Reference Model Guided Engineering

Zhao, Z.; Jeffery, K.

DOI
10.1007/978-3-030-52829-4_5

Publication date
2020

Document Version
Final published version

Published in
Towards Interoperable Research Infrastructures for Environmental and Earth Sciences

License
CC BY

Link to publication

Citation for published version (APA):
Zhao, Z., & Jeffery, K. (2020). Reference Model Guided Engineering. In Z. Zhao, & M. Hellström (Eds.), *Towards Interoperable Research Infrastructures for Environmental and Earth Sciences: A Reference Model Guided Approach for Common Challenges* (pp. 82-99). (Lecture Notes in Computer Science; Vol. 12003). Springer. https://doi.org/10.1007/978-3-030-52829-4_5

Reference Model Guided Engineering

Zhiming Zhao[1](✉) and Keith Jeffery[2]

[1] Multiscale Networked Systems, University of Amsterdam,
1098XH Amsterdam, The Netherlands
z.zhao@uva.nl
[2] Keith G Jeffery Consultants, Faringdon, UK
keith.jeffery@keithgjefferyconsultants.co.uk

Abstract. Environmental research infrastructures (RIs) support their respective research communities by integrating large-scale sensor/observation networks with data curation and management services, analytical tools and common operational policies. These RIs are developed as service pillars for intra- and interdisciplinary research; however, comprehension of the complex, interconnected aspects of the Earth's ecosystem increasingly requires that researchers conduct their experiments across infrastructure boundaries. Consequently, almost all data-related activities within these infrastructures, from data capture to data usage, need to be designed to be broadly interoperable in order to enable real interdisciplinary innovation and to improve service offerings through the development of common services. To address these interoperability challenges as they relate to the design, implementation and operation of environmental RIs, a Reference Model guided engineering approach was proposed and has been used in the context of the ENVRI cluster of RIs. In this chapter, we will discuss how the approach combines the ENVRI Reference Model with the practices of Agile systems development to design common data management services and to tackle the dynamic requirements of research infrastructures.

Keywords: Research infrastructure · Reference Model · Interoperability · Agile

1 Introduction

Many key problems in environmental science are intrinsically interdisciplinary; the study of climate change, for example, involves the study of the atmosphere, but also earth processes, the oceans and the biosphere. Modelling these processes individually is difficult enough, but modelling their interactions is another order of complexity entirely. Scientists are challenged to collaborate across conventional disciplinary boundaries, but must first discover, extract and understand data dispersed across many different sources and formats.

Data-centric research differs from classical approaches for analytical modelling or computer simulation insofar as new theories are measured first and foremost against huge quantities of observations, measurements, documents and other data sources culled from

Z. Zhao and M. Hellström (Eds.): Towards Interoperable Research
Infrastructures for Environmental and Earth Sciences, LNCS 12003, pp. 82–99, 2020.
https://doi.org/10.1007/978-3-030-52829-4_5

a range of possible sources. To enable such science, the underlying research infrastructure must provide not only the necessary tools for data discovery, access and manipulation but also facilities to enhance collaboration between scientists of different backgrounds.

Environmental research infrastructures (RIs) support user communities by providing federated data curation, discovery and access services, analytical tools and common operational policies integrated around large-scale sensor/observer networks, often deployed on a continental scale. Examples in Europe include LifeWatch[1] (concerned with biodiversity), EPOS[2] (solid Earth science), Euro-Argo[3] and EMSO[4] (ocean monitoring), as well as ICOS[5] and the new EISCAT_3D system (atmosphere)[6]. These infrastructures are developing into important pillars for their respective user communities, but are also intended to support interdisciplinary research as well as more specific research data aggregators such as Copernicus[7] within the context of GEOSS[8]. As such, it is very important that data-related activities are well integrated in order to enable data-driven system-level science [2]. This requires standard policies, models and e-infrastructure to improve technology reuse and ensure coordination, harmonization, integration and interoperability of data, applications and other services. However, the complex nature of environmental science seems to result in the development of environmental RIs that meet only the requirements and needs of their own specific domains, with very limited interoperability of data, services, and operation policies among infrastructures.

It is thus important to identify technical and organizational commonalities for the cluster of research infrastructures in environmental and Earth sciences and provide a unified data discovery and access services to the whole RI activity cycle. This chapter presents the engineering model developed in the EU H2020 projects ENVRI, ENVRIplus and ENVRI-FAIR [3] for 1) combining both domain-specific characteristics and common abstractions; 2) harmonising RI-specific requirements with common operations; and 3) accounting for both existing generic e-infrastructures already adopted by existing RIs. The chapter is an extension of the earlier publication in IEEE eScience 2015 [1].

2 Engineering Challenges in Environmental RIs

Environmental RIs collectively play an important role in environmental and Earth science research in Europe, as shown in Fig. 1, with more than half of them, prioritised in the roadmap of the European Strategy Forum on Research Infrastructures (ESFRI) [4].

The RIs are in one or across multiple environmental domains: atmosphere; bio- or ecological; aquatic; and solid earth. There is considerable variation in their states of development.

[1] http://www.lifewatch.eu/.

[2] http://www.epos-eu.org/.

[3] http://www.euro-argo.eu/.

[4] http://www.emso-eu.org/.

[5] http://www.icos-infrastructure.eu/.

[6] https://www.eiscat3d.se/.

[7] http://copernicus.eu/.

[8] http://www.earthobservations.org/geoss.php.

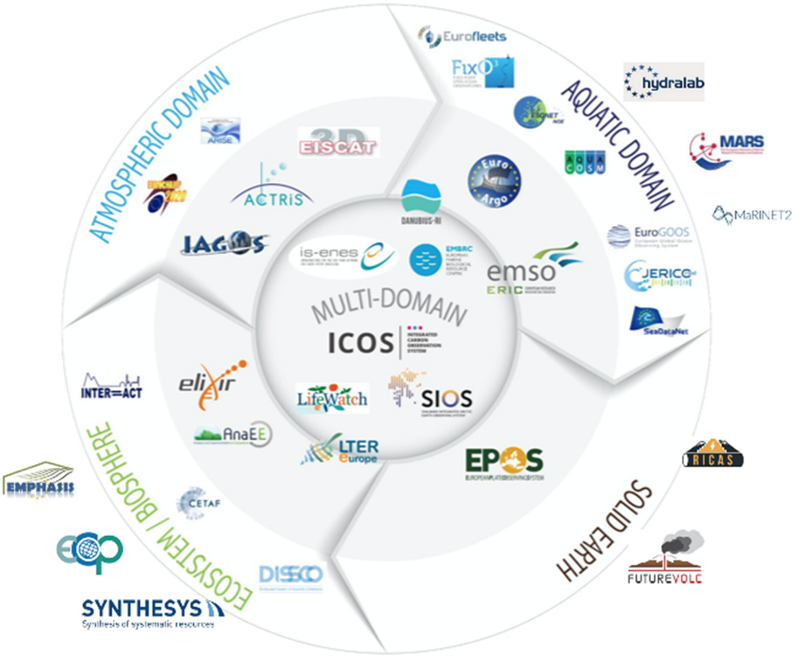


Fig. 1. Key European research infrastructures in environmental and Earth sciences.

2.1 Interoperability Challenges

In the earlier chapters, we discussed that one of the key missions in the cluster project of ENVRI is to provide reusable solutions to common problems these research infrastructures face and promote their interoperability for future system level of sciences [3].

In the ENVRI project, we reviewed existing interoperability solutions [5] from different specific aspects: infrastructure, middleware, and workflow. Typically, these solutions are realised iteratively, building adapters or connectors between two components and then deriving new service layer models for standardization via a community effort. Such a process of iteration can gradually promote the evolution of new standards for both infrastructures and the service layers above them, but will not completely solve all interoperability problems while the diversity between infrastructures and the gaps between standards remain significant [6]. White et al. [7] argued that an interoperability reference model is needed to complement models of application and infrastructure.

For those environmental RIs that are currently under construction or in preparation, it, therefore, becomes urgent to guide their development so that they can be immediately interoperable once operational.

2.2 Challenges for Enabling System-Level Science

To perform system-level environmental science, scientists face challenges with respect to data accessing, processing and publication:

1. *Obtaining and harmonizing data from different sources.* Data are often in different formats, annotated using different metadata, and retrieved via catalogues with different interfaces.
2. *Identifying different levels of data from the same instruments and experiment.* Data, being quality controlled and processed, are labelled as being of different levels during the data lifecycle, for example, raw input data (level 0) versus derived datasets (levels 1 or higher). Identifying different levels of data from the same instruments is crucial for precisely understanding their meaning.
3. *Selecting and combining data processing models from different domains.* Data processing models are often represented as workflows of services with attached datasets in different languages and require different execution engines to realise.
4. *Selecting optimal infrastructure upon which to execute applications.* Infrastructures often provide different scheduling and monitoring tools.
5. *Publishing data objects in different research infrastructures.* Data objects should be both identifiable and citable.

Environmental RIs provide the tools to help with this, but only if their services are sufficiently interoperable. To enable interdisciplinary research across RIs from different sub-domains of environmental science, there are a number of principles that any interoperable services and their supporting infrastructure should adhere to:

- *Simple but effective.* Scientists should be able to use, analyse, compose and store data from distributed sources in an easy but effective way, with appropriate metadata generated at all stages in order to trace data provenance.
- *Formal syntax.* the datasets should possess (a) a well-defined schema to describe attributes, types and permitted values (for validation); (b) referential integrity to avoid any updating problem; (c) functional integrity so that each attribute has no dependencies other than the object being described in order to ensure correct representation of the world of interest. Software services should have defined functionality through formally-defined APIs with parameter lists and defined non-functional properties covering performance and trust, security, privacy.
- *Bridgeable semantics.* A certain degree of semantic mapping is required to bridge the diverse complex knowledge organizing systems needed by different scientific and technical domains, but all the tools and resources need to be documented in a principled, formal way first. For datasets, the semantics of attribute values must be defined and for services the semantics of the parameters in the API must be defined. In both cases the semantics of descriptions and keywords in the catalogue require definition.
- *Extensible and robust.* Available resources change and user demands fluctuate; core RI services must be elastic and fault-tolerant, and provide programmatic interfaces for service composition.
- *Open yet secure.* Although most research data is open, there is a need to protect the privacy of researchers, attribute credit to individuals and organizations, embargo new research prior to publication and preserve authority and accountability constraints when transferring data between different technical and political domains.

In order to meet these rather wide-ranging principles, the ENVRIplus solutions build upon the results of earlier projects, the expertise of individual RIs, and the services of e-infrastructure initiatives. Filling in the gaps, the ENVRI community continues to work to:

1. Optimise data processing and develop common models, rules and guidelines for research data workflow documentation.
2. Facilitate data discovery and (re-)use following the FAIR principles[9], and provide integrated end-user information technology to access heterogeneous data sources.
3. Make data citable by building upon existing approaches with practical examples, exchanges of expertise, and agreements with publishers.
4. Facilitate the discovery of software services and their possible compositions.
5. Characterise users and build a community on top of existing RI communities.
6. Characterise ICT resources (including sensors and detectors) to allow virtualisation of the environment (for instance onto the grid- or cloud-based platforms) such that data and information management and analysis is optimised in terms of resource and energy expenditure.
7. Facilitate the connection of users, composed software services, appropriate data and necessary resources in order to meet end-user requirements.

2.3 Engineering Challenges

The development of Research Infrastructures in environmental Earth sciences has to consider not only the requirements discussed in Sect. 2.2, but also the status of the existing work, e.g. types of legacy assets, the maturity of available services, and usage of standards. Figure 1 shows a clear diversity among the research infrastructures in the cluster of environmental and Earth sciences:

1. *At different levels of development:* some infrastructures are partially or wholly in operation (e.g. Euro-Argo), while others are still under preparation or development when the ENVRIplus project starts.
2. *Having different development roadmaps:* infrastructures in the ESFRI roadmap (e.g. EMSO and LifeWatch) have a clear timeline with an established funding stream, while some other infrastructures are still funded under specific projects.
3. *Using different standards for specifying metadata information:* standards such as CERIF, used in EPOS, and ISO 19115, used in SeaDataNet, are also used for creating data catalogues.
4. *Providing different support for end-users to perform scientific experiments:* for example, ICOS provides a web-based environment, the Carbon Portal[10], to allow scientists to discover data, visualise its content, and perform customised data processing workflow, while LifeWatch provides specific deployments of software environments (virtual laboratories) to its users.

[9] https://www.force11.org/group/fairgroup/fairprinciples.
[10] https://www.icos-cp.eu/.

To be interoperable, the data or services from different RIs need to be discovered, accessed and integrated across their boundaries. It is important to identify the common problems faced by the RIs, and provide reusable solutions to those problems. To effectively deal with such issues, RI development faces a number of challenges:

1. How to effectively deal with the diversities, so that developers can identify and model the common problems faced by the RIs?
2. How to design reusable solutions to their common problems, so that each individual RI can effectively take the solution and customise it in their own software stacks?
3. How to effectively handle new requirements from each RI, e.g. demands from user communities?
4. How to effectively select technology and standards for prototyping the solutions to those common problems?

Based on those challenges, the ENVRI community proposed a reference guided approach, which we discuss in Sect. 4.

3 The State of the Art: Software Architecture and Development Models

In this section, we shall briefly review the software engineering technologies and methodologies from the perspectives of engineering model, software architecture, and reference model guidance.

3.1 Software Architecture

The architecture of a software system models the high-level structure of the system; the functional components and the logical relations among those components have been modelled using different orientations [10] e.g. of objects, components, software agents and services. Since 2000, service-oriented architecture has been widely adopted in the software industry for automating the cross-organization of business processes, hiding complexity in software delivery, and simplifying software reuse [11, 12]. In this context, a number of trends can be highlighted as arising during recent decades:

1. When running on virtualised infrastructure, loosely coupled distributed architectures are more scalable than the monolithic architectures in which all components reside in one integrated system;
2. Service-oriented architectures (SOA) are playing an increasingly important role in enterprise computing, and internet applications.
3. Web services can be deployed on remote hosts and can be invoked by remote clients via standardised internet-based protocols (e.g. HTTP). They can be implemented using Remote Procedure Call (RPC) based technologies, e.g. XML RPC or Simple Object Access Protocol (SOAP), or using Representational State Transfer (RESTful) mechanisms.

4. Microservices design the services in “suitable” granularity [13] with atomic functionality, which can be better reusable and scalable. The concept of microservice is typically driven by elastic computing in Cloud, where the required service function can be flexibly scaled out by adding more instances to overcome performance bottlenecks.

3.2 Reference Model and Architecture in System Development

Reference models or architecture have been widely in the IT industry to standardise the abstraction of certain new technologies, e.g. the OSI reference model for network development [8] and workflow management reference model [9] for business process management. A reference model for a computational system provides an ontological framework for involved parties to clearly communicate.

In both the ENVRI and ENVRIplus projects, a reference model has been recognised as a promising contribution for realising interoperability for diverse environmental RIs. In this section, we will first review the work of the ENVRI Reference Model, and then summarise the lessons learned. Afterwards, we will discuss the approach for the ENVRIplus Reference Model.

In the ENVRI project, the development of the Reference Model (ENVRI-RM) was based on an analysis of six RIs involved in the project: ICOS, Euro-Argo, EISCAT_3D, LifeWatch, EPOS, and EMSO. By interviewing specialists from each of these RIs, and examining the requirements, design documents, and use cases collected, we abstracted some common operations and design patterns. This analysis had to cope with different viewpoints and varying vocabularies between (and even within) RIs.

The methodology for developing ENVRI-RM was to decompose system descriptions based on viewpoints. Open Distributed Processing (ODP) [14] provides five viewpoints from which to describe systems: *enterprise* (about system scenarios, involved communities and roles), *computation* (about system interfaces and bindings between system components), *information* (about data objects and schemas of the system), *engineering* (about system middleware and engineering principles) and *technology* (technology standards and decisions). This decomposition of complex systems by viewpoint is a useful technique for managing complexity and providing information tailored to different kinds of stakeholders. ENVRI-RM employs these viewpoints to model the characteristics of environmental research infrastructures, but we replace the Enterprise viewpoint with a “Science” Viewpoint to align the ODP with the RI view of the world. The current version is available online[11] (Fig. 2).

ENVRI-RM focused on the design of a small set of RIs and was produced at a time when most of them were in their preparatory phase of development. Since ENVRI began, many of them have made significant progress in their development, to some extent exceeding the expressiveness of ENVRI-RM. As such, a number of lessons can be learned:

[11] www.envri.eu/rm.

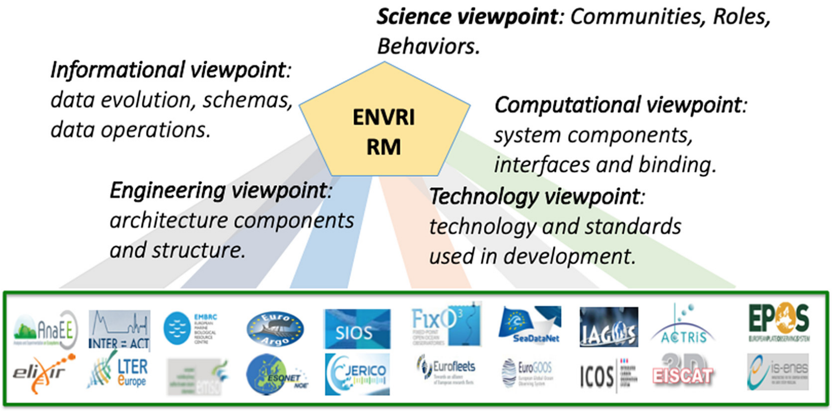

Science viewpoint: Communities, Roles, Behaviors.
Informational viewpoint: data evolution, schemas, data operations.
ENVRI RM
Computational viewpoint: system components, interfaces and binding.
Engineering viewpoint: architecture components and structure.
Technology viewpoint: technology and standards used in development.
AnaEE
INTER-ACT
EMBRC
Euro-Argo
SIOS
FixO3
SeaDataNet
IAGOS
ACTRIS
EPOS
elixir
LTER Europe
EMSO
ESONET
JERICO
Eurofleets
EuroGOOS
ICOS
EISCAT
is-enes

Fig. 2. The basic idea of ENVRI reference model.

1. The rapid evolution of environmental RIs was not sufficiently taken into account when building ENVRI-RM; there are mismatches between new insights in the RIs and what was encoded in the model.
2. Supporting use cases are required for validation and demonstration; the lack of these in ENVRI led to drifting requirements and difficulty explaining the model to potential users, although this was improved in ENVRIplus.
3. The development of the model did not involve enough domain-aware ICT specialists from the RIs themselves. This was partly due to the early development state of the RIs, but meant that the model was not really applied to that development.

3.3 Software Development Models

To efficiently manage the activities in the lifecycle of software development, different engineering models have been proposed and applied during recent decades. The waterfall model is a typical example, where requirement analysis, system design, software development, testing and integration, and delivery are organised sequentially. The development team focuses on a specific task at each stage. When the application problem is well understood and there is sufficient engineering time, the waterfall model is easy to apply in practice. However, when an application is difficult to describe precisely in the very beginning, or the time for delivery is fixed and urgent, e.g. when driven by specific market needs, the waterfall model exhibits a number of weaknesses: i) high cost in incorporating changing requirements or correcting mistakes, and ii) high risks in managing time because the project commonly is delayed if any mistakes are made at an earlier phase. The waterfall model has been adapted in different ways to overcome these issues:

1. The V model [15], in which the software testing and validation are performed against system design, architecture and requirements, as shown in Fig. 3-a.
2. The Iterative model [16], in which all phases in the lifecycle can provide feedback to the previous phase, and make corrections where necessary, as shown in Fig. 3-b;

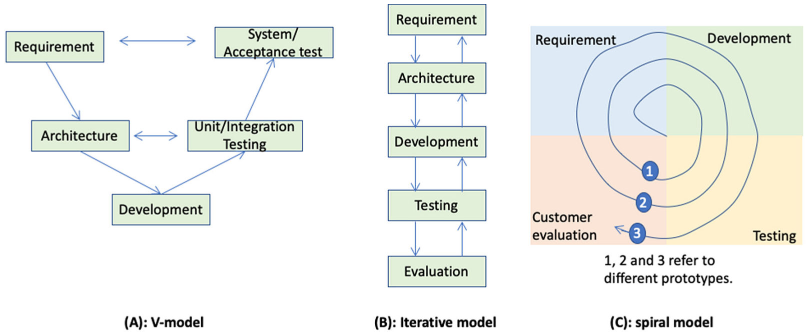


Fig. 3. Some example models for software development.

3. The Spiral model [17], in which the lifecycle is organised as a number of continuous phases, and each phase is a loop of all steps as defined in the waterfall model. The spiral model can reduce the risks of unbalanced time allocation and partial or inaccurate requirements analysis, as shown in Fig. 3-c.

In this evolution of software development models, we can clearly see several highlights: i) developers do not just execute engineering tasks sequentially and in a single round, ii) developers can flexibly switch engineering tasks forward or backward, and iii) the duration of the customer evaluation is also getting shorter. For applications which have clear time boundary and delivery constraints, a method called Agile development has emerged during the past decade, where the development team focuses on the prioritised tasks requested by the customer, and efficiently perform the development with well-controlled progress reviews. Highsmith [18] highlighted the key difference between classical waterfall model and the Agile model by using the relationships between **Feature(s), Cost** and **Time**. In the classic model, the set of **features** are derived from the requirements and commonly are fixed; the **timeline** and project **cost** often have to be adapted based on the original plan and the actual progress [21]. The Agile model is the opposite: the set of **features** has to be adaptable to meet the fixed **cost** and **timeline** of the project (Fig. 4).

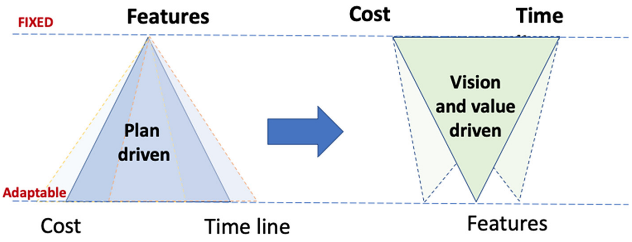


Fig. 4. Agile management: from plan driven to vision and value driven [18].

3.4 Summary

Targeting at the interoperability of more than 20 research infrastructures in the cluster of environmental and Earth science, the ENVRIplus data for science theme has to simultaneously interact with the development teams in each RI [23]. Within the period of the project, the theme developers had to continuously:

1. collect and analyse requirements from each RI,
2. tackle common challenges, and
3. deliver useful solutions to the development teams of the RIs, even while each RI clearly has its own development roadmap and timeline.

To effectively manage the development process of the theme team, and the interaction with individual RIs, the engineering approaches we reviewed above needed to be carefully selected and applied. A reference model guided approach was thus proposed.

4 The Reference Model Guided Approach

The **ENVRIplus reference model guided engineering model** builds upon abstracted concepts derived from analysing common operations of a selected set of RIs and subsequently defines an ontological reference model for all environmental RIs. Figure 5 shows the basic idea of the reference model guided approach.

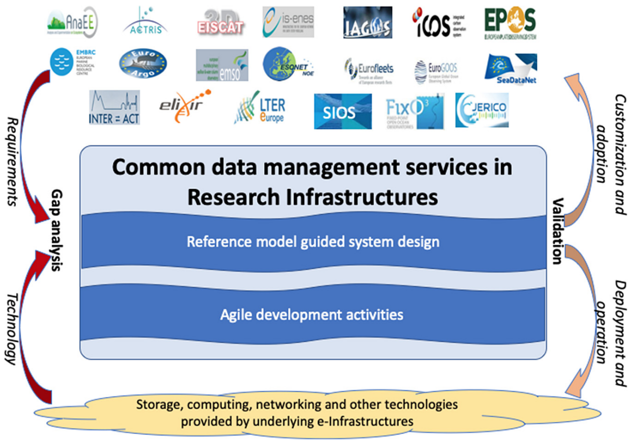

AnaEE
ACTRIS
EISCAT
is-enes
IAGOS
ICOS
EPOS
EMBRC
Euro-Argo
emso
ESONET
Eurofleets
EuroGOOS
SeaDataNet
INTER = ACT
elixir
LTER Europe
SIOS
FixO3
JERICO
Requirements
Gap analysis
Technology
Common data management services in Research Infrastructures
Reference model guided system design
Agile development activities
Validation
Customization and adoption
Deployment and operation
Storage, computing, networking and other technologies provided by underlying e-Infrastructures

Fig. 5. The basic idea of the reference guided approach.

The proposed approach uses the ENVRI-RM as the common ontological framework to:

1. formulate requirement collection questionnaires;
2. align the input acquired from different research infrastructures;

3. analyse the requirements from different viewpoints;
4. design and validate the solution using the architectural patterns provided by the reference model.

The development teams carried out the development tasks of the designed solution in an iterative way. In the meantime, a number of small parallel use case teams were dynamically established based on the demands and the priority of each solution development team. The use case projects were managed using the agile approach: via a dynamically maintained task list, the project teams aimed to deliver a rapid prototype or technical validation in a timely way. The successful results from the use case teams were curated and included regular development task teams interaction.

A high-level steering committee was established to control the selection of successful results and establish a portfolio for the entire theme.

In the rest of the chapter, we will discuss this approach in more detail.

4.1 Reference Model Guided: Requirement Collection, Technology Review and Gap Analysis

Based on the requirements collected from each of the four main environmental science domains and their respective RIs, we **identified and developed common operations**, by characterising RIs' individual current solutions with consideration given to underlying common technologies and engineering challenges. These individual operations will be characterised in terms of the engineering model, which will then be used in the design and implementation of common operations. The common operations are of two kinds: (a) those needed by any RI for data management, cataloguing, curation, provenance, analytics, visualisation; (b) those required for interoperation across RIs.

To benefit from existing technologies, we reviewed early results from specific RIs and interacted with computational e-infrastructures (such as EGI), data infrastructures (such as EUDAT[12]), and other initiatives (such as D4Science[13]) that work on related issues. We reviewed other interoperation technologies including CERIF [19] from EPOS for describing datasets, users, software, facilities, services and resources, and DCAT[14] for high-level exposure of basic dataset information.

This approach was used to (a) reduce risk; (b) maximise utilization of e-infrastructures in individual RIs developed with EC or other public funding; (c) provide an opportunity for convergence of ideas among the RIs without discarding work already done; and (d) maximise the chances of successful interoperation between environmental RIs, both technically and socially.

4.2 Identifying Common Data Management Services Using the ENVRI-RM

The ENVRI-RM assists in defining commonalities in the operations of environmental RIs, e.g. common services that support a particular subdomain of environment research,

[12] http://www.eudat.eu/.

[13] http://www.d4science.org/.

[14] http://www.w3.org/TR/vocab-dcat/.

or set of such sub-domains. ENVRIplus is not concerned with the unique services of a specific RI. The focus is on common services that are useful for significant subsets of environmental RIs.

We have identified six common concerns based on the demands of the RIs involved in ENVRIplus, which we will work to provide solutions for.

1. **Data identification and citation** requires the implementation of a common policy model for handling persistent identifiers for publishing and citing data. Moreover, services for assigning and handling identifiers and for retrieving data based on identifiers should also be provided.
2. **Interoperable data processing, monitoring and diagnosis** services make it significantly easier for scientists to aggregate data from multiple sources and to conduct a range of experiments and analyses upon those data. Expanding upon the data processing workflow modelled in ENVRI, this service focuses on the engineering aspects of managing the entire lifecycle of computing tasks and application workflows for efficient utilization of underlying e-infrastructure. In particular, the service enables scientists to enrich the data processing environment by injecting new algorithms to be reused by others.
3. **Performance optimization for big data science** is increasingly required in environmental science. ENVRIplus focused on high-level, generically-applicable optimization mechanisms for making decisions on resources, services, data sources and potential execution infrastructures, and on scheduling the execution of big data applications [22].
4. Data quality control and annotation were modelled as basic curation services in ENVRI-RM, although they have different (but related) requirements. **Self-adaptable data curation for system-level science** covers different levels of data. The service provided by ENVRIplus complies with data and metadata standards such as OASIS[15] and INSPIRE[16] and provides rich, interoperable metadata for geospatial semantic annotation. The quality of user experience, when checking the quality of data and when annotating different data using the aforementioned metadata standards, is explicitly modelled and considered in the development of curation services.
5. To perform complex data-driven experiments, scientists want simple but effective mechanisms to discover data recorded in catalogues and to integrate data into computing processes. An **interoperable data cataloguing** service provides interoperable solutions for accessing, retrieving and integrating data from different catalogues. The service extended the open search tools developed in the ENVRI project by reusing the latest technologies. It investigated key issues in interoperable cataloguing and metadata harmonization with consideration of other ongoing initiatives.
6. Higher-level data products provided by RIs have to be clearly reproducible. Therefore, provenance services that record the evolution of data by tracking each operation processed have to be further developed and integrated within existing RIs. A **cross-RI data provenance** service provides tracing services for data manipulation between

[15] https://www.oasis-open.org/.

[16] http://inspire.ec.europa.eu/.

different infrastructures. Standardised interfaces for querying, accessing and integrating provenance data will be realised, building on current standardization efforts such as W3C-PROV[17] or natively in CERIF as used in EPOS.

4.3 Reference Model Guided System Design

The architectural patterns defined in the ENVRI-RM provides an abstraction for designers to design a data management service. The current ENVRI-RM provides the following information:

1. Science viewpoint: different roles involved in the service, and the interaction among those roles via the service;
2. Information viewpoint: the data evolution in the service including data schemas and data objects, and the actions that modify those data objects;
3. Computational viewpoint: the binding among components in the service, including key computational objects, and the artefacts transferred among those objects;
4. Technology viewpoints: the standards and technologies to be employed in the service;
5. Engineering viewpoint: the architecture of the service. Currently, microservice-based architectures are highly recommended by the RM.

Using the patterns provided by the ENVRI-RM, a developer can model the basic interface of the data management service and identify the key internal components. Figure 6 presents a typical design scenario for infrastructure optimization service.

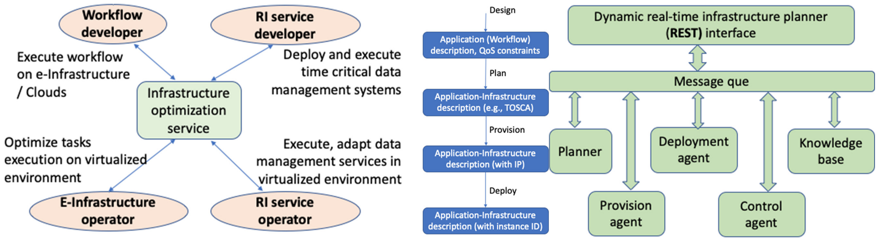

Workflow developer
RI service developer
Execute workflow on e-Infrastructure / Clouds
Deploy and execute time critical data management systems
Infrastructure optimization service
Optimize tasks execution on virtualized environment
Execute, adapt data management services in virtualized environment
E-Infrastructure operator
RI service operator
Design
Application (Workflow) description, QoS constraints
Plan
Application-Infrastructure description (e.g., TOSCA)
Provision
Application-Infrastructure description (with IP)
Deploy
Application-Infrastructure description (with instance ID)
Dynamic real-time infrastructure planner (REST) interface
Message que
Planner
Provision agent
Deployment agent
Control agent
Knowledge base

Fig. 6. Example of reference model guided design for infrastructure optimization service.

4.4 Agile Use Case Teams for Technology Investigation and Validation

The third step is **validation and service deployment**, deploying the implemented common operations within generic e-infrastructures (such as EGI or EUDAT), and operating them in the service of specific RIs. This approach aligns with ongoing work and trends in the provision of e-Infrastructure, especially grid-based (e.g. EGI), cloud-based (e.g. HELIX-Nebula) and data-centric projects (e.g. EUDAT), as well as the developments being proposed (and implemented) under the umbrella of Research Data Alliance

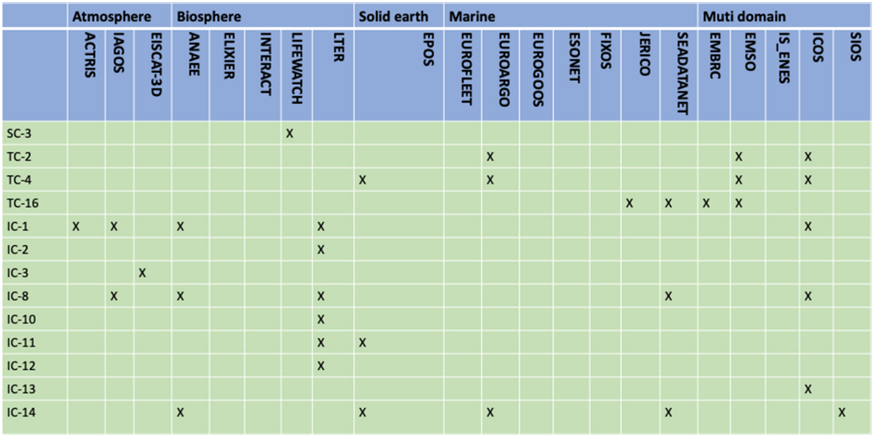
[17] http://www.w3.org/TR/prov-overview/.

(RDA)[18]. To enable the final usage of developed common services, the results will be tested and deployed in RIs, possibly via computing and data infrastructures such as EGI and EUDAT.

To engage the users of those data services in the loop in time, the requirements need to be formulated as "stories" and further elaborated as cases for the development teams. Based on the complexity of the cases, we identified three different levels: implementation cases, test cases, and science cases [20].

1. Implementation cases are relatively simple and can be finished in a relatively short time period. An implementation case often focuses on a specific feature of data management.
2. Test cases are those focusing on problem scenarios which require features from different services. Test cases are often bigger than implementation cases and need more time.
3. Science cases are often based on research problems which require data and services from different RIs. A science case can drive a number of test cases.

In a large project like ENVRIplus, more than 20 RIs participated in joint development activities. Cases were continuously collected and reviewed; the development teams of the specific data management services actively participated in the use cases, and established a use case project team, based on the Agile methodology, as explained in the next subsection (Fig. 7).



	Atmosphere			Biosphere					Solid earth	Marine							Muti domain				
	ACTRIS	IAGOS	EISCAT-3D	ANAEE	ELIXIER	INTERACT	LIFEWATCH	LTER	EPOS	EUROFLEET	EUROARGO	EUROGOOS	ESONET	FIXOS	JERICO	SEADATANET	EMBRC	EMSO	IS_ENES	ICOS	SIOS
SC-3							X														
TC-2											X							X		X	
TC-4									X		X							X		X	
TC-16															X	X	X	X			
IC-1	X	X		X				X												X	
IC-2								X													
IC-3			X																		
IC-8		X		X				X								X				X	
IC-10								X													
IC-11								X	X												
IC-12								X													
IC-13																				X	
IC-14				X					X		X					X					X

Fig. 7. The relation between use cases and specific domains. The complete list of use cases can be found in [20].

[18] https://rd-alliance.org/.

4.5 Coordinated Team Collaboration

In the ENVRIplus project, the development efforts were structured via different teams:

1. The developers for each common data service working for all RIs, rather than for one single RI, working on services for identification, processing, infrastructure optimization, curation, cataloguing and provenance.
2. The developers from each RI were identified. In many cases, these developers were distributed, due to the complexity of the infrastructure. These developers were responsible for developing and maintaining services in individual RIs.
3. Developers focusing on specific agile use case projects, which are created based on the dynamic needs of the RI communities.

Figure 8 depicts the interactions among the different teams.

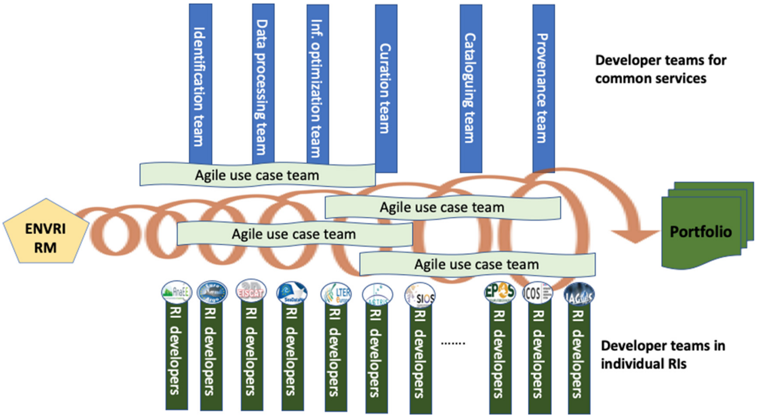


Fig. 8. How different developers interact in the engineering approach.

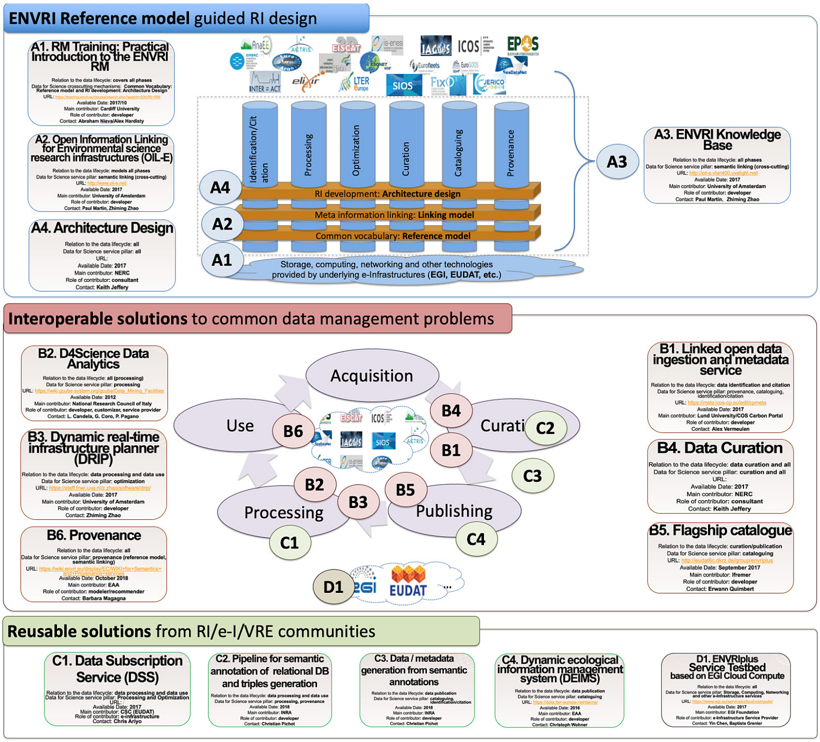
4.6 Portfolio Management

A service portfolio is a core repository that manages the evolution of the service and software assets that a company or organization delivers. It is an important strategy for the software industry to bridge the gaps among customer needs, development teams and the delivered software products (services). The portfolio is often broader than the service catalogue that an organization provides to the customer; it often contains the services to be developed, and inactive services after being replaced.

In the ENVRIplus project, the data for science theme adopts this strategy to manage the development plan of reusable solutions and use cases while interacting with the research infrastructures from different subdomains. We follow the practice from FITSM[19], based on the best practices from the e-Infrastructures EGI.

[19] https://www.fitsm.eu/.

In the ENVRIplus project, we organise the service portfolio in the data for science theme as four parts: 1) reference model related services and tools, 2) reusable solutions to common problems, 3) reusable solutions from use cases, and 4) testbeds. Figure 9 shows the basic idea.


ENVRI Reference model guided RI design
Identification/Citation
Processing
Optimization
Curation
Cataloguing
Provenance
A4
RI development: Architecture design
A2
Meta information linking: Linking model
Common vocabulary: Reference model
A1
Storage, computing, networking and other technologies provided by underlying e-Infrastructures (EGI, EUDAT, etc.)
A3
Interoperable solutions to common data management problems
Acquisition
Use
Curati
Processing
Publishing
B1
B2
B3
B4
B5
B6
C1
C2
C3
C4
D1
Reusable solutions from RI/e-I/VRE communities

A1. RM Training: Practical Introduction to the ENVRI RM
Relation to the data lifecycle: covers all phases
Data for Science crosscutting mechanisms: Common Vocabulary: Reference model and RI development: Architecture Design
URL: https://training.envri.eu/course/search.php?search=ENVRI+RM
Available Date: 2017/10
Main contributor: Cardiff University
Role of contributor: developer
Contact: Abraham Nieva/Alex Hardisty

A2. Open Information Linking for Environmental science research infrastructures (OIL-E)
Relation to the data lifecycle: models all phases
Data for Science service pillar: semantic linking (cross-cutting)
URL: http://www.oil-e.net/
Available Date: 2017
Main contributor: University of Amsterdam
Role of contributor: developer
Contact: Paul Martin, Zhiming Zhao

A4. Architecture Design
Relation to the data lifecycle: all
Data for Science service pillar: all
URL:
Available Date: 2017
Main contributor: NERC
Role of contributor: consultant
Contact: Keith Jeffery

A3. ENVRI Knowledge Base
Relation to the data lifecycle: all phases
Data for Science service pillar: semantic linking (cross-cutting)
URL: http://oil-e.vlan400.uvalight.net/
Available Date: 2017
Main contributor: University of Amsterdam
Role of contributor: developer
Contact: Paul Martin, Zhiming Zhao

B2. D4Science Data Analytics
Relation to the data lifecycle: all (processing)
Data for Science service pillar: processing
URL: https://wiki.gcube-system.org/gcube/Data_Mining_Facilities
Available Date: 2012
Main contributor: National Research Council of Italy
Role of contributor: developer, customizer, service provider
Contact: L. Candela, G. Coro, P. Pagano

B3. Dynamic real-time infrastructure planner (DRIP)
Relation to the data lifecycle: data processing and data use
Data for Science service pillar: optimization
URL: https://staff.fnwi.uva.nl/z.zhao/software/drip/
Available Date: 2017
Main contributor: University of Amsterdam
Role of contributor: developer
Contact: Zhiming Zhao

B6. Provenance
Relation to the data lifecycle: all
Data for Science service pillar: provenance (reference model, semantic linking)
URL: https://wiki.envri.eu/display/EC/WIKI+for+Semantics+and+Provenance+services
Available Date: October 2018
Main contributor: EAA
Role of contributor: modeler/recommender
Contact: Barbara Magagna

B1. Linked open data ingestion and metadata service
Relation to the data lifecycle: data identification and citation
Data for Science service pillar: provenance, cataloguing, identification/citation
URL: https://meta.icos-cp.eu/edit/cpmeta
Available Date: 2017
Main contributor: Lund University/COS Carbon Portal
Role of contributor: developer
Contact: Alex Vermeulen

B4. Data Curation
Relation to the data lifecycle: data curation and all
Data for Science service pillar: curation and all
URL:
Available Date: 2017
Main contributor: NERC
Role of contributor: consultant
Contact: Keith Jeffery

B5. Flagship catalogue
Relation to the data lifecycle: curation/publication
Data for Science service pillar: cataloguing
URL: http://eudat6c.dkrz.de/group/envriplus
Available Date: September 2017
Main contributor: Ifremer
Role of contributor: developer
Contact: Erwann Quimbert

C1. Data Subscription Service (DSS)
Relation to the data lifecycle: data processing and data use
Data for Science service pillar: Processing and Optimization
URL: /
Available Date: 2017
Main contributor: CSC (EUDAT)
Role of contributor: e-infrastructure
Contact: Chris Ariyo

C2. Pipeline for semantic annotation of relational DB and triples generation
Relation to the data lifecycle: data processing and data use
Data for Science service pillar: processing, provenance
Available Date: 2018
Main contributor: INRA
Role of contributor: developer
Contact: Christian Pichot

C3. Data / metadata generation from semantic annotations
Relation to the data lifecycle: data publication
Data for Science service pillar: cataloguing, identification/citation
Available Date: 2018
Main contributor: INRA
Role of contributor: developer
Contact: Christian Pichot

C4. Dynamic ecological information management system (DEIMS)
Relation to the data lifecycle: data publication
Data for Science service pillar: cataloguing
URL: https://data.lter-europe.net/deims/
Available Date: 2016
Main contributor: EAA
Role of contributor: developer
Contact: Christoph Wohner

D1. ENVRIplus Service Testbed based on EGI Cloud Compute
Relation to the data lifecycle: all
Data for Science service pillar: Storage, Computing, Networking and other e-Infrastructure services
URL: https://www.egi.eu/services/cloud-compute/
Available Date: 2017
Main contributor: EGI Foundation
Role of contributor: e-Infrastructure Service Provider
Contact: Yin Chen, Baptiste Grenier

Fig. 9. Service portfolio in the data for science theme.

Figure 9 depicts the structure of the snapshot of the service portfolio in 2018.

5 Summary

Conducting system-level environmental science research requires advanced systems for collecting, curating and providing access to scientific data products. Various environmental research infrastructures (RIs) are being constructed to address this requirement; however, there is no coherent standard approach to constructing interoperable RIs that would permit the kind of interdisciplinary research needed to fully exploit the data now being made available.

In this chapter, we discussed the reference model guided approach adopted in the ENVRIplus project. This approach provided a uniform way of characterising existing RIs to permit the definition of required common and cross-cutting (interoperation) services.

However, building the reference model for ENVRIplus was labour-intensive and there is an ongoing discussion of the cost-benefit. In the rest of the book, we will discuss more details of how this approach is applied in the context of different development teams.

Acknowledgements. This work was supported by the European Union's Horizon 2020 research and innovation programme via the ENVRIplus project under grant agreement No 654182.

References

1. Zhao, Z., et al.: Reference model guided system design and implementation for interoperable environmental research infrastructures. In: 2015 IEEE 11th International Conference on e-Science, Munich, Germany, pp. 551–556. IEEE (2015). https://doi.org/10.1109/eScience.2015.41
2. Foster, I., Kesselman, C.: Scaling system-level science: scientific exploration and IT implications. Computer **39**, 31–39 (2006)
3. Petzold, A., et al.: ENVRI-FAIR - interoperable environmental FAIR data and services for society, innovation and research. In: 15th IEEE International Conference on e-Science, San Diego, US (2019). https://doi.org/10.1109/eScience.2019.00038. https://zenodo.org/record/3462816
4. European Strategic Forum of Research Infrastructure (ESFRI) Roadmap. http://ec.europa.eu/research/infrastructures/index_en.cfm?pg=esfri. Accessed 03 Apr 2019
5. Zhao, Z., Grosso, P., de Laat, C.: OEIRM: an open distributed processing based interoperability reference model for e-science. In: Park, J.J., Zomaya, A., Yeo, S.-S., Sahni, S. (eds.) NPC 2012. LNCS, vol. 7513, pp. 437–444. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35606-3_52
6. Riedel, M., Laure, E., Soddemann, T., Field, L., et al.: Interoperation of world-wide production e-Science infrastructures. Concurr. Comput. Pract. Exp. **21**, 961–990 (2009)
7. White, L., et al.: Understanding interoperable systems: challenges for the maintenance of SOA applications. In: 2012 45th Hawaii International Conference on System Sciences, Maui, HI, USA, pp. 2199–2206. IEEE (2012)
8. ISO/IEC 7498-4:1989 – Information technology – Open Systems Interconnection – Basic Reference Model: Naming and addressing (1989)
9. Workflow reference model. http://www.wfmc.org/2-uncategorised/53-reference-model. Accessed 03 Apr 2019
10. Oussalah, M.C. (ed.): Software Architecture 1: Oussalah/Software Architecture 1. Wiley, Chichester (2014)
11. Carey, M.J.: SOA What? Computer **41**, 92–94 (2008)
12. Belqasmi, F., Singh, J., Bani Melhem, S.Y., Glitho, R.H.: SOAP-based vs. RESTful web services: a case study for multimedia conferencing. IEEE Internet Comput. **16**, 54–63 (2012)
13. Di Francesco, P., Lago, P., Malavolta, I.: Migrating towards microservice architectures: an industrial survey. In: 2018 IEEE International Conference on Software Architecture (ICSA), Seattle, WA, pp. 29–2909. IEEE (2018). https://doi.org/10.1109/ICSA.2018.00012
14. Linington, P.F. (ed.): Building Enterprise Systems with ODP: an Introduction to Open Distributed Processing. CRC Press, Boca Raton (2012)
15. Mathur, S., Malik, S.: Advancements in the V-Model. Int. J. Comput. Appl. (0975–8887), **1**(12), 29–34 (2010)

16. Choetkiertikul, M., Dam, H.K., Tran, T., Ghose, A., Grundy, J.: Predicting delivery capability in iterative software development. IEEE Trans. Softw. Eng. **44**, 551–573 (2018)
17. Boehm, B., Brown, W., Turner, R.: Spiral development of software-intensive systems of systems. In: Proceedings of the 27th International Conference on Software Engineering, ICSE 2005, St. Louis, MO, USA, pp. 706–707. IEEE (2005)
18. Highsmith, J.: Agile Project Management: Creating Innovative Products. Addison Wesley, Upper Saddle River (2010)
19. Jeffery, K., Houssos, N., Jörg, B., Asserson, A.: Research information management: the CERIF approach. Int. J. Metadata Semant. Ontol. **9**, 5 (2014)
20. Chen, Y., et al.: D 9.1. http://www.envriplus.eu/wp-content/uploads/2015/08/D9.1-Service-deployment-in-computing-and-internal-e-Infrastructures.pdf. Accessed 03 Apr 2019
21. Casale, G., et al.: Current and future challenges of software engineering for services and applications. CloudForward (2016). https://doi.org/10.1016/j.procs.2016.08.278
22. Koulouzis, S., et al.: Time-critical data management in clouds: challenges and a Dynamic Real-time Infrastructure Planner (DRIP) solution. Concurr. Comput. Pract. Exp. (2019). https://doi.org/10.1002/cpe.5269
23. Martin, P., Remy, L., Theodoridou, M., Jeffery, K., Zhao, Z.: Mapping heterogeneous research infrastructure metadata into a unified catalogue for use in a generic virtual research environment. Future Gener. Comput. Syst. **101**, 1–13 (2019). https://doi.org/10.1016/j.future.2019.05.076