Technical Report, No. 2008-TR-035, 2008
Consiglio Nazionale delle Ricerche - Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"

gCube v1.0: A Software System for Hybrid Data Infrastructures

Leonardo Candela*, Donatella Castelli, Pasquale Pagano

Abstract
gCube is a software system enabling the realization and operation of hybrid data infrastructures capable of supporting the notion of Virtual Digital Libraries (VDLs) / Virtual Research Environments (VREs), i.e., collaborative, web-based working environments through which community of practices members are provided with the entire spectrum of resources (data, services, and computing facilities) needed to accomplish their research tasks in an effective and efficient manner. gCube is currently used to govern the infrastructure set up by the European integrated project D4Science (Distributed colLaboratories Infrastructure on Grid Enabled Technology 4 Science).

Keywords
Hybrid Data Infrastructure — Virtual Research Environment — Virtual Digital Library — Interoperability — Open source

[1] *Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", Consiglio Nazionale delle Ricerche, Via G. Moruzzi 1, 56124, Pisa, Italy*
***Corresponding author**: leonardo.candela@isti.cnr.it

Contents

Introduction

gCube [1] is a software system specifically conceived to enable the creation and operation of an innovative typology of infrastructure that by leveraging Grid [2], Digital Library [3] and Service-orientation [4] principles and approaches is delivering a number of data management facilities *as-a-Service*. One of its distinguishing feature is the orientation to serve the needs of diverse *Communities of Practice* [5] by providing each of them with a dedicated, flexible, ready-to-use, web-based working environment, named *Virtual Digital Library* [6] or *Virtual Research Environment* [7].

From the technological point of view, gCube provides: (*i*) runtime and design frameworks for the development of services that can be outsourced to a Grid-enabled infrastructure; (*ii*) a service-oriented Grid middleware for exploiting the Grid and hosting Web Services on it; (*iii*) a set of application services for distributed information management and retrieval of structured and unstructured data.

Runtime frameworks are distinguished workflows that are partially pre-defined within the system; they include Grid-enabled services and application services, where the former coordinate in a pure distributed way the action of the latter, while relying on a high-level characterization of their semantics. Design frameworks consist of patterned blueprints, software libraries and partial implementations of state-of-the-art application functionality, which can be configured, extended and instantiated into bespoke application Grid services.

The service-oriented Grid middleware provides all the required capabilities necessary to manage Grid infrastructures. It eliminates manual service deployment overheads, guarantees optimal placement of services within the infrastructure and opens unique opportunities for outsourcing state-of-the-art implementations. Rather than interfacing with the infrastructure, the software which implements the application services is literally handed over to it, so as to be transparently deployed across its constituent nodes according to functional constraints and quality-of-service (QoS) requirements. By integrating the gLite system [8] released by the Enabling Grid for E-sciencE (EGEE) project for batch processing and management of unstructured data, gCube also allows the large computing and storage facilities provided by the EGEE infrastructure to be properly exploited. With over 20,000 CPUs and 5 million Gigabytes of storage, EGEE is the largest operational Grid infrastructure ever built.

gCube application services offer a full platform for distributed hosting, management and retrieval of data and information, and a framework for extending state-of-the-art indexing, selection, fusion, extraction, description, annotation,

transformation and presentation of content. gCube is equipped with services for manipulating information objects, importing external objects, managing their metadata in multiple formats, securing the information objects to prevent unauthorized access, and transparently managing replication and partition on the Grid.

The remainder of this report is organised as follows. Section 1 describes the architecture of the whole system. Section 2 report information on how to obtain and use the gCube software system. Section 3 provides a set of objective indicators about the gCube technology. Section 4 documents the list of institutions contributing to the gCube system. Section 5 describes success stories and potential exploitation scenarios.

1. Architecture

The gCube system architecture is depicted in Figure 1. It results by combining in a Service Oriented Architecture a number of subsystems.

Such subsystems are organised in a conceptually three-tier architecture consisting of a runtime-environment, a set of enabling services, and a set of application services.

The *gCube run-time environment* is the set of subsystems each gCube empowered machine is equipped with. These subsystems form the platform for the hosting and operation of the rest of system constituents. It is named gCube Hosting Node environment or simply gHN. Namely, it consists of the gCube Container (to run gCube Services), the gCore Framework, named gCF (to reinforce the gCube Container in supporting the operation of gCube Services), a number of local services, namely Deployer, gHNManager, and Delegation, and a number of libraries and stubs needed to manage the communication with all other gCube services [9].

The *gCube Infrastructure Enabling Services* is the set of subsystems constituting the backbone of the gCube system and responsible to implement (*i*) the operation of an infrastructure supporting resources sharing and (*ii*) the definition and operation of Virtual Research Environments;

The *gCube Application Services* is the set of subsystems implementing facilities for (*i*) storage, organisation, description and annotation of information in a VRE (Information Organisation Services), (*ii*) retrieval of information in the context of a VRE (Information Retrieval Services) and (*iii*) provision of VO and VRE users with an interface for accessing such an infrastructure.

The overall architecture has been designed following the Service Oriented Architecture principles:

- the main constituents of each subsystem are expected to be loosely-coupled Web Services (actually WSRF services [10]);
- the constituents of the gCube-based infrastructure are discovered thanks to the Information System subsystem that, as usual, becomes fundamental to guarantee the operation of the rest;

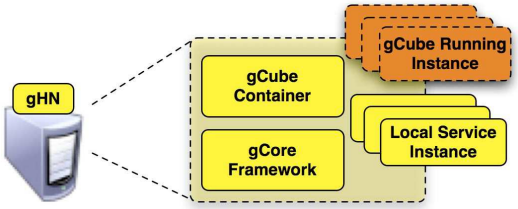

Figure 2. gHN Reference Architecture

- such loosely-coupled Services can be organised in workflows as to form compound services whose orchestration is guaranteed by the Process Management subsystem.

It is worth noting in this reference architecture that the runtime environment is an integral part of the overall system because the management of the environment hosting the services and the management of the service lifetime is part of the gCube business logic. Thanks to the gHN capabilities, other gCube services can be dynamically deployed on remotely gHNs to serve the needs of Virtual Research Environments. Figure 2 presents the *gCube Hosting Node (gHN)* Reference Architecture.

In the remainder of this section the constituents of the Reference Architecture are introduced starting from the lower layer.

1.1 The gCore Framework

The gCore Framework (gCF) is a Java framework for the development of high-quality gCube services and service clients. It provides an application framework that allows gCube services to abstract over functionality lower in the web services stack (WSRF, WS Notification, WS Addressing, etc.) and to build on top of advanced features for the management of state, scope, events, security, configuration, fault, service lifetime, and publication and discovery.

Its main design goals are, in order of priority:

- to simplify and standardize all systemic aspects of service development, particularly those which relate to the fulfillment of gCube-specific requirements;
- to promote the adoption of best practices in distributed programming for concerns such as safety of concurrent access and autonomicity of behavior.

In particular, the framework:

- manages the entire lifecycle of gCube services, engaging in autonomic interactions with the infrastructure and the local environment, and allowing customisations on state transitions, including deployment, initialization, activation, and failure;
- enforces the scoping and security rules associated with shareable resources, handling the acquisition and renewal of service credentials, the delegation of caller

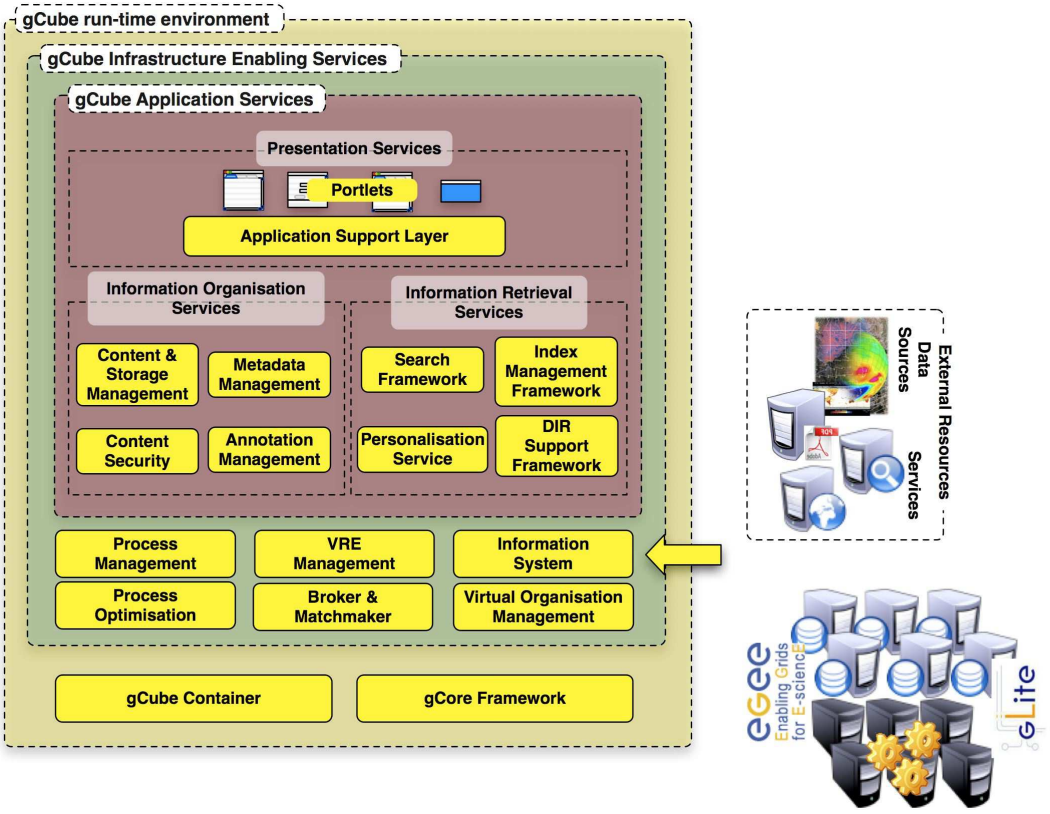

Figure 1. gCube Reference Architecture

credentials, and the propagation of scope and credentials from incoming to outgoing service calls;

- implements WSRF standards for publication, access, and notification of change to service state, offering a rich set of abstractions for modelling it, governing transparently its full lifetime, and managing its persistence on different storage media, including its recovery from remote media upon service migrations;
- standardises the use of systemic faults within service interfaces and implementations, transparently supporting retry-same and retry-equivalent semantics and converting faults into equivalent lighter-weight exceptions at service boundaries;
- mediates access to configuration resources on classpath and local file system, redirecting read failures to backups created prior to write operations and exposing object bindings for all aspects of service and container configuration;
- simplifies resource discovery via object bindings, templating, and XPath inspection for a range of queries to the Information services of the infrastructure;
- simplifies multiprogramming via arbitrary combinations of event-based synchronisation, scheduling, parallelisation, and sequencing of local processes; simplifies distributed programming through the customisation of best-effort discovery and interaction strategies with stateless and stateful services.

The gCube Infrastructure Enabling Services

The gCube Infrastructure Enabling Services is the family of subsystems implementing the foundational services that guarantee the operation of the infrastructure. Such functions are organised in four main areas: (*i*) organisation and execution of Virtual Research Environments (VRE Management) by guaranteeing an optimal consumption of the available resources (Broker and Matchmaker); (*ii*) registration of the infrastructure constituents (Information Service); (*iii*) the authentication and authorization policy enforcement enabling the highly controlled sharing of infrastructure constituents (Virtual Organisation Management); (*iv*) definition and orchestration of complex workflows (Process Management) by guaranteeing an optimal consumption of the available resources (Process Optimisation).

The *VRE Management* is the set of services responsible for: (*i*) the definition of VREs and (*ii*) the dynamic deployment of VRE resources across the infrastructure [11]. VREs definitions are declaratively specified through an appropriate and user-friendly user interface in a dedicated language and inform the derivation of an optimal deployment plan. The plan is based on availability, QoS requirements, resource inter-dependencies, and sharing policies, but also on moni-

toring of failures (resources are dynamically redeployed) and load (resources are dynamically replicated). Three distinguished services (Software Repository, Deployer, gCube Hosting Node Manager) support VRE definition and dynamic deployment by, respectively, collecting service implementations, deploying service implementations and their dependencies on gHN, and hosting such service implementations at selected nodes.

The *Broker and Matchmaker* is the service that identifies the set of gHNs where to deploy a set of services. In particular, given a set of packages to be deployed, their requirements versus the environment and/or other services, it identifies the set of gHN to be used as target hosts for the deployment action.

The *Information Service* is the service catering for the publication of descriptive information about resources (Infrastructure constituents) and their operational context, the discovery of resources based on descriptive information, and the real-time monitoring of resources based on subscription and notification mechanisms. Heavily relied upon all the functional layers of gCube, the Information Service is a replicated service in which instances communicate in peer-to-peer fashion to maximize availability, response time, and fault tolerance.

The *Virtual Organisation Management* is the set of services responsible for equipping gCube with a robust and flexible security framework for managing *Virtual Organizations* (VOs) [12]. gCube exploits the VO mechanism to enforce a trusted and controlled environment in each dynamically created VRE. The main features consist in user and group management, authentication support, authorization definition, delegation, and enforcement of the security credential. These services rely on and integrate VOMS [13] and Globus Security Infrastructure (GSI) [14] to provide gCube with a security framework supporting various configurations. The actors of this framework are humans as well as services.

The *Process Management and Process Optimization* is the set of services supporting the definition and execution of processes, i.e., workflows combining gCube services, external services and gLite jobs to deliver new functionalities (also known as programming in the large [15]). In particular, these services provide the basic functionality for (*i*) creating processes either via a graphical process modelling tool or via a BPEL definition, (*ii*) reliably executing processes in a fully distributed and decentralized, thus highly scalable, way and (*iii*) optimizing processes both at build-time and at runtime. Process execution facility has been designed and implemented to take full advantages of the Grid, i.e., process steps are outsourced to the resources forming the infrastructure and the process is executed in a distributed peer-to-peer modality. In particular, the Process Management service integrates the gLite [8] software, thus enabling gCube to run such processes on EGEE resources. A monitoring front-end allows to get information on individual process instances which are not materialized on a single host because of their distributed execution. This monitoring front-end allows administrators to follow the state of execution of a process instance online and also shows where the different parts are being executed.

1.2 The gCube Application Services

The gCube Application Services is the family of subsystems delivering three outstanding functions of any Virtual Research Environment: (*i*) storage, description and annotation of information in a VRE (gCube Information Organisation Services), (*ii*) retrieval of information in the context of a VRE (gCube Information Retrieval Services), and (*iii*) provision of VRE users with an interface for accessing such an information and the rest of functions equipping a VRE (gCube Presentation Services).

The gCube Information Organisation Services

The gCube Information Organisation Services is the family of subsystems implementing the foundational services guaranteeing the management (storage, organisation, description and annotation) of information by implementing the notion of Information Objects, i.e., logical unit of information potentially consisting of and linked to other Information Objects as to form compound objects. Such functions are organised in three main areas: (*i*) the storage and organisation of Information Objects and their constituents (Content and Storage Management); (*ii*) the management of the metadata objects equipping each Information Object (Metadata Management); and (*iii*) the management of the annotations objects potentially enriching each Information Objects (Annotation Management).

The *Content & Storage Management* is the set of services offering transparent access to Information Objects managed through gCube. In particular, they provide basic functionality for: (*i*) manipulating Information Objects and/or collections, i.e., creating, accessing, storing, and removing; (*ii*) orchestrating distributed storage nodes and providing a transparent access to them; (*iii*) a notification mechanism to maintain derived data upon changes in content; and (*iv*) importing Information Objects from different content providers through wrappers. The kind of Information Object manageable by the Content & Storage Management services is generic and flexible enough to model and thus support several content types. To make full exploitation of Grid storage facilities, the Storage Management service provides an abstract interface to the underlying distributed and heterogeneous actual storage interfaces and technologies (e.g., DPM via SRM [16] and the GFAL interface [8]). Thanks to the gCube replication management subsystem and by integrating gLite, the gCube Storage Management service is capable to exploit the storage capacity of the EGEE infrastructure and maintain multiple copies of the Information Objects as to maximise availability.

The *Metadata Management* is the set of services offering functionality for managing metadata objects, i.e., additional data attached to Information Objects. In particular, these services support (*i*) the manipulation of metadata objects and

metadata collections, i.e., creating, accessing, storing, and removing metadata objects compliant to one or more metadata format, (*ii*) the definition of metadata formats, (*iii*) the transformation of metadata objects into diverse formats via user-defined transformation programs, and (*iv*) the search for metadata objects. These characteristics make the services capable to manage multiple formats of metadata. Moreover, the support for diverse metadata formats and the relative transformation programs are an important feature for dealing with heterogeneity issues. To store metadata objects the services exploit the storage facilities provided by the Content & Storage Management and thus guarantee improvement in reliability and access of managed objects.

The *Annotation Management* is the set of services responsible for cross-model, and cross-media back-end management of annotations, a manually authored and subjective specialisation of metadata objects. The services mediate between interactive annotation front-ends and Metadata Management services by: (*i*) enforcing a consistent modelling of annotation relationships between Information Objects, and (*ii*) increasing the simplicity, granularity, and flexibility with which annotations are created, collected, deleted, updated, and interrelated as specific forms of metadata objects.

The gCube Information Retrieval Services

The gCube Information Retrieval Services [17] is the family of components offering Information Retrieval (IR) facilities to the gCube infrastructure, i.e., allowing searching over data and information by a wide range of techniques. The IR family of services can be decomposed in three major categories, which are presented below and are entitled as "frameworks" due to the fact that they are not standalone services. Instead, they are rather large collaborating systems based on protocols, specifications and software, which expose remarkable extensibility to the gCube system they empower.

The *Search Framework* is the set of services focused on the search-specific aspects of the gCube platform. More analytically, it consists of the search orchestrator component, search operators, query processor components and the data transfer mechanism. The workflow required for computing a user-query is the following. The search orchestrator receives queries from the gCube portal, communicates with the gCore IS service for retrieving information on the operational context. In the next step, the orchestrator feeds this information along with the query to the query processor components which ultimately produce an execution plan. This plan is forwarded to the gCube execution engine (one of which is the Process Management Service) which orchestrates the execution by invoking the search operators, as dictated by the plan. The data transfer is performed by the ResultSet component of the Search Framework. The final results are then forwarded to the user (portal). The Search Operators cover most of the traditional relational algebra operations, as well as some advanced ones, such as geospatial search and similarity search, thus providing a full fledged set of capabilities to the final user.

The *Index Management Framework* is the set of services involved in the creation and management of gCube indices. Management refers to all aspects of an index lifecycle as well as to provide support for search capabilities. In gCube a rich set of indices including text, forward, feature, and geospatial indices, is employed, thus offering a comprehensive set of search capabilities regarding various data types and models. The services of Index Management Framework communicate with the Content & Storage Management services in order to acquire the data set to be indexed and also to preserve the state of indices. They also employ the gCore IS capabilities so as to publish themselves and therefore cater for their exploitation by any client.

The *Distributed Information Retrieval Support Framework* is the set of services which enhance and support the IR system. This framework provides higher-level IR capabilities which include content ranking, source selection and result set fusion (ranked merging of various data sets). Components of this framework communicate with the Index Management Services for statistic extraction and the IS service for information publication. Search Framework employs the advanced capabilities offered by DIR framework in order to enhance its search capabilities, by refining queries, enhancing produced search results and finally exhibiting a higher level of services.

The *Personalisation Service* is an additional component which does not belong to any of the frameworks mentioned above, but acts independently and improves the search quality. It is indirectly invoked by the Search Framework, through an appropriate wrapper, and used for enhancing user queries, by injecting additional "personalized" information.

The gCube Presentation Services

The gCube Presentation services form the logical top layer of a gCube-based infrastructure. Their objective is twofold:

- To provide the means to build user interfaces for interacting with and exploiting the gCube system and infrastructure.
- To provide a full range of user interfaces for achieving interaction with the system, out-of-the-box.

The gCube presentation layer is based on the Application Support Layer (ASL), which is a framework that abstracts the complexity of the underlying infrastructure so that the front-end developer focuses on the objectives of presentation rather the details of the protocols and rules for interacting with the underlying (WSRF) services. The ASL exposes to the developer well known tools as session and credential management and is accessible through various interfaces (currently HTTP and JAVA-native). On top of the ASL the developer can develop the user interface components needed for a particular application, depending on the execution environment that will host them (e.g., php web server, desktop application, application server, etc).

The execution environment is normally provided by existing systems and can be powered by bare Operating Systems / Virtual Machines (e.g., desktop applications), plain html pages, dynamic web-sites (php, asp, jsp, etc), portals, application servers etc.

gCube presentation layer, offers an initial set of components currently running under the JRS168 specification, hosted by GridSphere portlet container [18], while, apart from gCube core services, it is based on Java and servlets technologies for offering it services. These components, which are called "portlets", cover end-user and administrative functionalities. In particular, it includes portlets for search and browse tasks, user profile editing, transforming metadata, managing annotations, viewing and editing metadata, defining and monitoring processes.

2. Availability

gCube is an open source software system. It is expected that the institutions not directly involved in its development are willing to exploit this technology to set up their own infrastructures as well as that software developers are willing to contribute to it with by integrating their own facilities. This section contains information supporting such tasks.

2.1 Software Location

The gCube software is made available via its website [1]. The whole system consists of a number of artefacts including a gHN distribution, i.e., a packaging of selected components forming the gCube runtime environment 1, and a several components realising a part of a subsystem previously described. For each component, it is made available a software archive (a self contained packaging of the component), the source code, and the accompanying documentation including the JavaDoc and the guide documenting it. In order to have access to the software artefacts the user is requested to be registered on the project web site.

2.2 Documentation

gCube is an evolving system, so is the documentation accompanying it. Besides, the technical documentation accompanying the software (namely the Javadoc) it was decided to create a number of guides, each focusing on a typology of actor, and to rely on a Wiki to make this material available.

The *Administrator's Guide* [19] is specifically designed for providing system administrators with instructions for installing and configuring a gCube-based infrastructure. These instructions include the how-to install the gCube components and well as their external dependencies, known issues, incompatibilities, etc. Moreover, this guide provides the reader with guidelines and instructions for the daily administration and upgrading of a gCube-based infrastructure and the existing VREs.

The *User's Guide* [20] is specifically designed for providing end user with information on how-to exploit the facilities of a Virtual Research Environment. In particular, it focuses on the VRE portal. The guide is conceptually comprised of two parts: (*i*) a part that describes common functionality offered by any portal, and (*ii*) an application specific part describing the portals implemented for serving specific communities of practice, namely the Environmental Monitoring and the Fishery and Aquaculture Resources Management (cf. Sec. 5).

The *Developer's Guide* [21] is specifically designed for providing software designers and developers with technical information about the system architecture, the design principle governing the development of the system as a whole as well as those guiding the development of the single constituents, either services and software libraries. This kind of information is suited for: (*i*) those who want to implement new services be relying on the gCube available services and facilities – e.g., programmers willing to interact with a gCube library/service to build their own tools; (*ii*) those who want to modify/extend existing services by intervening in the source code – e.g., programmers willing to fix a bug, developers willing to adapt a service to a specific application domain.

2.3 License

The gCube software is released under the European Union Public Licence (EUPL v.1.1) license [22]. This is the first European Free/Open Source Software (F/OSS) licence. It has been created on the initiative of the European Commission. It is also the first Free/Open Source Licence with the same official value in many European languages. Because of these characteristics, it nicely fits with the gCube settings.

3. Metrics

As discussed, gCube has been designed by following the principles of component-based software engineering, thus it consists of various software components interacting each other to implement the expected functionality. In particular, it has been implemented according to the Service Oriented Architecture principles and related standards while the presentation-oriented components are based on portlets. It currently consists of:

- 54 Web Services;
- 14 Software Libraries;
- 34 Portlets.

4. List of Contributors

gCube is the result of the collaborative effort, led by the Institute of Information Science and Technologies ISTI-CNR (IT), of more than 48 researchers and developers in twelve different academic and industrial research centres. Besides ISTI-CNR, the following institutions have contributed to the gCube system development: University of Athens (GR), University of Basel (CH), Engineering Ingegneria Informatica SpA (IT), University of Strathclyde (UK), FAST Search &

Transfer (NO), CERN – European Organisation for Nuclear Research (CH), 4D SOFT Software Development Ltd (HU), European Space Agency ESA (FR), Scuola Normale Superiore (IT), RAI Radio Televisione Italiana (IT), and European Research Consortium for Informatics and Mathematics (ERCIM).

5. Use and Reuse potential

The gCube technology as a whole can be used by any Institution or initiative that is planning to set up and operate an Hybrid Data Infrastructure capable to promote the development of a number of Virtual Research Environments. Besides the entire software systems, the single constituents can be reused in diverse application scenario by tacking care of properly configuring them with respect to the operational environment.

The gCube technology is exploited to set up and operate the D4Science infrastructure. At the time of writing this report such an infrastructure consist of approximately 75 gCube nodes, 80 gLite Worker nodes and 850 TB disk space spread across 4 sites in Italy, Greece and Switzerland besides the resources obtained via the EGEE infrastructure. The D4Science infrastructure is operated to serve the needs of two communities of practice affiliated with the *environmental monitoring* and the *fisheries resources management*. Some of the institutions partaking in these communities of practice, thus benefitting from the gCube technology deployed in D4Science, are the European Space Agency (ESA), the Food and Agriculture Organization (FAO) of the United Nations, and the International Center for Living Aquatic Resources of the Consultative Group on International Agriculture Research (CGIAR). These institutions analysed the gCube technology in real-life application scenarios including: (*a*) the *Implementation of Environmental Conventions (ImpECt)* which supports the generation of environmental reports with the possibility of periodically maintaining and publishing them (through ad-hoc on-demand processing). It involves the ESA User Services Ground Segment Operation department and associated users (e.g., REMPEC, EMSA, ICRAM). The envisioned high-level services on quality geospatial information operating on top of a powerful and reliable Grid infrastructure involve and bring benefits to a large scientific community, beyond the ImpECt team; (*b*) the *Fishery Country Profiles Production System (FCPPS)* which supports the generation of fishery profile reports, containing focused global information on the state of fisheries in a country-specific format to enhance decision-making and promote advocacy in fisheries, in particular, in the sustainable use and conservation of fish stocks. The production of the country profiles requires complex aggregation and editing of continuously evolving data from various heterogeneous sources. The availability of the dedicated VRE permits updating and web-publishing these vital reports as frequently as the community requires, which is currently impossible. (*c*) the *Integrated Capture Information System (ICIS)* which helps FAO to respond to UN Recommendations to revise its global fisheries statistics so to distinguish catches within and outside Exclusive Economic Zones (EEZs), and to Coordinating Working Party (CWP) agreements to develop an integrated regional and global catch statistics database. The VRE provides the facilities that facilitate the integration and harmonization of regional and global capture and species spatial distribution data including time series and GIS information, and the implementation of peer review mechanisms with the view to improve sources' content. These comparisons concern different sources of capture time series (e.g., regional and global), different sources of species spatial distribution, as well as geographic localization of catch statistics with species distribution. This integrated environment is exploited to provide data owner communities and end users with powerful processing on-line facilities for data extraction and visualization of annual catch data by species and countries.

Acknowledgments

The work reported has been partially supported by the DILIGENT project (FP6 of the European Commission, FP6-2003-IST-2, Contract No. 004260) and the D4Science project (FP7 of the European Commission, INFRA-2007-1.2.2, Contract No. 212488).

References

[1] gCube Development Team. gCube Website. `https://www.gcube-system.org`, 2008.

[2] I. Foster and C. Kesselman. *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan-Kaufmann, 2004.

[3] L. Candela, D. Castelli, N. Ferro, Y. Ioannidis, G. Koutrika, C. Meghini, P. Pagano, S. Ross, D. Soergel, M. Agosti, M. Dobreva, V. Katifori, and H. Schuldt. *The DELOS Digital Library Reference Model - Foundations for Digital Libraries*. DELOS: a Network of Excellence on Digital Libraries, February 2008. ISSN 1818-8044 ISBN 2-912335-37-X.

[4] M. N. Huhns and M. P. Singh. Service-oriented computing: key concepts and principles. *IEEE Internet Computing*, 9:75–81, 2005.

[5] E. Wenger. *Communities of Practice: Learning, Meaning and Identity*. Cambridge University Press, 1998.

[6] L. Candela. *Virtual Digital Libraries*. PhD thesis, Information Engineering Department, University of Pisa, 2006.

[7] G. Wills, L. Gilbert, Y. W. Sim, C. Wang, and L. Carr. Virtual research environments: A literature review. Technical Report ECSTR-IAM05-001, University of Southampton, 2005.

[8] 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*, 12(1):33–45, 2006.

[9] S. Boutsis, L. Candela, P. Di Marcello, P. Fabriani, L. Frosini, G. Kakaletris, P. Koltsida, L. Lelii, D. Milano, P. Pagano, P. Polydoras, P. Ranaldi, F. Simeoni, M. Simi, V. Tsagkalidou, and A. Turli. D4Science System High-level Design. Deliverable DJRA1.1a, D4Science, September 2008.

[10] T. Banks. Web Services Resource Framework (WSRF) - Primer. Committee draft 01, OASIS, December 2005. `http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-01.pdf`.

[11] 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 B. Christensen-Dalsgaard, D. Castelli, B. Ammitzbøll Jurik, and J. Lippincott, editors, *12th European Conference on Research and Advanced Technology for Digital Libraries, ECDL 2008, Aarhus, Denmark, September 14-19*, volume 5173 of *Lecture Notes in Computer Science*, pages 122–134. Springer, 2008.

[12] I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organization. *The International Journal of High Performance Computing Applications*, 15(3):200–222, 2001.

[13] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, Á. Frohner, K. Lõrentey, and F. Spataro. From gridmap-file to VOMS: managing authorization in a Grid environment. *Future Generation Computer Systems*, 21(4):549–558, 2005.

[14] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke. Security for grid services. In *High Performance Distributed Computing, 2003. Proceedings. 12th IEEE International Symposium on*, pages 48–57. IEEE, 2003.

[15] G. Wiederhold, P. Wegner, and S. Ceri. Toward Megaprogramming. *Communication of the ACM*, 38(11):89–99, November 1992.

[16] A. Shoshani, A. Sim, and J. Gu. Storage resource managers. In J. Nabrzyski, J. M. Schopf, and J. Węglarz, editors, *Grid Resource Management*, volume 64 of *International Series in Operations Research & Management Science*, pages 321–340. Springer US, 2004.

[17] 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 L. Kovács, N. Fuhr, and C. Meghini, editors, *Research and Advanced Technology for Digital Libraries, 11th European Conference, ECDL 2007, Budapest, Hungary, September 16-21, 2007, Proceedings*, volume 4675 of *Lecture Notes in Computer Science*, pages 161–173. Springer-Verlag, 2007.

[18] J. Novotny, M. Russell, and O. Wehrens. Gridsphere: a portal framework for building collaborations. *Concurrency and Computation: Practice and Experience*, 16(5):503–513, 2004.

[19] gCube Development Team. Administrator's Guide. `https://technical.wiki.d4science.research-infrastructures.eu/documentation/index.php/Administrator%27s_Guide`, 2008.

[20] gCube Development Team. User's Guide. `https://technical.wiki.d4science.research-infrastructures.eu/documentation/index.php/User%27s_Guide`, 2008.

[21] gCube Development Team. Developer's Guide. `https://technical.wiki.d4science.research-infrastructures.eu/documentation/index.php/Developer%27s_Guide`, 2008.

[22] European Commission. European Union Public Licence (EUPL v.1.1). `http://joinup.ec.europa.eu/software/page/eupl/licence-eupl`.

[23] L. Candela, D. Castelli, and P. Pagano. gCube: A Service-Oriented Application Framework on the Grid. *ERCIM News*, (72):48–49, January 2008.

[24] L. Candela, F. Akal, H. Avancini, D. Castelli, L. Fusco, V. Guidetti, C. Langguth, A. Manzi, P. Pagano, H. Schuldt, M. Simi, M. Springmann, and L. Voicu. DILIGENT: integrating Digital Library and Grid Technologies for a new Earth Observation Research Infrastructure. *International Journal on Digital Libraries*, 7(1-2):59–80, October 2007.

[25] D. Castelli, L. Candela, P. Pagano, and M. Simi. DILIGENT: A DL Infrastructure for Supporting Joint Research. In IEEE Computer Society, editor, *2nd IEEE-CS International Symposium Global Data Interoperability - Challenges and Technologies*, pages 56–69, 2005.

[26] L. Candela, D. Castelli, and P. Pagano. A Reference Architecture for Digital Library Systems: Principles and Applications. In C. Thanos, F. Borri, and L. Candela, editors, *Digital Libraries: Research and Development, First International DELOS Conference, Pisa, Italy, February 13-14, 2007, Revised Selected Papers*, volume 4877 of *Lecture Notes in Computer Science*, pages 22–35. Springer, 2007.