Virtual Research Environments as-a-Service by gCube

Massimiliano Assante, Leonardo Candela, Donatella Castelli, Gianpaolo Coro, Lucio Lelii, Pasquale Pagano
Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" – Consiglio Nazionale delle Ricerche
via G. Moruzzi, 1 – 56124, Pisa, Italy
Email: {name.surname}@isti.cnr.it

***Abstract*—Science is in continuous evolution and so are the methodologies and approaches scientists tend to apply by calling for appropriate supporting environments. This is in part due to the limitations of the existing practices and in part due to the new possibilities offered by technology advances. gCube is a software system promoting elastic and seamless access to research assets (data, services, computing) across the boundaries of institutions, disciplines and providers to favour collaborative-oriented research tasks. Its primary goal is to enable *Hybrid Data Infrastructures* facilitating the dynamic definition and operation of *Virtual Research Environments*. To this end, it offers a comprehensive set of data management commodities on various types of data and a rich array of "mediators" to interface well-established Infrastructures and Information Systems from various domains. Its effectiveness has been proved by operating the D4Science.org infrastructure and serving concrete, multidisciplinary, challenging, and large scale scenarios. This paper gives an overview of the gCube system.**

***Keywords*—Virtual Research Environment; Social Networking; Science Gateway;**

I. Introduction

Science calls for ever innovative practices, tools and environments supporting the whole research lifecycle – from data collection and curation to analysis, visualisation and publishing [1], [2]. In particular, scientists are asking for integrated environments providing them with seamless access to data, software, services and computing resources they need in performing their research activities independently of organisational and technical barriers. In these settings, approaches based on ad-hoc and "from scratch" development are neither viable (e.g., high "time to market") nor sustainable (e.g., technological obsolescence risk).

The *as-a-Service* model [3], [4], consisting in outsourcing functionality to professional and dedicated providers, is a very promising trend for both science and beyond. This model makes it possible to leverage economies of scale to keep costs low, to count on a known quality of service and to scale service delivery to peak demands.

Depending on the typologies of service(s) that are made available by a service provider, a gap between the offering and the scientist expectations might occur in terms of research environments (functional mismatch). The consumption of the available services and their exploitation to realise the scientific workflows should be an easy task that does not require extra skills nor distract effort from the pure scientific investigation (long learning curves, "entry barriers"). Science Gateways (SGs) [5] and Virtual Research Environments (VREs) [6] have been proposed to close the gap between service providers and scientific communities, both the functional mismatch and the entry barriers. Very often these are ad-hoc portals built to serve the needs of a specific community only.

The development of such environments expected to facilitate scientists tasks is challenging from the system engineering perspective, several technologies and skills are needed [7], [8]. Moreover, (*a*) people having the requested expertise (mainly the IT ones) are not always available in the scientific contexts calling for VREs, and (*b*) technology is in continuous evolution thus offering new opportunities for implementing existing facilities in innovative ways or integrating innovative facilities in existing VREs. This should discourage scientific communities from building their own solutions and should suggest to outsource this task to approaches that delivers them with the as-a-Service model.

This paper gives an overview of gCube[1]. gCube is a software system specifically conceived to enable the creation and operation of an innovative typology of e-Infrastructure (*Hybrid Data Infrastructure*) that by aggregating a wealth of resources from other infrastructures offers cutting-edge *Virtual Research Environments as-a-Service*. gCube complements the offerings of the aggregated infrastructures by implementing a comprehensive set of value-added services supporting the entire data management lifecycle in accordance with collaborative, user friendly, and Open Science compliant practices. gCube supports the D4Science.org infrastructure that hosts more than 50 VREs, supports more than 2500 scientists in 44 countries; integrates more than 50 data providers, executes more than 25,000 models & processes per month, provide access to over a billion quality records, 20,000 temporal datasets, 50,000 spatial datasets.

II. gCube System Overview

In order to offer VREs as-a-Service, the gCube system has been designed according to a number of guiding principles described below.

Component orientation: gCube is primarily organised in a number of physically distributed and networked *services*. These services offer functionality that can be combined

[1] www.gcube-system.org

together. In addition, it consists of (*a*) auxiliary components (*software libraries*) supporting services development, service-to-service integration, and service capabilities extension, and (*b*) components dedicated to realise the user interface (*portlets*).

Autonomic behaviour: Some components are dedicated to manage the operation of a gCube-based infrastructure and its constituents, e.g., automatic (un-)deployment, relocation, replication. These components realise a middleware providing the resulting infrastructure with an autonomic behaviour that reduces its deployment and operation costs.

Openness: gCube supplies a set of generic frameworks supporting data collection, storage, linking, transformation, curation, annotation, indexing and discovery, publishing and sharing. These frameworks are oriented to capture the needs of diverse application domains through their rich adaptation and customisation capabilities [9].

System of Systems: gCube includes components realising a rich array of mediator services for interfacing with existing "systems" and their enabling technologies including middlewares for distributed computing (e.g., EMI [10]), cloud (e.g., Globus [11], OCCI [12]) and data repositories (e.g., OAI-PMH [13], SDMX [14]). Via these mediator services, the storage facilities, processing facilities and data resources of external infrastructures are conceptually unified to become gCube resources.

Policy-driven resources sharing: gCube manages a resource space where (*a*) resources include gCube-based services as well as third party ones, software libraries, portlets and data repositories, (*b*) resources exploitation and visibility is controlled by policies realising a number of overlay sets on the same resource space. This approach is key to have a flexible and dynamic mechanism for VRE creation, since they are realised as aggregations of resources.

As a Service: gCube offering is exposed by the "as a Service" delivery model [3]. The advantage is that the actual management is in the hand of expert operators who manage the infrastructure (*i*) to provide reliable services, (*ii*) by leveraging economies of scale, and (*iii*) by using elastic approaches to scale. Via gCube nodes (servers enriched with microservices) the system offers storage and computing capacities as well as service instances management (dynamic (un)deployment, accounting, monitoring, alerting). Via gCube APIs the system gives a flexible and powerful platform to which developers can outsource data management tasks. Via gCube services the system offers a number of ready to use applications.

These guiding principles allow providing VREs as-a-Service, i.e., authorised users can aggregate – by using a wizard – existing resources (including data) to form innovative working environments and make them available via a plain web browser or even via a thin client.

gCube key components resulting from the above principles are organised in four main areas (cf. Fig. 1). These areas are described in the next sections.

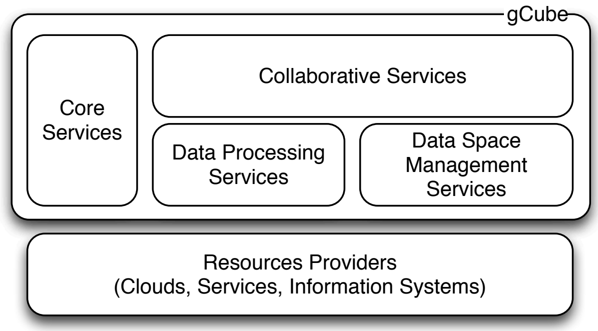

Fig. 1. gCube Functional Areas

A. Core Services

gCube core services offer basic facilities for resources management, security, and VRE management described below.

Resources Management: Facilities for the seamless management (discovery, deployment, monitoring, accounting) of *resources* (proprietary or third party ones) encompassing hosting nodes, services, software and datasets are offered. Included are (*a*) an *Information System* acting as a registry of the entire infrastructure and giving global and partial views of the resources and their operational state through query answering or notifications; (*b*) a *Resource Management Service* realising resource allocation and deployment strategies (e.g., dynamically assigning selected resources to a given context such as a VRE, assigning and activating both gCube software and external software on hosting nodes). (*c*) a *Hosting Node*, a software component that once installed on a (virtual) machine transforms it into a server managed by the infrastructure and makes it capable to host running instances of services and manage their lifecycle. This type of node can be configured to host a worker service thus making the machine capable to execute computing tasks (cf. Sec. II-C).

Security Framework: Facilities for authentication and authorisation are supported. They are based on standard protocols and technologies (e.g., SAML 2.0) providing: (*a*) an open and extensible AA architecture; (*b*) interoperability with external infrastructures and domains obtaining Identity Federation (e.g., OpenID). For authorisation, gCube implements a token based authorization system with an attribute-based access control paradigm. For authentication, users are requested to sign-in with their account (including third-party accounts like Google or LinkedIn). Once logged in, the user is provided with a user token that it transparently used to perform calls on behalf of the user. Whenever a user action implies a call to a service requiring authorization, gCube security framework (*a*) automatically collects the credentials to use from the credentials wallet, i.e., a service where the credentials to access each service are stored in encrypted form by using VRE-specific symmetric keys, (*b*) decrypts them with the specific VRE key, and (*c*) performs the authorized call by passing the credentials and the user token. In this way the connection to the service is established in a secure way while the token is used to verify that the specific user is authorized to call the service.

VRE Management: Facilities for the specification (wizard-based) and automatic deployment of complete VREs in terms of the data and services they should offer are supported [15]. These facilities are implemented by dynamically acquiring and aggregating the needed resources, including user interface constituents, from the resource space. This is a very straightforward activity consisting of: (*i*) a *design* phase where authorised users are provided with a wizard-based approach to specify the data and the services characterising the envisaged environment by selecting among the available ones; (*ii*) a *deployment* phase where authorised users are provided with a wizard-based approach to approve a VRE specification and monitor the automatic deployment of the real components needed to satisfy the specification; and (*iii*) an *operation* phase where authorised users are provided with facilities for managing the users of the VRE and altering the VRE specification if needed. Details on this approach have been presented in previous works [15], [16] while a screenshot of the wizard supporting the VRE specification is in Figure 2.

B. Data Space Management Services

Data occupies a key role in science and scientific workflows, thus VREs are called to support their effective management. However, data to be managed are very heterogeneous (formats, typologies, semantics), disaggregated and dispersed in multiple sites (including researchers drawers), possibly falling under the big data umbrella. In the context of a VRE, it is likely that compound information units are produced by using constituents across the various solutions. To cope with this variety, gCube offers an array of solutions ranging from those aiming at abstracting from the heterogeneity of data (file-oriented and information objects) to those focusing on specific data typologies having different levels of semantic embodiment (tabular, spatial, biodiversity data). Independently of the data typologies, all these solutions are characterised by (*a*) support for aggregation of data residing in existing repositories; (*b*) scalability strategies enabling to dynamically add more capacity; (*c*) comprehensive metadata to capture key aspects like context, attribution, usage licenses, lineage [17]; and (*d*) policy-driven configurability to adapt the data space to specific needs, e.g., by selecting the repositories and datasets. In particular, the following facilities are supported.

File-oriented data: The *Storage Manager* is a Java based software library supporting a unique set of methods for services and applications to manage files efficiently. It relies on a network of distributed storage nodes managed by specialized open-source software for document-oriented databases. In its current implementation, three possible document store systems can be seamlessly used [18], MongoDB, Terrastore and U.STORE [19], while new ones can be added by implementing a specific mediator.

Information Objects: The *Home Library* is a Java based software library enabling objects consisting of a tree of nodes with associated properties. It is compliant with Java Content Repository API and implemented by relying on Apache Jackrabbit for the object structure and node properties, while the node content is outsourced to other services. Every data to be managed in a VRE has a manifestation in terms of Information Objects. The unification of the entire data space makes it possible to realise services across the boundaries of specific data typologies. Among these unifying services there is an innovative search engine [20] that makes it possible to seamlessly discover objects in the data space.

Tabular data: The *Tabular Data Manager* offers a comprehensive and flexible working environment for accessing, curating, analysing and publishing tabular data. It enables a user to ingest data – from a file or a web location – that are represented in formats including CSV, JSON and SDMX. In the case of "free" formats, namely CSV, the system offers facilities to transform data in a well-defined table format where the types of the columns are basic data types including temporal and spatial dimensions as well as references to controlled vocabularies, e.g., Code List. Tables formats can be defined in an interactive way as well as by relying on templates. Besides constraints on table column types, templates can contain additional validation rules as well as specification of data operations to be performed when an error occurs. Once a tabular data resource is created, it can be manipulated and analysed by benefitting of well known tabular data operations, e.g., adding columns, filtering, grouping, as well as advanced data analytics tasks (cf. Sec. II-C). To guarantee a proper and real time management of the data lineage, the service relies on an underlying cluster of RDBMs where any operation on a tabular dataset leads to a new referable version.

Spatial data: gCube owns services realising the facilities of a *Spatial Data Infrastructure* (SDI) by relying on state-of-the-art technologies and standards [21]. It offers standard-based services for data discovery (a catalogue), storage and access (a federation of repositories), and visualisation (a map container). The catalogue service enables the discovery of geospatial data residing in dedicated repositories by relying on GeoNetwork and its indexing facilities. For data storage and access, gCube offers a federation of repositories based on GeoServer and THREDDS technologies. In essence, the infrastructure hosts a number of repositories and a *GIS Publisher Service* that enables a seamless publication of geospatial data while guaranteeing load balancing, failure management and automatic metadata generation. It relies on an open set of back-end technologies for the actual storage and retrieval of the data. Because of this, the GIS Publisher Service is designed with a plug-in-oriented approach where each plug-in interfaces with a given back-end technology. To enlarge the array of supported technologies it is sufficient to develop a dedicated plug-in. Metadata on available data are published by the catalogue. For data visualisation, the infrastructure offers *Geo Explorer* and *GIS Viewer*, two components dedicated to support the browsing and visualisation of geospatial data. In particular, the Geo Explorer is a web application that allows users to navigate, organize, search and discover layers from

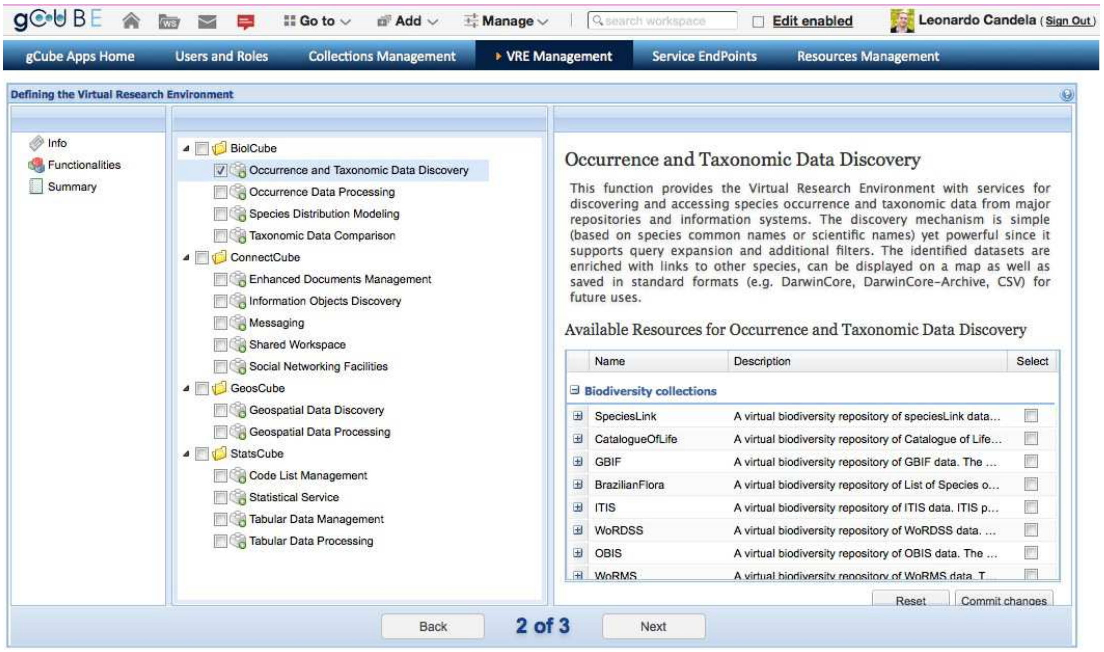

Fig. 2. Virtual Research Environment definition phase: selecting the expected facilities

the catalogue via the CSW protocol. The GIS Viewer is a web application that allows users to interactively explore, manipulate and analyse geospatial data.

Biodiversity data: The *Species Data Discovery and Access Service* (SDDA) [22] provides users with facilities for the management of nomenclature data and species occurrences. As data are stored in authoritative yet heterogeneous information systems, SDDA is mainly conceived to dynamically "aggregate" data from them and unify their management. It is designed with a plug-in based architecture. Each plug-in interacts with an information system or database by relying on a standard protocol, e.g., TAPIR, or by interfacing with its proprietary protocol. Plug-ins conform queries and results from the query language and data model envisaged by SDDA to the features of a particular database. SDDA promotes a unifying discovery and access mechanism based on the names of the target species, whether the scientific or the common ones. To overcome the potential issues related to taxonomy heterogeneities across diverse data sources, the service supports an automatic query expansion mechanism, i.e., upon request the query is augmented with "similar" species names. Also, queries can be augmented with criteria aiming at explicitly selecting the databases to search in and the spatial and temporal coverage of the data. Discovered data are presented in a homogenised form, e.g., in a typical Darwin Core format.

C. Data Processing Services

In addition to the facilities for managing datasets, VREs offer services for their processing. It is almost impossible to figure out "all" the processing tasks needed by scientists, thus the solution is to have environments where scientists can easily plug and execute their tasks. gCube offers two typologies of engines: one oriented to enact tasks executions at system level, another oriented to enact task execution at user level. Both of them are conceived to rely on a distributed computing infrastructure to execute tasks. A description of these two engines is given below.

System oriented workflow engine: The *Process Execution Engine* (PEng) is a system orchestrating flows of invocations (processes). It builds on principles of data flow processing appropriately expanded in the direction of interoperability [23]. According to this, PEng includes the plan (flow of execution), the operators (executable logic), the transport and control abstraction, the containers (areas of execution), the *state* holders (e.g., storage), the *resource profiles* (definitions of resources characteristics for exploitation in a plan). It allows to distribute jobs on several machines. Each job defines an atomic execution of a more complex process. Relations and hierarchies among the jobs are defined by means of Direct Acyclic Graphs (DAG). The DAGs are statically defined according to the Job Description Language (JDL) specifications [24].

Data analytics engine: The *Statistical Manager* (SM) is conceived to provide end users with an environment to execute computational analysis of datasets through both service provided algorithms and user defined algorithms [25]. At VRE creation time, the SM is configured with respect to the algorithms to be offered in that context. The SM currently supplies more than 100 ready to use algorithm implementations which include real valued features clustering, functions and climate scenarios simulations, niche modelling, model performance evaluation, time series analysis, and analysis of marine species and geo-referenced data. New algorithms can be easily integrated, in fact the SM comes with a development framework dedicated to this. A scientist willing to integrate a new algorithm should develop it by implementing some basic Java interface defining algorithm's inputs and outputs. In the case of non-Java software, e.g., R scripts, the framework provides functionalities to invoke it as external software. Integrated algorithms can be shared with coworkers by simply making them publicly available. The SM is designed to operate as a federation of SM instances, all sharing the same capabilities in terms of algorithms. Depending on the characteristics of the algorithm and the data, each SM instance executes the algorithm locally or outsources part of it to the underlying infrastructure including gCube workers (cf. Sec. II-A). A queue-based messaging system dispatches information about the computation, which includes (*i*) the location of the software containing the algorithm to be downloaded on the nodes and then executed, (*ii*) the subdivision of the input data space, which establishes the portion of the input to assign to each node, (*iii*) the location of the data to be processed, (*iv*) the algorithm parameters. Workers are data and software agnostic, which means that when ready to perform a task, they consume information from the queue and execute the software in a sandbox passing the experimental parameters as input. SM instances and workers share a data space for input and output consisting of a RDBMS, the Storage Manager, and the Home Library (cf. Sec. II-B).

D. Collaborative Services

VREs are easy to use (e.g., requested skills do not exceed the average scientist's ones), have limited adoption costs (e.g., no software to be installed), look like an integrated whole (e.g., the boundaries of the constituents are not perceived), and have an added value with respect to the single constituent's capabilities (e.g., simplify data exchange).

gCube offers its VREs via thin clients, e.g., a plain web browser. All the facilities so far described are made consumable via specific components, i.e., *portlets*, that are web-based user interface constituents conceived to be aggregated, configured and made available by a portal at VREs creation.

To complete its offering and provide its users with added value services, gCube equips its VREs with a social networking area.

Social Networking: gCube offers facilities promoting innovative practices that are compliant with Open Science [26]. Among the services there is a *Home Social* resembling a social network timeline where VRE users as well as applications can post messages, information objects, processing results and files. Such posts can be discussed and favourited (or blamed) by VRE members. Every member is also provided with a *Workspace*, i.e., a folder-based virtual "file system" allowing complex information objects, including files, datasets, workflows, and maps. Objects residing in the workspace can pre-exist the VRE or be created during the VRE lifetime, all of them are managed in a simple way (e.g., drag & drop), can be downloaded as well as shared in few clicks.

III. Related work

Virtual Research Environments, Science Gateways, Virtual Laboratories and other similar terms [6] are used to indicate web-based systems emerged to provide researchers with integrated and user friendly access to data, computing and services of interest for a given investigation that are usually spread across many and diverse data and computing infrastructures. Moreover, they are conceived to enact and promote collaboration among their members for the sake of the investigation. There are many frameworks that can be used to build such systems. Shahand et al. [8] have recently identified eleven frameworks explicitly exploited to develop Science Gateway including Apache Airavata, Catania SG Gateway, Globus, HUBzero(+Pegasus), ICAT Job Portal, and WS-PGRADE/gUSE. Such frameworks are quite diverse, e.g., Apache Airavata offers its facilities via an API while the Catania SG Gateway offers its facilities via a GUI and a RESTful API. However, they share certain characteristics that make them operate at a lower level of abstraction with respect to the one of gCube. For data management, these frameworks mainly focus on files while gCube tries to capture an extensive domain offering specific services (cf. Sec. II-B). Moreover, such specific services are conceived to make it easy to collect data from / interface with existing data providers thus to make their content available to VRE members. For data processing, the frameworks analysed by Shahand et al. focus on executing jobs while the gCube Data analytics engine (cf. Sec. II-C) complements this key yet basic facility with mechanisms enabling scientists to easily plug their methods into an environment transparently relying on distributed computing solutions. Moreover, every single algorithm once successfully integrated is automatically exposed with a RESTful API (OGC Web Processing Service) thus making it possible to invoke it by workflows. Finally, the mechanism gCube offers for the creation of a VRE is unique (cf. Sec. II-A). In essence, authorised users can simply create a new VRE via a wizard driving them to produce a characterisation of the needed environment in terms of existing resources. The software (including GUI constituents) and the data needed to satisfy the VRE specification are automatically deployed, no sysadmin intervention is needed.

IV. Conclusions

gCube as a whole is a unique system since it covers the entire spectrum of facilities needed to deliver scientific applications as-a-Service. These gCube enabled applications are actually an integrated web-based environment resulting from the aggregation of an open set of constituents. Its data space management and data processing facilities are built by cutting-edge technologies yet complementing them thus to comply with the challenges arising in scientific domains.

The experiences made while exploiting gCube to operate the D4Science.org infrastructure somehow demonstrate that the principles governing the VREs delivery and the system openness are key in the modern science settings [27]. The supported Virtual Research Environments serve diverse domains ranging from biodiversity [28] to environmental sciences [29], humanities research [30], and geosciences. The currently supported VREs are available via dedicated portals, some of these VREs are openly available for exploitation and test.[2].

Acknowledgment

The authors would like to thank the colleagues of the team contributing to the implementation of gCube. The work reported has been partially supported by the *iMarine* project (FP7 of the European Commission, FP7-INFRASTRUCTURES-2011-2, Grant agreement No. 283644), the *BlueBRIDGE* project (European Union's Horizon 2020 research and innovation programme, Grant agreement No. 675680), and the *ENVRIplus* project (European Union's Horizon 2020 research and innovation programme, Grant agreement No. 654182).

References

[1] T. Hey, S. Tansley, and K. Tolle, *The Fourth Paradigm: Data-Intensive Scientific Discovery*, T. Hey, S. Tansley, and K. Tolle, Eds. Microsoft Research, 2009.

[2] S. Bartling and S. Friesike, "Towards another scientific revolution," in *Opening Science*. Springer International Publishing, 2014, pp. 3–15.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672

[4] B. Allen, J. Bresnahan, L. Childers, I. Foster, G. Kandaswamy, R. Kettimuthu, J. Kordas, M. Link, S. Martin, K. Pickett, and S. Tuecke, "Software as a service for data scientists," *Communications of the ACM*, vol. 55, no. 2, pp. 81–88, 2012.

[5] S. Gesing and N. Wilkins-Diehr, "Science gateway workshops 2014 special issue conference publications," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 16, pp. 4247–4251, 2015.

[6] L. Candela, D. Castelli, and P. Pagano, "Virtual research environments: an overview and a research agenda," *Data Science Journal*, vol. 12, pp. GRDI75–GRDI81, 2013.

[7] K. A. Lawrence, N. Wilkins-Diehr, J. A. Wernert, M. Pierce, M. Zentner, and S. Marru, "Who cares about science gateways? a large-scale survey of community use and needs," in *9th Gateway Computing Environments Workshop*, 2014, pp. 1–4.

[8] S. Shahand, A. H. C. van Kampen, and S. D. Olabarriaga, "Science gateway canvas: A business reference model for science gateways," in *SCREAM '15 Proceedings of the 1st Workshop on The Science of Cyberinfrastructure: Research, Experience, Applications and Models*, 2015.

[9] F. Simeoni, L. Candela, D. Lievens, P. Pagano, and M. Simi, "Functional adaptivity for Digital Library Services in e-Infrastructures: the gCube Approach," in *13th European Conference on Research and Advanced Technology for Digital Libraries, ECDL 2009*, 2009.

[10] C. Aiftimiei, A. Aimar, A. Ceccanti, M. Cecchi, A. Di Meglio, F. Estrella, P. Fuhrmam, E. Giorgio, B. Konya, L. Field, J. Nilsen, M. Riedel, and J. White, "Towards next generations of software for distributed infrastructures: The european middleware initiative," in *E-Science (e-Science), 2012 IEEE 8th International Conference on*, 2012.

[11] R. Ananthakrishnan, K. Chard, I. Foster, and S. Tuecke, "Globus platform-as-a-service for collaborative science applications," *Concurrency and Computation: Practice and Experience*, vol. n/a, p. n/a, 2014.

[12] A. Edmonds, T. Metsch, A. Papaspyrou, and A. Richardson, "Toward an open cloud standard," *IEEE Internet Computing*, vol. 16, no. 4, pp. 15–25, 2012.

[13] C. Lagoze and H. Van de Sompel, "The open archives initiative: building a low-barrier interoperability framework," in *Proceedings of the first ACM/IEEE-CS Joint Conference on Digital Libraries*. ACM Press, 2001, pp. 54–62.

[2] http://services.d4science.org/ offers an up to date list of gCube-based Virtual Research Environments.

[14] SDMX Development Core Team, *SDMX: Statistical Data and Metadata Exchange*.

[15] M. Assante, L. Candela, D. Castelli, L. Frosini, L. Lelii, P. Manghi, A. Manzi, P. Pagano, and M. Simi, "An Extensible Virtual Digital Libraries Generator," in *12th European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, 2008, pp. 122–134.

[16] M. Assante, P. Pagano, L. Candela, F. De Faveri, and L. Lelii, "An approach to virtual research environment user interfaces dynamic construction," in *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL 2011)*. Springer, 2011, pp. 101–109.

[17] Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance in e-science," *SIGMOD Rec.*, vol. 34, no. 3, pp. 31–36, Sep. 2005. [Online]. Available: http://doi.acm.org/10.1145/1084805.1084812

[18] R. Cattell, "Scalable SQL and NoSQL data stores," *SIGMOD Rec.*, vol. 39, no. 4, pp. 12–27, May 2011.

[19] F. A. Durão, R. E. Assad, A. F. Silva, J. F. Carvalho, V. C. Garcia, and F. A. M. Trinta, "USTO.RE: A Private Cloud Storage System," in *13th International Conference on Web Engineering (ICWE 2013) - Industry track*, Aalborg, 2013.

[20] F. Simeoni, L. Candela, G. Kakaletris, M. Sibeko, P. Pagano, G. Papanikos, P. Polydoras, Y. E. Ioannidis, D. Aarvaag, and F. Crestani, "A Grid-Based Infrastructure for Distributed Retrieval," in *11th European Conference on Research and Advanced Technology for Digital Libraries*, 2007, pp. 161–173.

[21] M. Selamat, M. S. Othman, N. H. M. Shamsuddin, N. I. M. Zukepli, and A. F. Hassan, "A review on open source architecture in geographical information systems," in *Computer Information Science (ICCIS), 2012 International Conference on*, vol. 2, June 2012, pp. 962–966.

[22] L. Candela, D. Castelli, G. Coro, L. Lelii, F. Mangiacrapa, V. Marioli, and P. Pagano, "An infrastructure-oriented approach for supporting biodiversity research," *Ecological Informatics*, vol. 26, pp. 162–172, 2014.

[23] M. M. Tsangaris, G. Kakaletris, H. Kllapi, G. Papanikos, F. Pentaris, P. Polydoras, E. Sitaridi, V. Stoumpos, and Y. E. Ioannidis, "Dataflow processing and optimization on grid and cloud infrastructures," *IEEE Data Eng. Bull.*, vol. 32, no. 1, pp. 67–74, 2009.

[24] E. Laure, S. M. Fisher, A. Frohner, C. Grandi, P. Kunszt, A. Krenek, O. Mulmo, F. Pacini, F. Prelz, J. White, M. Barroso, P. Buncic, F. Hemmer, A. Di Meglio, and A. Edlund, "Programming the grid with glite," *Computational Methods in Science and Technology*, vol. 12, no. 1, pp. 33–45, 2006.

[25] G. Coro, L. Candela, P. Pagano, A. Italiano, and L. Liccardo, "Parallelizing the execution of native data mining algorithms for computational biology," *Concurrency and Computation: Practice and Experience*, vol. n/a, no. n/a, p. n/a, 2014.

[26] M. Assante, L. Candela, D. Castelli, P. Manghi, and P. Pagano, "Science 2.0 repositories: Time for a change in scholarly communication," *D-Lib Magazine*, vol. 21, no. 1/2, 2015.

[27] L. Candela, D. Castelli, A. Manzi, and P. Pagano, "Realising Virtual Research Environments by Hybrid Data Infrastructures: the D4Science Experience," in *International Symposium on Grids and Clouds (ISGC) 2014 23-28 March 2014, Academia Sinica, Taipei, Taiwan, PoS(ISGC2014)022*, ser. Proceedings of Science, 2014.

[28] R. Amaral, R. M. Badia, I. Blanquer, R. Braga-Neto, L. Candela, D. Castelli, C. Flann, R. De Giovanni, W. A. Gray, A. Jones, D. Lezzi, P. Pagano, V. Perez-Canhos, F. Quevedo, R. Rafanell, V. Rebello, M. Sousa-Baena, and E. Torres, "Supporting biodiversity studies with the EUBrazilOpenBio Hybrid Data Infrastructure," *Concurrency and Computation: Practice and Experience*, vol. n/a, p. n/a, 2014.

[29] Y. Legre, "ENVRI, integrated infrastructures, environmental research in harmony," *International Innovation*, 2012.

[30] T. Blanke, L. Candela, M. Hedges, M. Priddy, and F. Simeoni, "Deploying general-purpose virtual research environments for humanities research," *Philosophical Transactions of the Royal Society A*, vol. 368, pp. 3813–3828, 2010.