Application-level Research e-Infrastructures: the gCube Approach

Fabio Simeoni†, Donatella Castelli‡, Pasquale Pagano‡, Manuele Simi‡ and Richard Connor†

† University of Strathclyde, Glasgow ‡ Centro Nazionale delle Ricerche, Pisa

For nearly four years now, we have considered e-Infrastructures for research collaborations among parties which are widely dispersed and otherwise autonomous. The collaborations we have in mind are cross-discipline and rely upon data and processing intensive workflows, such as those required for the management of large-scale data sources or the synthesis of digital artefacts from cross-media sources. Equally, they may depend on high-level interactions between end-users and content which are found in conventional digital libraries, from federated searches across heterogeneous sources to metadata and annotation authoring.

Working with prototypical examples in areas as different as Cultural Heritage, Environmental Monitoring and Fishery and Aquaculture Resource Management, we have identified functional and non-functional requirements - particularly security, cost-effectiveness, and autonomic adaptation to highly dynamic environments - which defeat the scope of commodity technologies and ad-hoc developments alike. We thus turn to e-Infrastructures and the Grid vision which underlines them for the provision of the human, hardware and application resources which are required to support true *Virtual Research Environments* (VREs).

As we broaden the scope of resource sharing well into application domains, we align with holistic interpretations of the original Grid vision: like computing cycles, storage, and even data before, we now expect system, domain, and application logic to become a commodity within an infrastructure which abstracts over its physical location at any point in time. This is a hard requirement and has strong implications for the design of infrastructures and the systems which control them:

(i) in line with historical evolution, we expect application-level infrastructures to be layered upon lower-level ones, and associate layers with extensions in the scope of sharing. In our work, we reach the application domain by building upon the GEANT infrastructure for shared connections and the EGEE infrastructure for shared hardware;

(ii) we also expect equally high-level control systems based on service-orientation, rich in content-oriented functionality, and open to third-party services as runtime and development platforms. In our work, we implement SOAs with second-generation Web Service standards (e.g. [8, 2]) and populate them with distributed implementations of conventional digital library services. We also group key services into runtime frameworks, and offer development transparencies through client libraries and a dedicated service container and application framework built upon Globus technology [7];

(iii) most noticeably, we expect control systems to: *a)* support the interactive definition of VREs, including the selection, configuration, and orchestration of the services provided by and shared through the infrastructure; *b)* deploy such services strategically across the available hardware; *c)* monitor their lifetime and redeploy them on demand according to QoS requirements and in response to failures. As the control system itself is the foremost example of shared software, the base requirement is for a state-of-the-art autonomic system. The main output of our work is precisely one such system, *gCube*.

gCube comprises about 140 components functionally distributed across three layers (cf. Fig.1). The top-layer offers a range of presentation services (portlets); these may be dynamically composed into portals through which VRE end-users and administrators can interact with services in the lower layers. The middle layer groups services for information management, primarily a stack of content-management services and a runtime framework of search-management services. The first is rooted in a unifying information model of binary relationships laid upon storage replication and distribution services, either natively developed or built upon those of lower-level infrastructures [11]; services higher up in the stack specialise the semantics of relationships in order to collate, describe, annotate, and transform cross-media content. The search framework processes and optimises structured and unstructured queries over a federation of forward, geo-spatial, or inverted indices of dynamically selected content sources. The bottom layer includes the services which confer autonomic behaviour to the whole system. Process management services consume graphically defined workflows of service invocations, distributing the optimisation, monitoring, and execution of individual steps across the infrastructure. Security services enforce authentication and authorisation policies, abstracting over lower-level technologies [12] to renew and delegate credentials of VRE users and services. VRE management services host service implementations and translate interactive VRE definitions into declarative specifications for their deployment and runtime maintenance, most noticeably the activation and dynamic configuration of the workflows which co-ordinate their execution. The brokering and match-making algorithms which govern deployment strategies are based upon static and dynamic information about the available hardware, data, and services. A peer-to-peer network of information services gathers the information and makes it available to all the services within the infrastructure.

Started in 2004, the development of gCube is the ongoing responsibility of an international consortium partly funded by the European Community [5, 4]. Its core functionality has been designed and initially used in testbed infrastructures for the Environmental Monitoring [1] and the Cultural Heritage domains [6, 10]. Over the next two years, the system will be tested and refined as the control system of production-quality infrastructures, again in Environmental Monitoring abut also for scientific communities in Fisheries and Aquaculture Resource Management [9, 3].

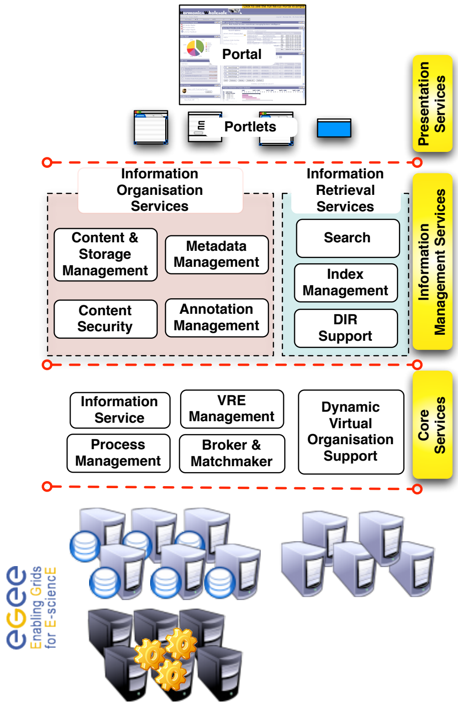

Figure 1: gCube Architecture

References

[1] European Space Agency. Esa protecting the environment. `http://www.esa.int`.

[2] F Bellas. Standards for second-generation portals. *IEEE INTERNET COMPUTING*, Jan 2004.

[3] The World Fish Center. Worldfish. `http://www.worldfishcenter.org`.

[4] The D4Science Consortium. Distributed collaboratories infrastructure on grid enabled technology for science. `http://www.d4science.eu/`.

[5] The Diligent Consortium. Diligent, a digital library infrastructure on grid enabled technology. `http://www.diligentproject.org/`.

[6] Scuola Normale Superiore di Pisa. Center for the data processing of texts and images in the literary tradition (ctl). `http://www.sns.it/it/laboratori/laboratorilettere/testieimmagini/`.

[7] I Foster. Globus toolkit version 4: Software for service-oriented systems. *Journal of Computer Science and Technology*, Jan 2006.

[8] I Foster, K Czajkowski, D Ferguson, and J Frey. Modeling and managing state in distributed systems: the role of ogsi and wsrf. *Proceedings of the IEEE*, Jan 2005.

[9] Fishery Department of the Food and Agriculture Organization Of the United Nations (FAO). Fao fishery. `http://www.fao.org/fishery/`.

[10] RAI. Rai radio televisione italiana s.p.a. `http://www.rai.it/`.

[11] G Stewart, D Cameron, G Cowan, and G McCance. Storage and data management in egee. *Proceedings of the fifth Australasian symposium on ACSW ...*, Jan 2007.

[12] V Welch, F Siebenlist, I Foster, and J Bresnahan. Security for grid services. *High Performance Distributed Computing*, Jan 2003.