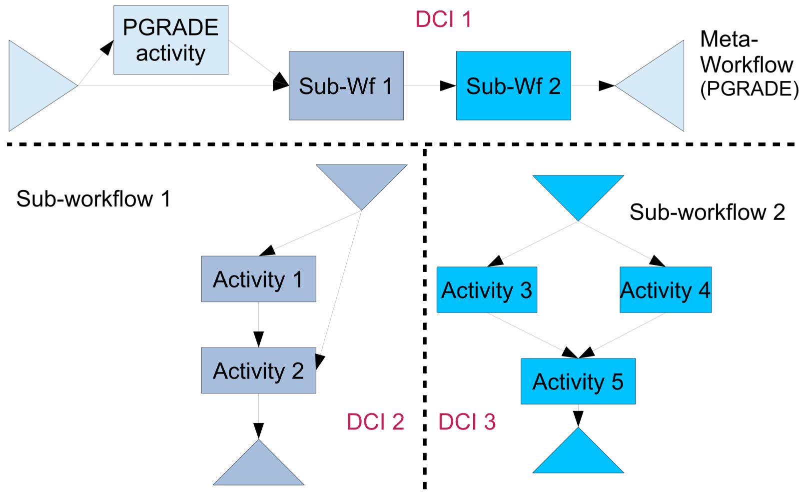

Figure 1. A typical meta-workflow as executed in the SSP: a master PGRADE workflow, executing on computing infrastructure DCI1, embeds a native activity and two sub-workflows as sub-activities. A potentially different workflow engine, potentially using a different computing infrastructure (DCI2 or DCI3), executes each sub-workflow. The inputs and outputs of the sub-workflows are chained with the master workflow process. The master workflow input and outputs are stored in files and received from / returned to the SSP user interface.

To execute the meta-workflow illustrated in Figure 1, several data exchanges need to be considered (see Figure 2 below, where orange arrows show data transfers explicitly). Data received from the user interface ① and data produced by the master workflow ② may be sent to sub-workflows. Data produced by a sub-workflow may be consumed by another sub-workflow ③ or by the master workflow ④.

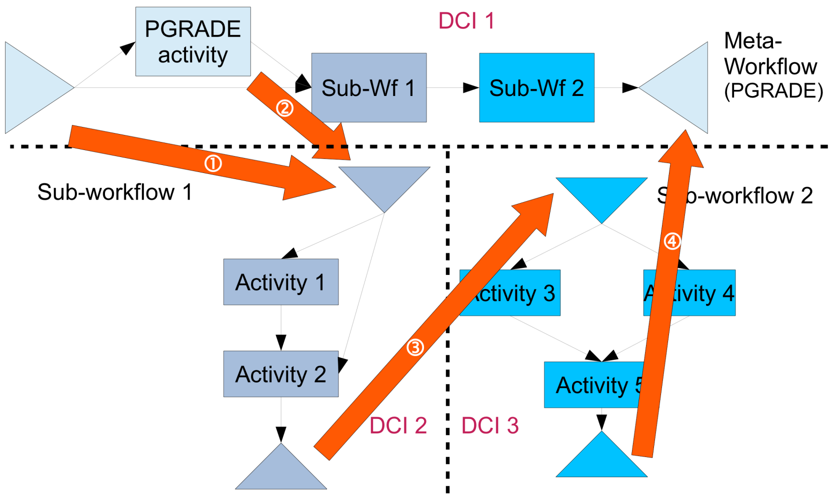

Figure 2. Data transfer needed between the master workflow and the embedded sub-workflows. Sub-workflow 1 receives as input a mixture of data from the master workflow input and data produced by the native PGRADE activity. Sub-workflow 2 receives as input data produced by sub-workflow 1. Master workflow receives as output data produced by sub-workflow 2. The data exchanged may be exchanged by direct parameters passing, through a specific data management system (*e.g.* data records in a relational database) or through files. In case of files, transfer across different DCIs may be needed.

Currently, the input data is passed to the master workflow from the graphical interface and the end user retrieves output data from the master workflow as files. A file identifier may be specified for each input port associated to an input activity or each output port associated to an output activity of the master workflow. Similarly, data is exchanged between the master workflow and embedded sub-workflows as file name identifiers (file names sent to the input ports of the activity wrapping a sub-workflow are transferred to the sub-workflow engine and file names generated by sub-workflow engines are mapped to output port of the wrapping activity). There is no explicit management of the file transfers between different infrastructures, nor ability to express non-file parameters. Data is always exchanged as a single file name per input/output port, regardless of the input/output interface of the embedded sub-workflow engines.

Ideally, the data interoperability mechanism should ensure:

- File exchange (**objective 2** in Section 1.2):
    - Input / Output data transfers between different wokflow engines / DCIs, whether data is stored into files or represented as non-file parameters.
    - Transfer of Input / Output files across DCIs, taking into account the discrepancies between file access control systems and file transfer protocols that may exist.
    - Transfer of Input / Output files between the platform and external machines (*e.g.* the user's machine).
- Interoperable data representation (**objective 1** in Section 1.2):
    - Transformation of non-file parameters taking into account different data representations that may be in use.
    - Adaptation of input / output data sets description to the workflow engine data interface.

The following section produces recommendation regarding the way to achieve this data interoperability level and studies the impact on the execution platform.

2 Infrastructure aspects of data interoperability

Implementing data interoperability in ER-Flow will have an impact on several components of the SSP. Section 2.1 describes the SSP components which are involved in this process. Section 2.2 discusses the need for a common representation between workflow management systems integrated in the platform. Section 2.3 identifies which data format could be considered in this context.

2.1 SSP architecture

The following components of the SSP are involved in the management of workflow data:

- The SSP portal is used for designing master workflows and configuring executable workflows. In particular, it includes:
  - The PGRADE workflow designer through which master workflow input and output ports are defined;
  - The executable workflow configuration interface through which files are associated to inputs and outputs; and
  - A proxy manager through which user credentials are uploaded.
- A MyProxy server is used to store medium-lived user X509 proxies.
- Various workflow management systems that can be used for sub-workflows execution are embedded. The GEMLCA[15] legacy application wrapper is used as a common invocation interface between the PGRADE master engine and embedded workflow systems.

As outlined in the previous Section, a data interoperability solution requires using a common data representation and enabling cross-DCI file transfers. The common data representation management will at least have an impact on:

- The SSP executable workflow configuration interface, so that non-file parameters and/or multiple I/O files may be specified. If the current interface based on a single exchange file is kept, it could be use to specify a meta-file containing all information on parameters and files which constitute the data set though. Similarly, the PGRADE workflow designer may be impacted in case a data set composed with multiple files is mapped to a single workflow activity port. To preserve the current designer, a single meta-file could be used. Meta-file specification may follow the Object Exchange and Reuse (ORE) specification defined by the Open Archives Initiative[16], which defines a standard format for file bundles.
- The I/O interface of all workflow system embedded in the platform (including the PGRADE master system), so that I/O data sets can be exchange between different systems. Potentially, this will have an impact on the GEMLCA wrapper.

In addition, cross-DCI transfer is not available in the SSP. A dedicated service will be needed to deliver this functionality (e.g. one of those identified in Section 1.3). This service needs to be synchronized with the platform proxy management system, potentially requiring an adaptation of this component.

2.2 Adapters

The data sets that workflow consumes as input or produces as outputs are composed of parameters (*e.g.* simple values such as an integer, etc) and/or data files. Data files are identified through symbolic file identifiers (*e.g.* URIs), which can be considered as textual parameter values (although a file identifier manipulated by a given system may be unknown from another one and be unusable without file transfer). All I/Os will therefore be considered as parameters in the remainder. Other modes of I/O descriptions, such as relational

[15] Grid Execution Management for Legacy Code Applications (GEMLCA), http://www.cpc.wmin.ac.uk/cpcsite/index.php/Gemlca

[16] Object Exchange and Reuse (ORE) standard, http://www.openarchives.org/ore/

database records, are considered too specific and excluded from this study. They can only be processed by dedicated workflow activities in a non-generic system.
The workflow systems embedded in the SSP all have a specific interface to describe their I/O parameters. They may use:

- Input and/or Output parameters described on the command line;
- Input (resp. Output) parameters in system-specific Input (resp. Output) files;
- Input parameters read from the process standard input and/or Output parameters written to the process standard output stream;
- Or any combination of the above.

Invoking a workflow engine therefore requires adapting to its specific interface. Given that any workflow system may exchange data with any other in the SSP (see Figure 2), the number of adaptors needed grows as the square of the number of workflow engines supported. Alternatively, the use of a common data sets description format, known from all embedded systems, reduces the number of adaptors to be developed significantly, as it can be used as a pivot representation and a single pair of adaptors (two-ways conversion between the workflow internal format and the pivot format) is then sufficient for each workflow system. It should also be noted that the adaptors might be integrated:

- Invasively, by modifying the workflow systems to make them aware of this pivot format; or
- Non-invasively, by developing a two-ways wrapper that receives an input pivot format file describing the workflow input data set, adapts it to the native workflow invocation interface, retrieves the workflow output in its native format, and convert the result in an output pivot format file.

Given the nature of the SSP platform, which aims at facilitating the integration and exploitation of existing workflow engines, a non-invasive approach is much preferred. This does not prevent some workflow systems to adopt the pivot representation in their code base though. A single invocation interface recommendation is preferable in this latter case.
The master workflow system plays a specific role as it triggers the invocations of embedded workflow systems. It should adopt the pivot data representation itself to ease data exchanges, and it should also make sure that intermediate pivot I/O files are transferred between itself and the workflow systems embedded. In the context of the SSP, the GEMLCA wrapper that shields the master system from the idiosyncrasies of embedded systems can be used for handling these I/O files.
The I/O pivot file may reference some data files through DCI-specific file naming schemes. As these symbolic file names usually only have a meaning in the context of a particular DCI, it might be needed to copy the files referenced from one DCI and change the file identifiers by their copy counter-part on the target DCI. File transfers may be handled either non-invasively by a third party service, or be integrated invasively in the embedded workflow (augmenting the target workflow with data transfer activities). The former, non-invasive solution is preferable. The availability of cross-DCI data transfer tools (see Section 1.3) should help in its implementation. It should be noted that cross-DCI transfer requires proper management of user credentials over both the source and the target DCI: the user requesting files transfer should be both recognized on these two DCIs and authorized to access the files. This potentially implies the management of multiple credentials per users, or the use of robot certificates if these are accepted by the DCI usage policies.

2.3 Pivot data representation

As explained above, the use of a pivot data representation is strongly encouraged to solve the data representation challenge of inter-workflow data sets interoperability. This representation should be accompanied with a standardized in-file representation (serialization and deserialization process) to ease data exchanges across different systems and DCIs.

The pivot data representation should enable the description of workflow data sets, including primitive parameter values (numerical values, text strings…) and symbolic file identifiers. Scientific workflows usually manipulate large data sets made of lists or arrays of data values or files. The construction of such data sets should therefore be supported.

2.3.1 Primitive data types

Definitions of a wide variety of primitive data types as well as constructs to create complex data structures are standardized in the W3C XSD Datatypes document[17]. Built-in data types include numerical values, character strings, dates, times, and binary values among others. It is recommended to follow this standard for primitive data types representation. Yet, scientific workflows usually only make use of a subset of the primitive data types included in the W3C standard. The Interoperable Workflow Intermediate Representation (IWIR) defined in the context of the SHIWA project is a pivot workflow language only considering the following primitive data types for instance: booleans, integers, doubles, strings and file identifiers. Also, all file management systems in DCIs accessible from the SSP are using URIs for file identification. It is therefore recommended to restrict the XSD primitive data types to those six:

- boolean, with two-values space {true, false}.
- long, a mathematical integer number between -9223372036854775808 and 9223372036854775807.
- double, an IEEE double-precision 64-bit floating point datatype[18]
- string, an XML character string.
- anyURI, an International Resources Identifier Referent (IRI) used to identify a file (local or remote).
- base64Binary, arbitrary binary data encoded using the Base64 Encoding[19].

2.3.2 Data arrays

Homogeneous arrays of either primitive type atomic values (simple arrays) or sub-arrays (nested arrays) are needed to represent workflow data sets. Arrays may be defined in XSD using the Complex Type Definition Schema Component and restricting the underlying sequence of items to all have the same data type (note that List data types, deriving from the Simple Type Definition Schema, could be used to represent arrays of primitive types but not nested arrays).

2.3.3 File representation

Using XSD makes pivot file representation standard, conforming to the XSD schema and using XML serialization.

2.3.4 Pivot indexing scheme and manipulation

Pivot data sets should be accessible uniformly regardless of the storage technology used in the backend. A URL should be associated to a data object to identify it similarly to what Digital Object Identifiers (DOI) are for scientific papers. A data set manipulation API may be defined to allow for basic manipulation of these data sets, for instance:

- Listing the physical storage location(s) associated to the data set.
- Listing the metadata associated to a data set.
- Downloading a data set.
- Registering a data set.

[17] XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, http://www.w3.org/TR/xmlschema11-2/
[18] IEEE double-precision floating point, IEEE standard 754-2008, http://dx.doi.org/10.1109/IEEESTD.2008.4610935
[19] Base64 Data Encoding, http://www.ietf.org/rfc/rfc3548.txt

- Deleting a data set.
- Changing the permissions on a data set.
- Transferring a data set to another physical location.

3 Conclusions

Implementing data interoperability in the context of the ER-Flow project requires tackling the data representation and the cross-DCIs file transfer challenges. It is recommended to follow the W3C XSD specification to deliver a pivot format representation and exploring existing multi-DCI file transfer tools such as jSAGA or the SCI-BUS Data Bridge to address them. Incrementing the SSP with data interoperability would then imply to:

- Write adapters for each supported workflow system to map pivot data sets to the system-specific data interface;
- Make use of these adapters through the sub-workflow system invocation wrapper;
- Integrate data set manipulation functionality linked with user credential management services in the platform, such as cross-DCI file transfer, data set listing, or metadata handling; and
- Potentially make use of an archive format such as ORE to adapt to the current PGRADE portal and workflow engines, which implement communication with embedded workflow systems through a single file (otherwise the PGRADE engine interface and the SSP I/O specification GUI need to be updated accordingly).

References

[Kahn, 95] Kahn R. and Wilensky R., “Digital Object Architecture”, Technical Report cnri-dlib/tn95-01, Corporation for National Research Initiatives, 1995. http://www.cnri.reston.va.us/k-w.html.

[Koski, 09] Koski K., Gheller C., Heinzel S., Kennedy A., Streit A., Wittenburg P. “Strategy for a European Data Infrastructure”. White Paper, 2009. http://www.csc.fi/english/pages/parade.

[Plank, 11] Plankensteiner K., et al., “IWIR Specification”, SHIWA deliverable D6.1, https://www.shiwa-workflow.eu/documents/10753/55350/IWIR+v1.1+Specification.