Science gateways for biomedical big data analysis

Shahand, S.

Publication date
2015
Document Version
Final published version

About the Author

Shayan Shahand received his bachelor's degree in software engineering from Shahid Beheshti University in Tehran, Iran. He obtained his master's degree in computer engineering with focus on distributed systems from Nanyang Technological University in Singapore. He performed the research presented in this thesis in the e-Science group of the Academic Medical Center of the University of Amsterdam in the Netherlands. He has published and reviewed several articles in various international scientific journals, conferences, and workshops. He is enthusiastic about new technologies and he enjoys programming. He also likes to cook and travel and he is a certified rescue diver.

ISBN 978-94-6182-606-0

Science Gateways for Biomedical Big Data Analysis

Shayan Shahand

Science Gateways for Biomedical Big Data Analysis

Shayan Shahand

Science Gateways for Biomedical Big Data Analysis

Shayan Shahand

Science Gateways for Biomedical Big Data Analysis
Shayan Shahand
PhD Thesis, University of Amsterdam, The Netherlands
ISBN: 978-94-6182-606-0
http://dare.uva.nl/dissertaties

Layout:	Shayan Shahand with the help of F. Maggi, V. Gayevskiy, and the TEX community.
Cover design:	Danial Keshani (Cubex™).
Typeset:	XeLaTeX with packages: *acronym, amssymb, array, babel, biblatex, bidi, booktabs, chapterthumb, (x)color, diagbox, enumitem, eso-pic, etoolbox, fontspec, graphicx, hyperref, koma-script, memoir, microtype, multirow, nth, paralist, rotating, sidecap, subfig, url, verbatimbox, wasysym, xspace, xtab, etc.*
Text fonts:	Gentium Book Basic, Open Sans, Inconsolata, and IranNastaliq.
Printing:	Off Page.

Financial support by the Stichting ter Bevordering van de Klinische Epidemiologie Amsterdam, the Academic Medical Center Amsterdam, ChipSoft B.V., and ABN AMRO Bank N.V. for printing of this thesis is gratefully acknowledged.

Science Gateways for Biomedical Big Data Analysis

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor

aan de Universiteit van Amsterdam

op gezag van de Rector Magnificus

prof. dr. D. C. van den Boom

ten overstaan van een door het College voor Promoties ingestelde commissie,

in het openbaar te verdedigen in de Agnietenkapel

op donderdag 29 oktober 2015, te 10:00 uur

door **Shayan Shahand**

geboren te Isfahan, Iran

Promotiecommissie:

Promotor:	Prof. dr. A. H. C. van Kampen	Universiteit van Amsterdam
Co-promotor:	Dr. S. D. Olabarriaga	Universiteit van Amsterdam
Overige leden:	Prof. dr. J. N. Kok	Universiteit Leiden
	Prof. dr. B. Mons	Universiteit Leiden
	Prof. dr. A. Abu-Hanna	Universiteit van Amsterdam
	Prof. dr. M. T. Bubak	Universiteit van Amsterdam & AGH University of Science and Technology
	Prof. dr. C. B. L. M. Majoie	Universiteit van Amsterdam
	Dr. T. M. A. J. Glatard	Université de Lyon & McGill University
	Dr. ir. M. W. A. Caan	Universiteit van Amsterdam

Faculteit der Geneeskunde

The research described in this thesis was carried out in the department of Clinical Epidemiology, Biostatistics, and Bioinformatics of the Academic Medical Center of the University of Amsterdam and was supported by the Dutch national program COMMIT/ funded by NWO.

To my parents

and Narges & Samin

Contents

List of Figures v

List of Tables vii

1 Introduction 1
- 1.1 Science Gateway Research . . . 2
- 1.2 What Do Science Gateways Offer to Biomedical Scientists? . . . 3
- 1.3 The Science of Science Gateways . . . 5
- 1.4 Thesis Outline . . . 6

I Discovery, Requirements 9

2 Front-ends to Biomedical Data Analysis on Grids 11
- 2.1 Introduction . . . 13
- 2.2 An e-Infrastructure for Bioscience: e-BioInfra . . . 13
- 2.3 Usage of the e-BioInfra . . . 18
- 2.4 Related Work . . . 21
- 2.5 Conclusion and Future Work . . . 23

3 Integrated Support for Neuroscience Research: from Study Design to Publication 25
- 3.1 Introduction . . . 27
- 3.2 Neuroscience Study Phases . . . 27
- 3.3 Proposed Virtual Laboratory . . . 32
- 3.4 Existing Virtual Laboratories . . . 34
- 3.5 Discussion and Conclusion . . . 35

II Design, Development, Operation 39

4 A Grid-enabled Gateway for Biomedical Data Analysis 41
- 4.1 Introduction . . . 43
- 4.2 System Design . . . 44
- 4.3 Related Work . . . 46

4.4 System Architecture 48
4.5 Results 56
4.6 Discussion 60
4.7 Conclusion and Outlook 62

5 A Data-Centric Neuroscience Gateway: Design, Implementation, and Experiences 65
5.1 Introduction 67
5.2 Motivation for a New Gateway 68
5.3 Requirements Analysis 70
5.4 System Design and Implementation 72
5.5 User Feedback 82
5.6 Related Work 84
5.7 Discussion 85
5.8 Conclusions 87

III Insights 91

6 Reflections on Science Gateways Sustainability Through the Business Model Canvas: Case Study of a Neuroscience Gateway 93
6.1 Introduction 95
6.2 Business Model Canvas for Science Gateways 98
6.3 Case Study: AMC Neuroscience Gateway 100
6.4 Discussion and Conclusions 108

7 Science Gateway Canvas: A Business Reference Model for Science Gateways 111
7.1 Introduction 113
7.2 Science Gateway Canvas 114
7.3 Existing SG Frameworks 117
7.4 Related Work 122
7.5 Discussion 123
7.6 Conclusions 124

8 Discussion and Future Research 127
8.1 Requirements of Science Gateways for Biomedical Research 128
8.2 Design, Development, and Operation of Science Gateways for Biomedical Research 130
8.3 Offering Science Gateways as Sustainable Services 131
8.4 Essential Science Gateway Functions 132
8.5 Future Research 133
8.6 Closing Remarks 133

Appendix 137

List of Acronyms 139

Bibliography **145**

Summary **157**

Samenvatting **161**

Acknowledgments **165**

PhD Portfolio **169**

Publications **173**

Contributing Authors **177**

List of Figures

1.1 Overview of publications in the prominent events related to Science Gateways 2
1.2 High-level illustration of Science Gateways. 3

2.1 e-BioInfra layered architecture 14
2.2 Workflow Web Service usage 15
2.3 GridSync usage 15
2.4 The roles of the users categorized by the type of user interface they regularly use 19

3.1 The phases of a neuroscience study 28

4.1 Overview of the e-BioInfra Gateway 45
4.2 e-BioInfra layered architecture 49
4.3 Components involved in the execution of grid applications/workflows . . . 50
4.4 The e-BioInfra Gateway Data Transport component 52
4.5 Directory structure in the Data Transport component and on the grid storage 53
4.6 Simplified database schema of the e-BioInfra Gateway 54
4.7 Simplified e-BioInfra Gateway sequence diagram 55
4.8 Screenshot of the e-BioInfra Gateway GUI for a biomedical researcher . . . 58
4.9 Total execution time of all workflows executed via the gateway 60

5.1 The resources related to the AMC-NSG and their network location 70
5.2 Layered architecture of the AMC-NSG 72
5.3 Simplified entity-relationship model of the AMC-NSG 73
5.4 AMC-NSG submission state diagram 77
5.5 AMC-NSG eBrowser user view for data browsing 78
5.6 AMC-NSG eBrowser user view for processing monitoring 78
5.7 AMC-NSG Administrator processing view 79
5.8 Simplified use case diagram illustrating the interactions between the users and the various components of the AMC-NSG 80
5.9 Summary of the feedback received from the students who used the AMC-NSG in a course 83

6.1 AMC-NSG ecosystem showing most relevant actors and services 101
6.2 The Business Model Canvas applied to the AMC-NSG 103

6.3 Number of jobs submitted by the AMC-NSG users in 2014 for the three available applications 106

7.1 Science Gateway Canvas 114

List of Tables

2.1 Roles taken and types of interfaces used by the e-BioInfra users 20
2.2 Number of workflows submitted by the e-BioInfra users 22

3.1 Summary of the actors and their tasks in the neuroscience study phases . . 30

4.1 Summary of the usage statistics of the e-BioInfra Gateway 59
4.2 Total execution time by e-BioInfra Gateway users per application 61

5.1 Main differences between the previous and the new e-BioInfra Gateway (AMC-NSG) 86

7.1 Qualitative overview of the functions provided by selected Science Gateway technologies 122

8.1 Overview of the biomedical research communities involved in the four Science Gateway generations 127

CHAPTER 1

Introduction

According to the Californian Biomedical Research Association, biomedical research is a broad area of science that carefully investigates biological processes to advance our understanding of (patho)physiologic processes, prevent illness, identify biomarkers, and develop products for treatment and better quality of life [27]. As in other disciplines, biomedical research is also facing the challenges of the data deluge [98, 107, 126]. For examples, see the special issues of Nature Neuroscience [108] and Science [1, 78, 99, 126] that focus on data-related aspects of biomedical research.

Data-driven methods, also known as e-Science or e-Research methods, are defined as a combination of Information Technology (IT) and science that enables scientists to tackle the challenges resulting from the data deluge [10, 65, 67]. Cyberinfrastructures or e-Infrastructures, which are the environments that provide collaborative sharing of distributed and high-capacity computing and data resources, are the IT infrastructures that address these challenges [68]. For example, Buetow [22] introduces two of such e-Infrastructures for biomedical research, namely Biomedical Informatics Research Network (BIRN) [140] and myGrid project [154]. However, e-Infrastructures have the following two drawbacks:

- They fall short of high-level services that genuinely support the needs of scientists [68].
- Scientists find interacting with e-Infrastructures challenging as it requires detailed technical knowledge [56].

Science Gateway (SG) research addresses these drawbacks. SGs are web-based enterprise information systems that provide scientists with customized and easy access to community-specific data collections, computational tools, and collaborative services on e-Infrastructures [29, 172]. The research described in this thesis focuses on understanding fundamentals of SGs for biomedical research to facilitate design, development, and operation of new SGs.

1.1 Science Gateway Research

SGs are also referred to as collaboratories, Virtual Research Environments (VREs), and collaborative e-Research environments [3, 29][1]. Although the definition of collaboratories dates back to 1989 [178], the other two terms are defined in 2004 [4] as community-specific additional services and interfaces to the emerging e-Infrastructures [51]. By 2005, the trend of realizing VREs as web-based systems consolidated in "(Science) Gateways or Portals" as the flagship term, which is exhibited by the first International Workshop on the Gateway Computing Environments (GCE) [166] within the Supercomputing conference. Another important venue for discussing the latest advances on SGs is the International Workshop on Science Gateways (IWSG) series that is running since 2009 [69]. These two workshop series and their related special issues on the Journal of Grid Computing (JGC) [80] and the journal of Concurrency and Computation: Practice and Experience (CCPE) [56, 166, 172, 173] are the prominent channels through which a community of practice reports on their advances, challenges, insights, and solutions. Figure 1.1 summarizes the evolution of publications in these events[2], to show that SGs have been, and still are, a topic of global research.

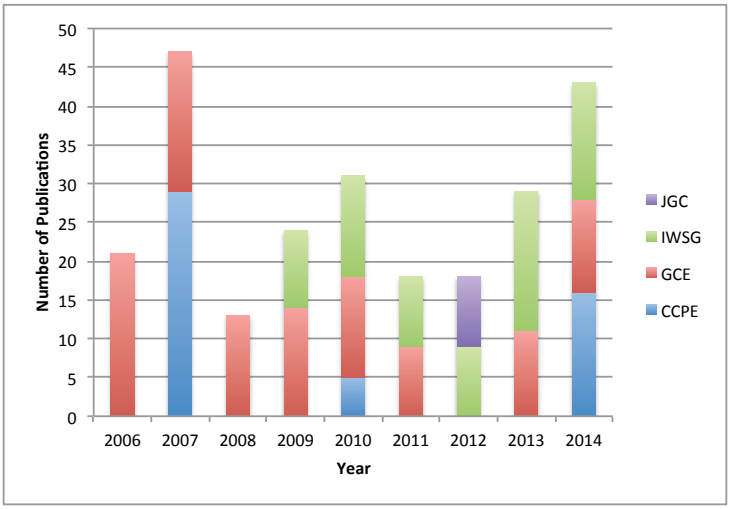


Figure 1.1: Number of publications in the prominent events focused on Science Gateways (2006-2014): International Workshop on Science Gateways (IWSG) and International Workshop on Gateway Computing Environments (GCE). Related special issues appeared in the Journal of Grid Computing (JGC) and the Journal of Concurrency and Computation: Practice and Experience (CCPE).

[1] In this thesis the term "Science Gateway" is adopted, or in some occasions simply "Gateway" is used.

[2] Note that GCE 2005 did not have a separate proceedings, instead its papers appeared in one of the CCPE journal special issues in 2007 [166].

1.2 What Do Science Gateways Offer to Biomedical Scientists?

Modern biomedical research, marked with the advent of high-throughput measurement technologies, faces new challenges resulting from the data deluge. Biomedical scientists need to deal with large volumes of heterogeneous data [98]. For example, in 2013, the European Bioinformatics Institute (EBI) hosted 40 petabytes of data from all domains of life sciences [167]. Moreover, biomedical data is often generated by different centers and in various (non-standardized) formats. Furthermore, understanding biological processes requires integration of many layers of diverse biological data and computational methods. This is complex and computationally demanding [122]. Such a scale and complexity demands advanced computing facilities and multi-disciplinary collaboration between scientists from different organizations [180]. For example, in the Alzheimer's Disease Neuroimaging Initiative (ADNI), researchers from several institutes collect data such as Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET) images, genetics, cognitive tests, CerebroSpinal Fluid (CSF) and blood biomarkers as predictors for the Alzheimer's disease [136]. Another example is The Cancer Genome Atlas (TCGA) project, which represents the effort of scientists from various institutions in the war against cancer using high-throughput genome analysis techniques [144].

In addition to biomedical research challenges, there are also challenges of technical nature. These increasing growth in complexity and pose severe hurdles to most researchers, who lack adequate tools to address their following needs [98, 122]:

- Store, transfer, access, manage, and annotate large-scale, heterogeneous, distributed and complex datasets.
- Analyze data with a diverse set of tools and techniques that are often computationally demanding.
- Share data and analyses in collaboration among multi-disciplinary and (inter)national research groups.

SGs emerge as environments that provide facilities for these requirements. Figure 1.2 presents a high-level view of SGs. At the one end, they integrate data, computation, and collaboration resources, which are provided by e-Infrastructures or operated independently. At the other end, they provide customized, easy-to-use, and integrated services for collaborative data and computation management.

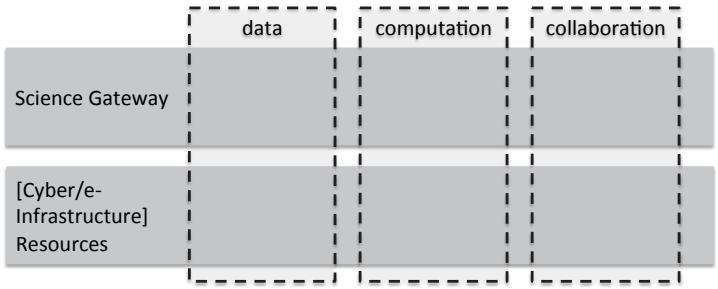


Figure 1.2: High-level illustration of Science Gateways.

From a functional perspective, SGs are understood as systems that have the following properties:

Usability is the "capability of being used"; in the case of SGs it refers to their fitness for use in research by the scientists. SGs have better usability compared to the usual low-level interfaces to e-Infrastructures, especially for domain scientists who do not have advanced IT knowledge. This is achieved by hiding the complexities of the underlying infrastructures from the scientists, and by providing customized and intuitive Graphical User Interfaces (GUIs). SGs allow scientists to focus on their scientific goal rather than getting involved into unnecessary technical details of the infrastructure. They also remove the burden of going through a steep learning curve from the shoulders of scientists, which is necessary if they want to use e-Infrastructures directly. For example, to perform computations on e-Infrastructures without SGs, typically users need to learn how to use Command-Line Interfaces (CLIs) on Unix-like systems, know how to configure them, and become familiar with the related jargon. In contrast, with SGs users initiate computations via web GUIs exposing only high-level configuration of domain-specific tools.

Scalability is the "ability to adapt to increasing demands"; in the case of SGs it refers to their capability to accommodate varying volumes of data and computation. To a large extent, the scalability offered by SGs results from their underlying e-Infrastructures. For example, scientists with access to distributed data and computing resources on e-Infrastructures will have more storage capacity and computational power. Moreover, SGs can implement mechanisms that provide even more scalable solutions, for example, by incorporating multiple e-Infrastructures. Such scalable solutions allow scientists to achieve more in less time or mitigate capacity limitations. For instance, they can perform larger data processing, which can lead to greater statistical power.

Integration is the capability of "combining into a whole" thereby providing added value; for SGs this refers to combination of data, computation, and collaboration resources into a single environment. SGs can provide solutions to integrate heterogeneous data from various data sources with different formats and access protocols, for example by transparent translation between standards and formats. Additionally, SGs can provide mechanisms to integrate processing methods and computing resources to accommodate the often computationally demanding and complex data analyses. Scientists can work collaboratively with their data and processing methods and utilize data and computing resources seamlessly because SGs integrate them all into a single environment.

Automation is the "technique of making a process operate by itself with no human interference"; for SGs this refers to automating and streamlining parts of the research processes that do not necessarily require human intervention. Such automation is required to address the data deluge challenges or to comply with research protocols for reproducible science. Transporting data files, orchestrating and monitoring the flow of data processing steps, and capturing provenance information are examples of

such processes. Automation allows scientists to perform collaborative data analysis and management on e-Infrastructures more easily and efficiently.

Sharing and reuse in the context of SGs refer to giving others access to resources that can be (re)used in research. SGs enable collaboration by facilitating sharing and reuse of data and processing methods for cross-fertilization, both within a same discipline and among different ones. For example, SGs facilitate sharing and reuse by implementing mechanisms to define and manage virtual organizations where the roles of participants define their access to a shared pool of resources, data and processing methods. Moreover, SGs pave the road toward efficient sharing and reuse of data and processing methods by collecting and providing provenance information. This allows scientists to trace back the data and methods history for audit, building trust and giving credit to those who were involved.

There are various examples of SGs for biomedical research that fulfill the needs and have the properties mentioned above. However, since SGs are by definition community-specific, with the consequence that, more often than they are reused, new SGs are developed for specific communities.

1.3 The Science of Science Gateways

The construction of SGs is complex, takes time and needs efforts due to the large number of requirements to be addressed, and the overwhelming amount and diversity of solutions. There have been initiatives to provide some guidance, however they mostly focus at the technical level. The European Grid Infrastructure (EGI) SG Primer [43] and the eXtreme Science and Engineering Discovery Environment (XSEDE) SG cookbook [97], for example, both present technical aspects and best practices to be considered during the SG design and development. In practice, the design of SGs has been approached as isolated technological efforts without a clear methodology, leading to a fragmented landscape that hampers cross-fertilization and further research on SGs. However, the construction of SGs is not merely a technical software engineering problem. It should be approached as a community building process to ensure their efficient development, uptake and sustainability [29].

This research largely advanced our understanding of *the fundamentals of SGs for biomedical research*. Understanding fundamentals of SGs is important because it promotes cross-fertilization among different communities by sharing methodologies, best practices, and even software. Moreover, it facilitates design, development, and operation of new SGs by providing a common structure and terminology to understand, analyze and reason about them. Lastly, it guides future research on SGs by facilitating identification of drawbacks and opportunities. These fundamentals can be derived from understanding the design, development, and operation concerns and the essential functions of SGs that facilitate biomedical big data analysis on e-Infrastructures.

The following questions guided this research to understand the fundamentals of science gateways:

- What are the requirements of biomedical researchers to efficiently use e-Infrastructures?
- How to build SGs that address these requirements? What are the design, development, and operation considerations?
- How to offer these SGs as sustainable services for biomedical researchers?
- What are the essential functions of SGs? How can these guide future research and development on SGs?

To answer these research questions, the following approach was adopted: A number of SGs have been designed, developed, and evaluated iteratively; this resulted in four SG generations. The first and second generations were prototypes and exploratory, and were evaluated with a small set of users or in courses. The third and fourth generations, which are detailed in this thesis (Chapters 4 and 5), were deployed and evaluated based on their operation. This approach follows the design science research methodology [117], which is defined as iteration over the following steps: problem identification and motivation, definition of objectives for a solution, design and development, evaluation and reflection, and communicating methodologies and best practices.

Additionally, the research presented in this thesis is performed in partnership with several biomedical research communities. Because of available collaborations and in-house expertise, the computational neuroscience, omics, and medical chemistry researchers from the Academic Medical Center (AMC) of the University of Amsterdam have been involved during this research. This approach follows the recommendations of the Joint Information Systems Committee[3] report [29] about user-driven and bottom-up approach involving specific research communities throughout design, development, and evaluation of SGs.

1.4 Thesis Outline

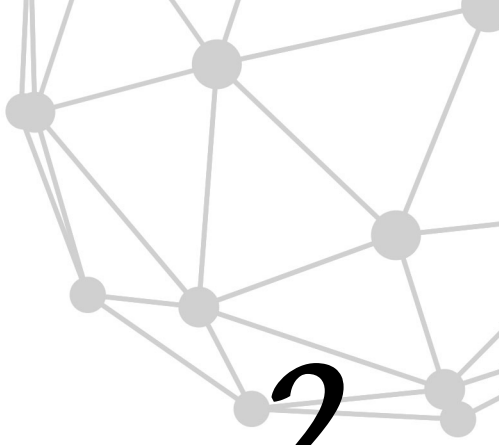
This thesis is divided into three parts. Part I describes the research activities aimed at understanding the requirements of AMC biomedical scientists for analysis of biomedical big data in answer to the first research question. Chapter 2 describes the profiles and roles of the e-Infrastructure users who were involved in biomedical research at the AMC. Chapter 3 describes the functional requirements of computational neuroscience users. Part II refers to the second research question. It presents the design, development, and operation of two AMC SG generations to address the identified requirements. Chapter 4 is about the third generation that was mainly focused on providing processing power to biomedical scientists. Chapter 5 is about the fourth generation that enriches the SG utility by integrating data management. These SGs have been used by AMC scientists

[3] Joint Information Systems Committee (JISC) is a United Kingdom non-departmental public body that supports higher education and research by providing leadership in the use of Information and Communications Technology (ICT) in learning, teaching, research, and administration [152].

daily for their own biomedical research. Part III is about the gained insights. Chapter 6 explores the path to find out how to sustain SG operation, addressing the third research question. Chapter 7 answers the fourth research question. It describes a functional reference model for SGs that can be used to understand their required or essential capabilities, design them, or compare available SGs. Chapter 8 concludes the thesis with a discussion and possible directions for future research.

PART I

Discovery, Requirements



CHAPTER 2

Front-ends to Biomedical Data Analysis on Grids

In Proceedings of the HealthGrid Conference, Bristol, UK, 2011.

Shayan Shahand, Mark Santcroos, Yassene Mohammed,
Vladimir Korkhov, Angela C. M. Luyf,
Antoine H. C. van Kampen, Sílvia D. Olabarriaga

Abstract

The e-Infrastructure for bioscience (e-BioInfra) is a platform integrating various services and middleware to facilitate access to grid resources for biomedical researchers at the Academic Medical Center of the University of Amsterdam. In the past six years the user interfaces with the e-BioInfra have evolved from command-line interfaces to a Java desktop application, and later to an easy-to-use Web application for selected biomedical data analysis. This evolution represents improvements to accommodate the requirements of a broader range of biomedical researchers and applications. In this chapter we present the current user interfaces and analyze their usage considering the typical biomedical data analysis on the e-BioInfra, the roles assumed by the users in the various phases of data analysis life cycle, and the user profiles. We observe that in order to support a wide spectrum of user profiles, with different expertise and requirements, a platform must offer a variety of user interfaces addressed to each user profile.

DOI: 10.6084/m9.figshare.1372471

2.1 Introduction

Biomedical research applications become increasingly compute-intensive and data-intensive, and as such can benefit from Distributed Computing Infrastructures (DCIs), e.g., grid computing. The *e-Bioscience Infrastructure (e-BioInfra)* was introduced by Olabarriaga et al. [112] to support medical image analysis on the Dutch e-Science grid infrastructure. Since 2005 it has evolved into a platform integrating various services and middleware to facilitate access to grid resources. Currently it is adopted by biomedical researchers at the Academic Medical Center (AMC) of the University of Amsterdam (UvA), and a similar approach is also adopted at CREATIS, Lyon, France, for medical imaging research [28].

The platform evolution from a "low hanging fruit" up to a "user-ready" phase is presented in [113]. In the initial phase only Command-Line Interfaces (CLIs) were available. In the user-ready phase a service-oriented approach was adopted to encapsulate the complexity of accessing grid resources into high-level services that could be accessed via the Virtual Resource Browser (VBrowser) which was the single point of access to these services. Workflows are used as basic technology to perform grid computation, enabling applications to be executed on the grid. Our experience shows that even though this technology facilitates data analysis on grids, adopting it remains difficult for biomedical researchers. They do not want to get involved in the grid computing complexities, neither are interested in knowing all technical details. They just want to use applications developed and gridified by others to analyze their data. For these users, a Web interface has been recently added to the e-BioInfra. We realized, however, that all the three types of user interfaces remained in use because users have different preferences, requirements and expertise. In this chapter we analyze the usage of the various interfaces and attempt to characterize the user profiles for each type.

The chapter is organized as follows: In Section 2.2 we present the e-BioInfra architecture and the functionality of its components. In Section 2.3 we present an analysis of the phases in the life cycle of biomedical data analysis, roles, usage patterns and user profiles we observed. We finish the chapter with a brief overview of related work in Section 2.4, and conclusion and future work in Section 2.5.

2.2 An e-Infrastructure for Bioscience: e-BioInfra

In this section we present the latest developments of the improved system architecture as an update to the description in [112]. The main differences concern the Web applications layer and a dedicated solution to transport data to/from grid resources. These were necessary to accommodate the new requirements of a broader range of biomedical research applications and users. We follow a bottom-up approach to describe the building blocks of the e-BioInfra architecture as illustrated in Figure 2.1. Note that the components marked with star are new to the system architecture described in [112].

2.2.1 Grid Fabric and Services Layers

The bottom layer is the grid fabric composed of hardware resources, including Compute Elements (CEs), Worker Nodes (WNs) and Storage Elements (SEs). Resources are pro-

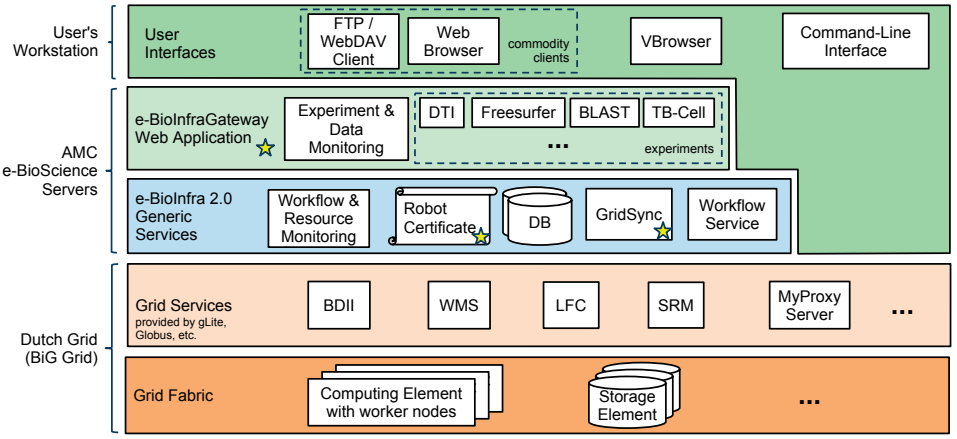


Figure 2.1: e-BioInfra layered architecture: grid fabric and services, generic services, e-BioInfra Gateway Web application and user interfaces. Components marked with star are new in the e-BioInfra.

vided by the Dutch e-Science Grid (BiG Grid project [139]) and currently consist of around 7,000 cores and 60 TB of storage. Access to the resources is granted via membership to the *vlemed* Virtual Organization (VO), using Grid Security Infrastructure (GSI) authentication and authorization. Users in possession of a valid grid certificate are allowed to use these resources.

The middleware components form the grid services layer, for instance, Lightweight Middleware for Grid Computing (gLite) Workload Management System (WMS), LCG File Catalog (LFC), Storage Resource Management (SRM), and Berkeley Database Information Index (BDII). These services are operated by the BiG Grid project.

2.2.2 e-BioInfra Generic Services

The generic services layer offers a higher level of abstraction to interact with the grid services, wrapping them to facilitate usage. The e-BioInfra generic services include:

Workflow Service Web Service that wraps the MOTEUR workflow management engine [59], which is used to enact workflows on the grid resources. Workflows are described in the GWENDIA language [105]. As illustrated in Figure 2.2, MOTEUR uses WMS or the DIANE [36] pilot job framework [106] to execute workflow tasks as jobs on the grid. This Web Service also provides status information about submitted workflows. The service can be invoked directly through various user interfaces, as discussed in Section 2.2.4.

GridSync this service is responsible for transporting data and results between the hospital network and the grid storage as illustrated in Figure 2.3, serving novice users who are not familiar grid protocols nor have a grid certificate. GridSync runs on

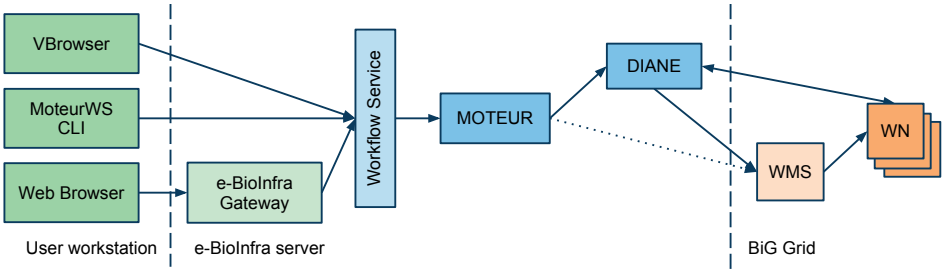


Figure 2.2: Workflow Web Service is used through different user interfaces to submit workflows to the grid using DIANE or the WMS directly (dotted line).

a separate server within the hospital network and it initiates all transfers, for both incoming and outgoing data. All data transported by this service is assumed to be anonymized to satisfy privacy regulations. This service is tightly coupled with the e-BioInfra Gateway, meaning that it is configured to transfer data to and from pre-configured locations on the grid storage resources. Note that the data staging is only a stepping stone for the files and it is not to be used for permanent storage; all files on the data staging server are considered volatile and removed regularly with user awareness. The users can choose a familiar data transfer client, e.g., sFTP or WebDAV client, to manage their data on this server, and the service takes care of the synchronization with the grid storage. The basic principle is that all files put by the user in a particular directory are automatically mirrored to a pre-defined directory on the grid resources. Similarly, all files resulting from grid jobs are stored in a given directory and mirrored back to the data staging server automatically.

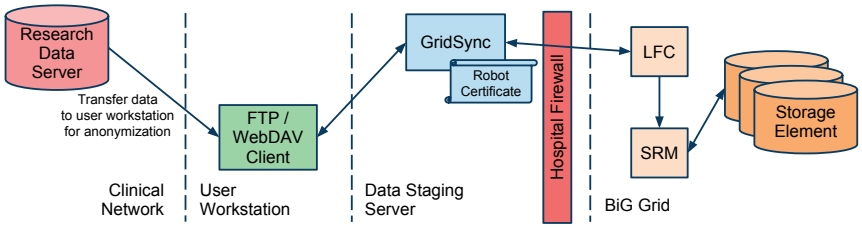


Figure 2.3: GridSync transfers data to/from the grid

Robot Certificate a valid grid certificate is used to generate a grid proxy which is essential for using grid resources. A robot certificate is used periodically to generate a grid proxy automatically on the MyProxy server. This generated grid proxy is then used by the GridSync service to synchronize directories with the grid storage, or by the e-BioInfra Gateway Web application to submit workflows to the workflow service. The robot proxy is only accessible by certain e-BioInfra services (i.e., users cannot use

it directly), to enable authorized users who do not have a grid certificate use the grid infrastructure.

Workflow and Resource Monitoring component is used to monitor and track user activities, for example providing detailed information about the workflows and jobs, DIANE pilot agents; It also provides information about available storage and resource status for provisioning and support purposes.

Databases store structured information about users, experiments, scientific datasets, workflows, etc. It is used by the workflow and resource monitoring component and the e-BioInfra Gateway Web application (see Section 2.2.3).

2.2.3 e-BioInfra Gateway Web Application

The e-BioInfra Gateway Web application [42] is built on top of the e-BioInfra generic services. It is developed using the Spring framework [161], providing a generic substrate into which workflows can be integrated and executed using a Web interface. In addition to submitting the integrated workflows, a.k.a. *experiments*, the e-BioInfra Gateway also generates a predefined directory structure for input data and results as well as identifiers to facilitate experiment management. Each experiment is implemented as a Spring Web Flow making the e-BioInfra Gateway easily extensible.

After successful registration, the user is granted access to the e-BioInfra Gateway, using username and password. It is not necessary for the user to be in possession of a grid certificate because all of the interactions between the Web application and the grid infrastructure are authenticated and authorized by the robot proxy. The information about the experiment and file owners is stored in the database. In addition, all user activities are logged for accountability purposes.

To start a new experiment, first the user is directed to a secured personal FTP directory where he/she can upload the input data; optionally the user can upload the input data using VBrowser directly to the grid storage. The data is pushed automatically to the grid storage by the GridSync service. When ready, the user can start the experiment via the Web interface. After an experiment finishes successfully, the GridSync service fetches the results automatically from the grid storage and stores them in a predefined directory on the server.

Currently there are five experiments integrated into the e-BioInfra Gateway for applications in medical imaging and DeoxyriboNucleic Acid (DNA) sequencing:

- analysis of magnetic resonance Diffusion Tensor Imaging (DTI) in particular the construction of a “brain atlas” for a particular study (see more details in [26]);
- segmentation of Magnetic Resonance Imaging (MRI) data with the Freesurfer toolbox [48];
- DNA sequence alignment with the Basic Local Alignment Search Tool (BLAST) toolbox [91];
- analysis of T/B-cell variation in different organisms [82];

- comparing genomes of related species based on the occurrence of common genes.

The last two experiments have been integrated recently by external developers, showing that the gateway can be extended with other experiments in a straightforward manner.

Note that, whereas there is a clear trend to use frameworks to implement portal interfaces (see Section 2.4), such technology was not our first choice due to the steep learning curve. Instead we chose for a light-weight solution to implement the Web application.

2.2.4 User Interfaces

Users may choose between three types of interfaces to interact with the platform based on their roles, expertise and requirements:

Command-Line Interface (CLI) are the most flexible interfaces to the platform, but they require technical knowledge and experience. They are mostly used by advanced users who like to take control of all of the parameters in the system, those who wish to use them in scripts, or for debugging purposes. Examples are the MOTEUR Web Service client (MoteurWS CLI) and the gLite [88] CLI clients. CLIs require the proper software stack to be installed on the system (e.g., gLite), so they are typically executed in servers with the gLite User Interface (UI) installation. This also requires the user to possess a valid grid certificate.

Virtual Resource Browser or VBrowser [164], is a Java desktop application to access and manage local and remote (grid) resources [111]. It is equipped with a number of plug-ins, e.g., MOTEUR plug-in to submit workflows to the workflow service. This interface is typically used to manage data on the grid, start workflows and view some type of results (e.g., images). The VBrowser is installed on the user's workstation and configured to access the remote resources, and it requires the user to possess a grid certificate to access grid resources. Because of the Graphical User Interface (GUI), the VBrowser is easier to use, however its configuration and operation still requires some knowledge of grid usage and jargon.

Commodity clients are general purpose Web browsers, File Transfer Protocol (FTP) and WebDAV programs can be used to access the e-BioInfra Gateway and GridSync services respectively. These interfaces are publicly available and usually pre-installed on every workstation. The user does not need a grid certificate to use these interfaces, however authentication through username and password is essential. These user interfaces are the easiest for biomedical researchers without grid experience because they shield the grid completely from the user's perspective. However, they offer limited functionality, for example, only pre-defined workflows can be started from a Web interface.

2.3 Usage of the e-BioInfra

The e-BioInfra is being adopted by a growing community of biomedical researchers from medical imaging and genomics to perform data analysis, as well as members of the development and support team. These users have heterogeneous interests and backgrounds, and also display different usage patterns. In this section we present an analysis of the usage following closely the structure suggested in [114], which identified phases of the life cycle of a typical medical image analysis and a set of roles and tasks for each phase. Here we revisit that work in the broader context of data analysis in biomedical research using the e-BioInfra, reformulating the phases in the life cycle and the different roles. Finally we analyze the user profiles taking into account the life cycle, roles and tasks performed by the users.

2.3.1 Life-cycle Phases

First the components of the workflows are developed, either in-house or by third-party collaborators, and then the data analysis workflow is developed; this is called the *development phase*. The parameters of the workflow and its components are then optimized in the *optimization phase* and evaluated to certify that it is working accurately for representative data-sets in the *evaluation phase*.

When workflows are ready, they can be shared with a larger user community. To achieve this, the workflow can be published in some repository (directory on some shared storage such as the LFC) or embedded into some customized user interface (integrated as experiment in the e-BioInfra Gateway). We coin this as *deployment phase*; from this point on the workflow enters the *production phase* (clinical routine phase in [114]) in which it is used routinely for data analysis by users that are not necessarily aware of the technical details.

2.3.2 Roles and Tasks

In our analysis we could roughly identify the same roles and tasks identified in [114]. The component developer builds new data analysis methods that can be combined into workflows by the workflow developer. We observed that the same person also optimizes the parameters of the workflow and its components, as well as performs evaluation. Here we do not distinguish between these roles, referring to them collectively as *workflow developer*.

Moreover we could identify two additional roles. A *service developer* develops generic e-BioInfra services and/or integrates a workflow into the e-BioInfra Gateway, and the *operations support* team is responsible for keeping the platform running and providing user support, for example, communication with the BiG Grid support team. Finally the workflows or the integrated experiments are executed by a *biomedical researcher* to perform data analysis (called clinical user in [114]) .

2.3.3 Usage Patterns and User Profiles

We collected information about the e-BioInfra to identify the roles taken by each user and the interfaces they use most regularly. We also determined whether or not a user has a grid certificate, which is normally an indication of user expertise. From a total number of 40 registered users, seven are inactive and seven are only interested in accessing data in a particular project, so they were not considered here. Table 2.1 summarizes the data collected for the remaining 26 users, and Figure 2.4 summarizes the totals for each role that adopt a given type of user interface. Note that biomedical researchers prefer the Web interface, whereas the developers and support team prefer the CLI and VBrowser interfaces. The VBrowser is widely used by all groups. Although the usage results show that most VBrowser users are also CLI users, this should not be interpreted as disadvantage of any over the other. All of the interfaces are complementary and the users choose an interface to the e-BioInfra based on their preference and requirements. For example, e-BioInfra Gateway is useful for highly-demanded well-established applications, VBrowser is more friendly for occasional usage, and CLIs are better for repetitive tasks.

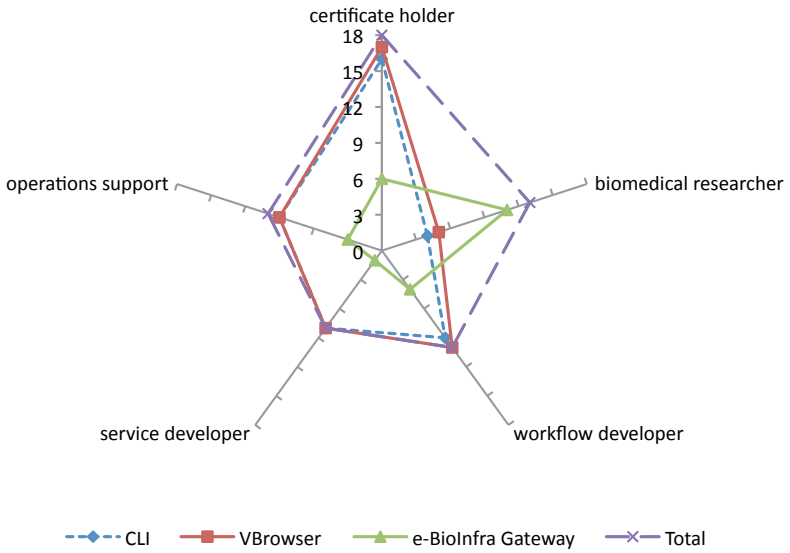


Figure 2.4: The roles of the users categorized by the type of user interface they regularly use.

Although there is no clear distinction between user profiles, we observed five major classes within a wide spectrum of expertise and interest:

(i) users with biomedical background who just want to perform data analysis using existing tools. These are usually not interested in technical details, being

Table 2.1: Roles taken and types of interfaces used by the e-BioInfra users. See text for user profiles.

																					Total	**%**
User	8∗ ☺		☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	26	
Certificate holder			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	18	69
Biomedical researcher	8∗ ✓		✓	✓	✓	✓	✓														13	50
Workflow developer			✓	✓	✓	✓		✓	✓	✓	✓	✓	✓								10	38
Service developer			✓					✓	✓	✓				✓	✓	✓	✓				8	31
Operation support								✓	✓		✓	✓			✓	✓	✓	✓	✓	✓	10	38
CLI user			✓	✓	✓	✓		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	16	62
VBrowser user			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	17	65
e-BioInfra Gateway user	8∗ ✓				✓	✓	✓				✓	✓					✓				14	54
User profile	(i)		(ii)					(iii)						(iv)				(v)				

most familiar with GUIs. They usually take only the biomedical researcher role, preferring a Web interface and avoiding use of grid certificates;

(ii) users with biomedical background who are also interested in technical details and try to improve their data analysis by developing new methods and workflows. They have programming skills and do not mind using CLIs, but they prefer interacting with the system through a GUI. They usually take both the biomedical researcher and workflow developer roles;

(iii) users that only develop workflows (workflow developer role). These usually have technical expertise, but not necessarily a biomedical background. The VBrowser and CLI are the preferred interfaces.

(iv) programmers and software engineers who usually have the informatics background and take the middleware service developer role. CLI in this case is unavoidable for low-level service development and debugging; and

(v) system administrators who are familiar with Unix-based operating systems and comfortable with command-line interfaces. They usually take the support role as external parties (e.g., operators of the BiG Grid facilities).

We also analyzed the activity associated with user profiles and roles. Because of the distributed, loosely coupled and layered architecture of the grid, it is rather difficult or even impossible to collect complete usage information per user. For instance, the workflow service collects the number of workflows per user, but not the CPU usage. Although it is in principle possible to extract this usage from the DIANE pilot job framework, the information is not complete and comparable because not every user uses DIANE. Moreover, some sites publish grid activity only in anonymized way and/or for coarse-grained granularity (i.e., per VO), so the accounting information available for us is not reliable for this purpose. As an indication of activity we therefore only present the number of executed workflows. Table 2.2 summarizes the total number of workflows submitted by the users in the first five months of 2011, when all of the three user interfaces were operational. The numbers are categorized based on two orthogonal qualities: the interface used to submit them (Web and non-Web) and whether the activity refers to "real" data analysis or to development and support. As expected, note that the number of workflows submitted via the Web interface is larger for real data analysis, whereas the CLI and VBrowser are more used for development and support. Also note that the number of workflows for both types of activities are rather similar, although we expect that the workflows executed for data analysis are larger and run longer. Unfortunately we lack data to confirm this hypothesis.

2.4 Related Work

The same basic set-up (MOTEUR Web Service and VBrowser interface) is adopted by GATE-Lab project, which enables running GATE [74] on several computing platforms. It splits the simulation into sub-tasks that are submitted to the grid, monitors the

Table 2.2: Number of workflows submitted by the e-BioInfra users.

	Data analysis	**Development**	**Total**
Submitted via CLI or VBrowser	1450	1350	2800
Submitted via the e-BioInfra Gateway	566	84	650
Total	2016	1434	3450

simulation until completion, retrieves and merges the outputs into a location accessible with a simple Web URL, and keeps track of the simulations history [28].

Web interfaces are common in a large number of projects providing access to grid resources. The European Grid Infrastructure (EGI) User Support Website [44] provides a list of discipline-specific gateways to enable researchers to operate their data analysis in a manner that is more closely aligned to their own particular domain-specific skill sets. For example, the eNMR and WeNMR projects created an e-Infrastructure for the biomolecular Nuclear Magnetic Resonance (NMR) user community. The platform integrates and streamlines the computational approaches necessary for NMR data analysis and structural modeling. Access to the e-Infrastructure is provided through a portal integrating commonly used NMR software and grid technology [19].

Portal frameworks compatible with the Java Portlet Application Programming Interfaces (APIs) (JSR-168 and JSR-286) are becoming popular to build user interfaces to grid-enabled systems. GridSphere is a common portal solution used in grid computing environment [109] that provides a framework compatible with JSR-168. Portlets have a standard API and provide a model for developing specific components for each application; core portlets are available e.g., for authentication. GridSphere is used in many grid community projects, e.g., the MediGRID project [85], the ViroLab project [21]. The P-GRADE portal supports creation, execution, and management of traditional and parameter study grid workflows using a variety of middleware, e.g., gLite and Globus. The first generation of P-GRADE adopts GridSphere as framework [45], but the new generation, coined WS-PGRADE, adopts the Liferay portal framework [153]. P-GRADE is the basis of the simulation platform under development by the SHaring Interoperable Workflows for large-scale scientific simulations on Available DCIs (SHIWA) project [160], which will enable the execution of existing workflows on distributed computing infrastructures using various workflow engines (e.g., ASKALON, Pegasus, P-GRADE, MOTEUR, Triana, and GWES). As part of the SHIWA Simulation Platform, the SHIWA Repository facilitates publishing and sharing workflows, and the SHIWA Portal enables their actual enactment. The Grid Enabled web eNvironment for site Independent User job Submission (GENIUS) is an application independent Web-based portal dedicated to Enabling Grids for E-sciencE (EGEE) gLite infrastructure [13]. It provides general security, job submission and data management services. The portal is implemented using another framework, EnginFrame [147]. The Liferay-based Molecular Simulation Grid (MoSGrid) portal provide grid services for performing molecular simulations on the D-Grid infrastructure, also including services for the annotation of results and data mining [57].

2.5 Conclusion and Future Work

In the past six years, the e-BioInfra evolved to facilitate access to grid resources for biomedical researchers. To support a wide spectrum of user profiles, with different expertise and requirements, the platform now offers a variety of user interfaces. Users choose which interface to use based on the role they take and their preference. We observed that an easy to use Web interface is more popular among biomedical researchers, whereas CLIs and desktop applications are more used for development and support.

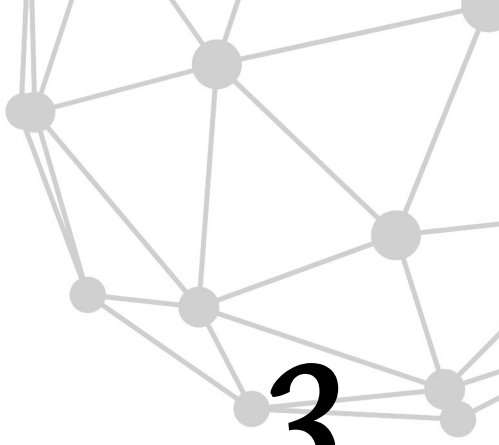
We showed that the platform is used by people within a wide spectrum of expertise and interest, for service development, workflow development and biomedical data analysis. The service oriented architecture of the platform enables it to be flexible and extensible, as we observed its improvement at the same time as it was used in production. In addition we observed that a fruitful e-Science biomedical research community consists of people with a variety of expertise, from informatics background to biology, where some provide operations support such that others can focus on biomedical data analysis.

The e-BioInfra is not complete and there is still much room for improvement. Our vision is to enhance the Web interface to enable collaboration between researchers and facilitate access to grid resources more easily and dynamically. We plan to adopt the Liferay portal framework, and we expect to reuse a major portion of the e-BioInfra Gateway code when migrating to it. Finally, today the user interfaces are disconnected. Finding out about how users choose for one or another will enable us to instrument them to be aware of each other and facilitate navigation between the various types of interfaces.

Acknowledgments

We are grateful to Matthan Caan, Barbera van Schaik, Marcel Willemsen, and Aldo Jongejan for being such faithful users, to Carsten Byrman for his contribution to the development of some services, to the anonymous reviewers of this paper for their invaluable comments, and to all of the users of the e-BioInfra and the e-BioInfra Gateway. We also thank the BiG Grid project and its support team, in particular Jan Just Keijser, as well as the contributors to the following projects: VBrowser, MOTEUR, DIANE, Spring Framework, Glassfish application server, and many other (open-source) projects we benefited from.

This work is financially supported by the AMC-ADICT fund, the BiG Grid programme funded by the Netherlands Organisation for Scientific Research (Dutch: Nederlandse Organisatie voor Wetenschappelijk Onderzoek) (NWO), the SHIWA project co-funded by the European Commission (FP7 contract number 261585), and the COMMIT project “e-Biobanking with imaging for healthcare”.



CHAPTER 3

Integrated Support for Neuroscience Research

from Study Design to Publication

In Studies in Health Technology and Informatics, volume 175, pages 195–204, 2012.

Shayan Shahand, Matthan W. A. Caan,
Antoine H. C. van Kampen, Sílvia D. Olabarriaga

Abstract

Computational neuroscience is a new field of research in which neurodegenerative diseases are studied with the aid of new imaging techniques and computation facilities. Researchers with different expertise collaborate in these studies. A study requires scalable computational and storage capacity and information management facilities to succeed. Many virtual laboratories are proposed and developed to facilitate these studies, however most of them cover only the parts related to the computational data processing. In this chapter we describe and analyze the phases of the computational neuroscience studies including the actors, the tasks they perform, and the characteristics of each phase. Based on these we identify the required properties and functionalities of a virtual laboratory that supports the actors and their tasks throughout the complete study.

DOI: 10.3233/978-1-61499-054-3-195

3.1 Introduction

With the advent of imaging techniques such as Magnetic Resonance Imaging (MRI), the living human brain can now be imaged and examined. This opened up a new field of neuroscience research, in which the (mal)functioning of the brain is being studied with the aid of computation facilities. This new field is known as *computational neuroscience*, which for simplicity we refer to it as *neuroscience* in this chapter. Comparative *studies* into neurodegenerative diseases seek to find and interpret pathological processes in the brain. To account for normal variation in brain structure between subjects, a group of patients is compared to a group of healthy controls. The average difference between these groups used as a measure for damage caused by the studied disease.

Several researchers with different expertises should collaborate to succeed in neuroscience studies. These researchers are usually dispersed among several departments or organizations and need to exchange messages and large datasets. Additionally, they require scalable computational and storage capacity, and information management facilities to process and store their datasets. Many systems and environments, also known as virtual laboratories, are proposed and developed to facilitate and support the studies, however most of them cover only the computational data processing tasks, whereas the studies are in fact broader and include more tasks.

As an example of the complexity and diversity involved in such neuroscience research, let us consider the case of a collaborative study performed by the members of several organizations and departments in the Netherlands and Belgium. Amyotrophic Lateral Sclerosis (ALS) is a progressive motor neuron disease, and may be lethal within one year after diagnosis. Degeneration in regions of the brain controlling movement cause increasing disability to walk, move and breath, and eventually result in heart failure. In a study into ALS, brain scans were acquired as well as clinical parameters (e.g., finger tapping speed) in one hospital. The data processing and analysis was then performed in two departments, each focusing on the issues that required the department members expertise. Afterwards results were merged into one joint publication [63].

In this chapter, we argue that an effective virtual laboratory should cover all tasks that are performed in a study, from its very beginning to its end, and even to its reincarnation in follow-up studies. The contribution of this chapter is twofold: Firstly, we describe and analyze the phases of a neuroscience study in Section 3.2. This description serves as a framework for understanding the properties and functionalities of effective virtual laboratories that support and facilitate the neuroscience studies (Section 3.3). We summarize the properties of some of the existing virtual laboratories in Section 3.4, and close the chapter with discussion and conclusion in Section 3.5.

3.2 Neuroscience Study Phases

Neuroscience studies in which several research groups from various departments or organizations collaborate are known as *multi-site studies*. The dispersed research groups involved in these studies perform one or many of the study phases in parallel. Therefore they have to communicate constantly throughout the study especially where they fork and merge tasks. The Alzheimer's Disease Neuroimaging Initiative (ADNI) [136] is such

a joint effort with many sites in the United States of America involved. Data acquisition is carefully synchronized between sites and the acquired data is stored in a central *data store*. Also, external researchers can apply for a query on this data store to answer their research questions.

The phases of a neuroscience study are illustrated in Figure 3.1. Various actors are involved in the phases: project leaders, statisticians, physicists, image processing experts, computer scientists, medical doctors, and laboratory assistants. The actors, the tasks they perform, and the characteristics of each phase (e.g., type and the amount of data being handled, required computational capacity) are described and analyzed below. Table 3.1 provides a summary of the actors and their tasks in these phases.

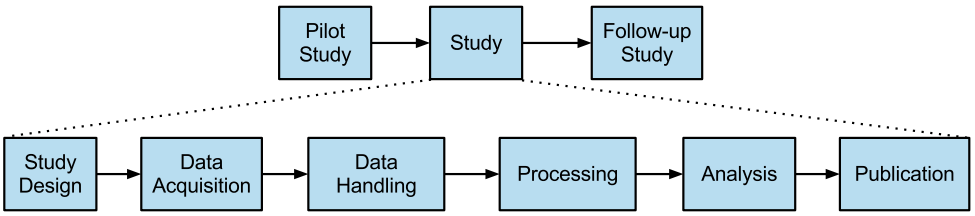


Figure 3.1: The phases of a neuroscience study, which usually starts with a pilot study and continues with follow-up studies.

Some neuroscience studies start with a *pilot study* in which the feasibility of the study is examined by performing it in a smaller scale. This means that a very small population of subjects is processed by going through all of the study phases. The pilot study is also useful to define the hypotheses, methodologies, and goals of the study more accurately and to estimate the required resources for the full study more precisely.

3.2.1 Study Design Phase

The project leaders *define the hypotheses and goals* of the study, for which they perform a (comprehensive) literature review and gap analysis. The statisticians perform a *power analysis* that calculates the minimum population size (sample size) so that the results of the study will be statistically significant.

Defining the *experiment setup* is the next task, in which measurement methods and imaging protocols are specified. The involved physicists who have the expertise in particular measurement devices devise the protocols in collaboration with the project leaders, so that the hypothesized effects are accurately captured.

Logistics provisioning comes next, where the involved physicists, image processing experts, and computer scientists define and estimate the amount of required resources, for example, software licenses, storage capacity, and computation cycles. The compatibility among the resources and experiment setup are considered here.

The data being processed in this phase consists of messages or text documents exchanged among the actors. Note that these documents also include the progress meeting reports and *journal entries* in which the actors explain their findings or actions. The

output of this phase is the project proposal which is sent to the funding organizations after the approval of the ethics committee.

3.2.2 Data Acquisition Phase

The medical doctors ask patients to participate in a study and recruit healthy volunteers. *Recruitment* is done via the clinic or advertisements in the media. Recruited participants are scheduled for experiment session(s) depending on the availability of themselves and of the particular acquisition device(s). Specially for new studies, physicists perform *pilot scans* for a few participants in order to optimize the parameters of the acquisition devices.

The output of this phase is the raw *measurement data* that can be of several types, for example, blood test results, sequencing data, and MRI scans. In the case of multi-site studies the raw measurement data is gathered in different sites (organizations or departments) following the specified protocols. *Measurement data acquisition* is performed by laboratory assistants.

3.2.3 Data Handling Phase

The tasks in this phase are data import, adding metadata, integration, and pseudonymization. Laboratory assistants *import* raw measurement data into data stores and describe it thoroughly by *adding metadata*. The physicists, usually with the help of computer scientists, *integrate* measurement data over different devices to relate information in each particular data type. The data integration may involve reusing of existing measurement data from other studies; likewise this studies' data may be reused in the future. Throughout the study the raw measurement data is processed and the resulting data is also stored in the data stores. The raw data gathered in the data acquisition phase are of different types and usually saved in some well known or custom format required by the processing software. The raw data are stored on different storage devices near the location where they are acquired and should be integrated into a (virtual) unified data stores before they can be processed and analysed.

To ensure the privacy of the study participants, laboratory assistants *pseudonymize* the raw data. This means removing personal information (e.g., name, date of birth) and other information that can be used to identify the participant (e.g., participant's face in head scans). However, the degree of pseudonymization differs from case to case, for example, if the data is going to be processed inside the boundaries of the department, removing the personal information usually suffices. Note that designated actors should still be able to revert the pseudonymization process in special cases, for example, to support the (legal) protocols that follow the accidental findings.

The life expectancy of data should be taken into account to choose the method of storage in order to meet the required reliability and accessibility. For example, expensive MRI scans should be stored in highly protected storage devices for a long time, so that they can be reused in future studies and/or can be referred to for further investigation. In contrast, the intermediate results can be disposed of when the final results are obtained. In addition, neuroscience research needs to comply to complex hospital and legal regulations regarding patient safety, privacy and ethical

considerations. This implies robust storage of data for a period of 15 years, logging of access, modification and transport, and a strict security policy.

Table 3.1: Summary of the actors and their tasks in the neuroscience study phases.

Phases	Tasks	Project leader	Statistician	Physicist	Image processing expert	Computer scientist	Medical doctor	Laboratory assistant
Study Design	define hypotheses and goals	✓						
	power analysis		✓					
	experiment setup	✓		✓				
	logistics provisioning			✓	✓	✓		
Data Acquisition	participant recruitment						✓	
	pilot scan			✓				
	measurement data acquisition							✓
Data Handling	import & pseudonymize data							✓
	adding metadata							✓
	data store integration			✓		✓		
Processing	workflow (component) development				✓	✓		
	workflow optimization				✓	✓		
	workflow evaluation					✓	✓	
	routine data processing				✓	✓		
Analysis	feature extraction				✓			
	statistical analysis		✓		✓			
	interpretation of results	✓	✓	✓	✓		✓	
Publication	publication	✓	✓	✓	✓	✓	✓	

3.2.4 Processing Phase

The pre-processing of measurement data prepares the raw data for the actual processing, for example by correcting for subject motion during MRI scanning. Subsequently,

pre-processed data are processed to extract the relevant descriptive features to be further statistically analyzed in the subsequent phase. For the sake of simplicity, we refer to the pre-processing and processing steps collectively as *processing*. Workflows are used to describe the (complex) processing steps in a high-level structured and reusable way.

The processing of the measurement data is done either by (a) standard methods that are based on accepted processing algorithms routinely used in research; or (b) custom methods that are developed during the course of the study. Olabarriaga et al. [114] provide a comprehensive description of the tasks, actors and expertise involved in processing phases. In summary, four tasks are identified: development, parameter optimization, evaluation and routine research usage. In the *development* task the workflow and sometimes its components are developed by image processing experts and/or computer scientists. The image processing experts *optimize* the workflow parameters by running it on a limited set of input data. Medical doctors *evaluate* the workflow by making sure that the results are plausible and correct. Finally the workflow is deployed in production and used by the image processing experts for data processing in the study. The computer scientists manage and operate the processing phase in collaboration with the other actors. Their technical expertise is required because of the complexity and heterogeneity of the available technologies and infrastructures.

As also described by Olabarriaga et al. [114], each task in this phase has its own characteristics and requirements. In summary, the amount of data being processed grows throughout the tasks from development to routine usage, and so does the required CPU cycles. Faults in the data processing workflows are acceptable during the development task, but they are unacceptable during the evaluation and routine research usage.

If a problem with the outputs is detected after the data processing, the data may be processed again with different parameters, or data may be excluded from the analysis. Accidental findings in a particular participant should be assessed by physicians so that the study adheres to hospital ethical regulations.

The workflows do not always consist of automated steps, but may also include interactive steps in which an expert should perform a task that cannot (yet) be automated. For example, in some of the neuroscience workflows an expert should visually inspect the intermediate results for artifacts or flaws, or (s)he has to manually draw the region of interest on imaging data. These workflows are coined as *man-in-the-loop* workflows.

3.2.5 Analysis Phase

The image processing experts *extract the relevant features* from the data. The results are usually visualized with existing software packages to perform the comparison. For the interpretation of results it is important to know the lineage information, which is how the results are obtained, for example, the parameters used for data acquisition or processing.

A statistician and an image processing expert perform the *statistical analysis* of the data. For that, the appropriate data produced in the processing phase is retrieved and merged. Usually there are software packages to facilitate the statistical analysis and produce final results and figures. Medical doctors and project managers *interpret* the

comparison between control and subjects, and relate them to the hypothesized effects of the studied pathology. In order to do that, they receive support from project leaders, statisticians, physicists, and image processing experts.

3.2.6 Publication Phase

Writing the publication is a collaborative task among most of the actors. It is organized similarly to the initial study design, which requires exchange of messages and documents among all actors who were involved in the study. As part of the publication, a comparison is made between the results of the study and the literature. Reviewers might require further experiments, which could involve collecting additional data and rerunning data processing and analysis. This likely would have to happen within a short time frame.

3.3 Proposed Virtual Laboratory

The effective virtual laboratory should facilitate sharing of data, methodology, and expertise. Security and privacy rules and regulations are of vital importance in data sharing. Sharing enables the researchers to collaborate and reuse the output of others' effort. It also prevents methodologies from decaying over time by a combination of expert and community curation [60].

To handle the computational and storage capacity requirements of the neuroscience studies, the virtual laboratory should be scalable and flexible. Addition of new computational and storage resources should be transparent to the actors. The distributed system paradigm addresses these requirements. To cope with the dynamic nature of distributed systems, the service-oriented architecture, abstraction layers, logging mechanisms, and automatic failure detection and recovery systems should be leveraged. In addition to these features, below we describe other functionalities necessary to support the complete neuroscience study. Note that there are various tools and technologies exist that could be leveraged to realize the virtual laboratory, however discussing them is beyond the scope of this chapter.

Literature Discovery. Scientific publications are published and indexed in several databases. Specific to neuroscience publications, there are databases that provide coordinate-based search functionalities (e.g., BrainMap [141], Brede Database [142]). Searching all of these databases for related literature can be exhausting. Additionally it is not trivial to manually create a comprehensive search clause that covers all combinations of the keywords and their synonyms. Therefore it is important to support the actors with a literature discovery system that helps them to find existing literature related to the study. This functionality is used in the study design, analysis, and publication phases.

Communication and Notification. Several experts should communicate, via messages or documents, at all phases of the study, especially during the study design, analysis, and publication. The communications are of vital importance, especially in

multi-site studies, where the involved researchers need to collaborate and coordinate remotely. A communication system equipped with document versioning and tagging features helps experts to perform the collaborative tasks more smoothly. Additionally, the notifications from different systems acting on the virtual laboratory seek the attention of the responsible person on various events such as workflow status updates, new data, and software failures. An effective communication and notification system routes messages and event notifications to the designated person and stores them for later reference.

Metadata Management. Metadata plays a crucial role to discover and process data especially in multi-site studies, or when reusing a dataset from an old study or from a public data store. Also, metadata is a key factor for discovering knowledge in literature, internal messages and documents, workflows, system events, and measurement data collectively. Additionally, metadata facilitates the reusability of data and methods by describing them thoroughly. The goal of the metadata management system is to help the actors to annotate this collective knowledge in a well-structured and systematic way so that it can be discovered and used more efficiently. It is used in all phases of the study especially in the data handling, processing and analysis phases. An *ontology* is the structural framework for organizing information. It formally represents knowledge as a set of concepts and the relationships between those concepts. A comprehensive ontology is the core of the metadata management system.

Data Management. The measurement data vary in size and type, and are stored in different locations at different phases of a study. The data is transferred from one location to another and leaves/enters the boundaries of organizations constantly. An ideal data management system handles data transfers, data conversions, and applies data privacy regulations automatically, securely, and transparently.

Workflow Management. Scientific workflows are widely used to enable the composition and execution of complex data processing [34]. They aid in implementing medical image analysis methods as a composition of processing steps. Workflows enable collaborative research over shared methods and shared data [60]. A workflow management system executes data processing workflows on computing resources such as clusters or grids. It also provides functions to compose/edit workflows, monitor/manage their execution, and log provenance information. It is used extensively during the processing phase of the studies.

Provenance Information. Provenance information system provides detailed information about the production or delivery of data or methods. It is meant to answer questions on how many resources (e.g., CPU hours) were used, and which methods, processes, parameters, and people were involved to produce a particular data or method. If inconsistencies are observed in the outputs of the same input going through a processing phase at two different points in time, the provenance information system can highlight the differences in the complex pipeline through which the outputs were produced. It also helps to find the affected outputs if at some point in time a defect

is detected in the pipeline or to find the source of a problem in case of an error or a faulty output. It also plays a crucial role in evaluation and validation of the results for audit, reproducibility, and reusability. The provenance information system gathers information during all phases, especially in the data acquisition, data handling, and processing phases. It is then used to provide information to the actors in the processing, analysis, and publication phases.

Visualisation. The actors need to inspect and edit the medical image data at various phases, for example, to remove artifacts, extract features, or annotate a region of interest. Additionally, visualization of statistical data is vital for statistical analysis and interpretation of results. Several software packages exist for two- and three-dimensional visualization and they are used in all phases of the studies. For large studies visualization might require advanced hardware or High Performance Computing (HPC).

3.4 Existing Virtual Laboratories

Olabarriaga et al. [114] summarized a selection of problem solving environments related to the medical image analysis (processing phase). Frisoni et al. [52] also provided an overview of a selection of virtual laboratories for neuroscience research in neurodegenerative diseases. In this section, we summarize the properties and functionalities of some of the virtual laboratories for neuroscience research. Note that this list is not exhaustive.

The Laboratory Of Neuro Imaging (LONI) pipeline environment facilitates the integration of disparate data, tools, and services in complex neuroimaging data processing workflows. It supports neuroscientists with visual tools for data management and integration, and workflow development and execution. It also updates the data provenance automatically during the processing [38].

The Biomedical Informatics Research Network (BIRN) provides a data sharing infrastructure for biomedical research community. It provides capabilities such as data and metadata management, security, and information integration [64].

The CBRAIN platform links Canada's five leading brain imaging centres for data sharing and distributed processing. It provides transparent access to remote resources to manage, share, process, and visualise imaging data [52].

The neuGRID project aims at collecting and archiving of large amounts of imaging data and data processing workflows, and allowing access to computational resources. It provides services for data processing and visualization via a portal. It also includes generic services for workflow management, security and privacy, data and knowledge management, and provenance [120].

In the outGrid project, researchers try to consolidate the three existing infrastructures for computational neuroscience (neuGRID, CBRAIN and LONI) into a worldwide neuroscience infrastructure [72]. Similarly, the goal of the Diagnostic Enhancement of Confidence by an International Distributed Environment (DECIDE) project is to build an e-infrastructure upon neuGRID that offers comprehensive data store of brain image scans and diagnostic tools to identify image markers [71].

The NeuroLOG project aims at integration of heterogeneous data and providing data processing services for neuroscientists. It provides a middleware that includes generic services for data and metadata management, and processing tools and workflows. A semantic data framework is also provided for knowledge and tool discovery [104].

The MediGRID project [85], the Virtual Imaging Platform (VIP) [131], and the Virtual Laboratory for e-Science (VL-e) [112] provide tools and services for data and workflow management that are used by neuroscientists. VL-e has been extended with a Web portal through which neuroscientists can manage and process the medical imaging data [127], and a provenance information system [93].

To our knowledge none of the above virtual laboratories covers all of the neuroscience study phases.

3.5 Discussion and Conclusion

The number, dispersion and heterogeneity of involved researchers, the large volume and size of heterogeneous data, and the computational requirements for data processing make neuroscience studies challenging. Although many virtual laboratories developed to support neuroscience research, most of them cover only the data handling and processing phases of the studies. However, the neuroscience studies are not only about these two phases. We described the properties and functionalities of a virtual laboratory within which all phases of the neuroscience studies can be performed.

The envisioned virtual laboratory enables multidisciplinary and collaborative research between experts in different fields and facilitates sharing of data, methodology, and expertise. Researchers are no longer bound to one location. Management of communications, data, and processes are partly handled by the virtual laboratory, resulting in less overhead and liberating the actors so that they can focus on the content of the studies. For example, organization and versioning of measurement data alleviates the need of a time consuming and error prone manual organization by laboratory assistants.

Transparency in research is facilitated by granting other researchers access to the measurement data and provenance information, which also enables intermediate steps and results to be easily reproducible, adding value to publications. Cross-fertilization between studies by sharing data, which is currently uncommon, reduces measurement costs and increases statistical power. Finally, the virtual laboratory enables performing meta-analyses over multiple studies.

A critical step towards usability is that the actors with minor technical skills must be willing and able to learn to work within the virtual laboratory. Designing and developing a virtual laboratory that supports a research team over an entire study, will lead to significant advances in neuroscience research.

Acknowledgments

We are grateful to the members of the bioinformatics laboratory, the radiology department and the Brain Imaging Center (BIC) of the Academic Medical Center (AMC) of the University of Amsterdam (UvA) for sharing their enlightening insights with us. We

would also like to thank the anonymous reviewers for their feedback on this paper. This work is financially supported by the Academic Medical Center of the University of Amsterdam, and the COMMIT/ project “e-Biobanking with imaging for healthcare”.

PART II

Design, Development, Operation

CHAPTER 4

A Grid-enabled Gateway for Biomedical Data Analysis

In Journal of Grid Computing, 10(4):725–742, 2012.

Shayan Shahand, Mark Santcroos,
Antoine H. C. van Kampen, Sílvia D. Olabarriaga

Abstract

Biomedical researchers can leverage grid computing technology to address their increasing demands for data- and compute-intensive data analysis. However, usage of existing grid infrastructures remains difficult for them. The e-infrastructure for biomedical science (e-BioInfra) is a platform with services that shield middleware complexities, in particular workflow management and monitoring. These services can be invoked from a Web-based interface, called e-BioInfra Gateway, to perform large scale data analysis experiments, such that the biomedical researchers can focus on their own research problems. The gateway was designed to simplify usage both by biomedical researchers and e-BioInfra administrators, and to support straightforward extensions with new data analysis methods. In this chapter we present the architecture and implementation of the gateway, also showing statistics for its usage. We also share lessons learned during the gateway development and operation. The gateway is currently used in several biomedical research projects and in teaching medical students the principles of data analysis.

DOI: 10.1007/s10723-012-9233-4

4.1 Introduction

Biomedical research applications become increasingly compute-intensive and data-intensive [66], and as such can benefit from Distributed Computing Infrastructure (DCI), for example, grids [8]. Several research groups tailored specific purpose infrastructures (e.g., MediGRID [85], GATE [28], Biomedical Informatics Research Network (BIRN) [64]) by providing customized services and components on top of the generic purpose DCIs (e.g., Enabling Grids for E-sciencE (EGEE) / European Grid Infrastructure (EGI) [47], Open Science Grid (OSG) [5]) to meet the requirements of their applications. The *e-Bioscience Infrastructure (e-BioInfra)* is also such a specific purpose infrastructure that was introduced to support medical image analysis on the Dutch grid infrastructure [112]. Since 2005 it has evolved into a platform that integrates various services and middleware to facilitate access to and usage of grid resources [113]. Today the e-BioInfra is used on a daily basis by various researchers at the Academic Medical Center (AMC) of the University of Amsterdam (UvA) to perform large scale data analysis in medical imaging [26] and DeoxyriboNucleic Acid (DNA) sequencing [91] experiments.

The e-BioInfra platform adopts a component-based architecture in which high-level components encapsulate the complexity of the middleware needed to access grid resources. The core is a grid workflow management system to execute complex data analysis pipelines on grid resources with minimal human intervention. The workflow management service can be used from Command-Line Interfaces (CLIs) and a desktop application called the Virtual Resource Browser (VBrowser). Even though the e-BioInfra has lowered the barriers for biomedical researchers to perform large scale data analysis on grids, its adoption is still hampered by the skills required to develop and execute grid workflows. In practice the biomedical researchers still need to take care of various low-level details, such as programming workflow components, transferring data, executing and monitoring workflows, and organizing results produced in the various experiments. Whereas some biomedical researchers are comfortable with such details, others do not want to be involved in the (grid) computing complexities, neither are they interested in the technical implementation; they just want to reuse existing data analysis pipelines that have been developed, optimized and ported as workflows to grids by others. The e-BioInfra Gateway was developed for these users, to further simplify the execution of existing grid workflows. This gateway provides a Web interface, which is more accessible for biomedical researchers and easier to use than CLI and the VBrowser. It also provides means for easy authentication with grid resources, data transport to and from grid storage resources and management of large experiments that require the execution of various workflows or several weeks to complete. Moreover, the gateway is extensible, enabling new data analysis pipelines implemented as workflows to be added with minimal effort. Behind the gateway a group of experts takes care of development and maintenance of workflows for data analysis, operate and further extend the gateway, monitor activities and manage users. After 19 months in daily use, the e-BioInfra Gateway has become the virtual place where experts with various backgrounds collaborate to perform data- and compute-intensive biomedical research on grids.

In this chapter we present the design and functionality of the e-BioInfra Gateway, as well as an analysis of initial results after 19 months of activity. The system design is

presented in Section 4.2. An overview of related work is provided in Section 4.3. It is followed by the system architecture in Section 4.4. An analysis of results obtained with the gateway are presented in Section 4.5, followed by a discussion of lessons learned during the gateway development and operation in Section 4.6. We end the chapter with conclusion and outlook in Section 4.7.

4.2 System Design

To design a Web interface for the e-BioInfra platform, we identified the actors and envisioned their usage scenario, which helped us to identify the system requirements.

4.2.1 Actors

The typical actors who are involved in the biomedical research projects at the e-BioInfra are identified in another study [127]. In summary these actors take the following roles:

- *Workflow developers* compose data analysis pipelines by developing new data analysis methods and combining them with existing methods and/or workflows. They also perform evaluation, validation, and optimization of parameters of workflows and their methods.
- *e-BioInfra developers* develop generic components and/or integrate workflows into the e-BioInfra Gateway.
- *Administrators* operate and maintain the platform and provide user support.
- *Biomedical researchers* execute (existing) workflows to perform data analysis on grid resources.

4.2.2 Usage Scenario

Figure 4.1 presents an overview of the gateway and its utilization. Workflow developers compose and evaluate workflows that implement some data analysis pipelines. These workflows are then integrated into the e-BioInfra Gateway by e-BioInfra developers. Such integrated workflows are further referred to as *applications*.

When biomedical researchers want to run such applications, they sign into the gateway and upload the data to analyze. They choose one application to execute, select input data, define parameters, and then start it. They monitor the execution of the application, which is further referred to as an *experiment*. Upon completion they download results. Researchers interact with the gateway through one (Web-based) interface with no platform dependency and no or minimal software installation and configuration. The gateway also helps them organize their data and experiments through the course of research projects.

Administrators monitor user activity and system events in order to intervene for troubleshooting or user support when required. They also maintain system operation and configure its settings.

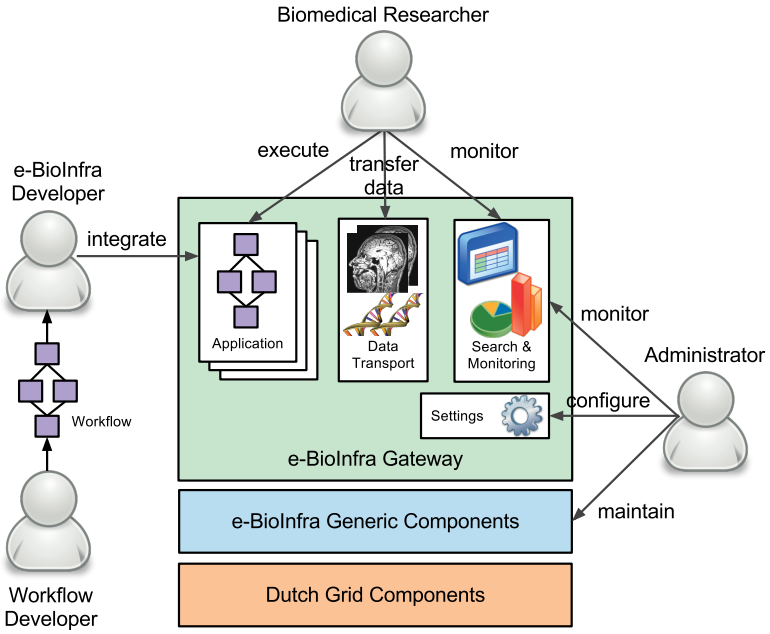


Figure 4.1: Overview of the e-BioInfra Gateway, the underlying e-BioInfra platform and grid resources, and the people involved in its utilization: biomedical researchers, administrators, workflow and e-BioInfra developers.

4.2.3 Requirements

To realize the usage scenario, and to overcome deficiencies observed in previous implementations [25], the following requirements have been identified:

- Cross-platform with no or minimal software installation and configuration on the researchers' machines.
- User authentication via username/password. Grid authentication should be provided invisibly by the gateway.
- Role-based user authorization (e.g., neuroscientist, administrator) to provide customized functionality.
- It should be easy to extend the gateway with new applications, as well as reuse existing code for higher efficiency.
- Efficient and flexible data transfer mechanism between local and grid storage in particular for large and many files. Users should not be bothered by grid protocols and custom grid enabled clients.

- Experiment management. The experiments executed via the gateway should follow best practices for organization of inputs, outputs, and temporary results, for example, in a fixed directory structure.
- Logging and monitoring functions to enable inspection of information related to workflow execution over long periods of time, until the results obtained have been published.
- Administrative functions, such as configuration and monitoring.

4.2.4 Implementation Considerations

When deciding upon the approach to implement the gateway, three alternatives were considered:

- Implementing the gateway from scratch by using software toolkit and libraries. Although this approach gives absolute freedom to design the gateway based on the identified requirements and existing software stack, it requires a lot of effort to implement and provide generic functionalities such as access control and database management, which are usually available through other approaches.
- Implement the gateway using a Web application framework such as Spring [161], Google Web Toolkit (GWT) [150], and Pylons [156]. This can be considered an intermediate approach because it gives freedom of design, whereas providing some generic functionalities such as role-based access control and database management. Compared with the other approaches it needs less investment because of relatively lower complexity in architecture and fewer different technologies.
- Extending an existing gateway (see Section 4.3.1) or portal framework such as Liferay [153], GridSphere [110], and EnginFrame [147]. Existing portals provide many high-level functionalities out of the box and are usually extensible via plug-ins, or in the case of Web portals, via portlets. On the other hand, this approach requires large investment to learn the usually complex architecture and technologies used. It is also sometimes restrictive in terms of design decisions and/or extensibility of existing software stack. Maintenance of such Web interfaces could be difficult because of their typical complexities and software dependencies.

Based on the identified requirements, the considerations above, and the available time and experience of the team, we chose to use a Web application framework. We also decided to follow the component-based approach by separating various functionalities of the gateway into loosely coupled parts. This resulted in independent components that support particular functionalities that can be ported to a portal framework later when time and experience are available.

4.3 Related Work

Several life science communities chose grid technology to realize (collaborative) medical research data analyses, which are compute- and/or data-intensive [20]. A large

number of grid portals have been developed by different research communities around the world to hide the complexity of underlying grid infrastructure behind more abstract and intuitive user interfaces. For example, see [171] for an overview of TeraGrid science gateways, and [56] for an overview of grid portals for life sciences and a comparison of tools and technologies for creating them. The EGI user support Website for "science gateways" provides a list of domain-specific portals to enable researchers to operate their data analysis in a manner that is more closely aligned with their own skill sets [44]. Here we discuss a few examples of general purpose and life sciences grid portals. These portals are built using the approaches explained in Section 4.2.4.

4.3.1 General Purpose Grid Portals

These portals provide basic tools that portal developers can use to interact with grid middleware. Examples are WS-PGRADE and GENIUS.

The Web Service – Parallel Grid Run-time and Application Development Environment (WS-PGRADE) portal [75], the latest version of the P-GRADE grid portal family, is an open source multi-grid portal based on Liferay that supports creation, execution and management of Directed Acyclic Graph (DAG) workflows. It provides high-level grid services such as personal proxy management, workflow management, application repository, and grid file browser. Earlier versions of the P-GRADE portal family were based on the GridSphere portal framework. Using WS-PGRADE was not an option for us because at the time it was undergoing the migration from GridSphere to Liferay portal frameworks and it was not released as an open source project yet.

The Grid Enabled web eNvironment for site Independent User job Submission (GENIUS) portal [12] is based on the EnginFrame portal framework. Its authenticated users can benefit from a robot proxy or download their personal grid proxy from a MyProxy server. The portal provides functionality to submit Triana workflows to the grid and to monitor their execution. Genius was not an option for us because it is based on the proprietary EnginFrame portal framework and we were looking for an open source solution.

4.3.2 Community Specific Grid Portals for Life Sciences

These portals support a specific research community to leverage grid computing. They are designed and implemented based on the existing software stack and requirements of a given research community. Additionally, they are not usually available for download and installation as one package, therefore using them was not an option for us. However, we took their applicable experience and suggestions into account wherever possible.

The MediGRID project [85] implements applications from different biomedical research fields using the Grid Workflow Description Language (GWorkflowDL). It provides a Web-based access to D-Grid resources for end-users with application-specific graphical interfaces. With the exception of guest users with limited functionality, MediGRID portal users store their personal proxy certificate in a MyProxy server for later usage of grid resources. MediGRID is based on the GridSphere portal framework, but in a recent effort a new version of MediGRID portal is under development based on Liferay.

Pandey et al. [116] describe tools and infrastructure for registration of functional magnetic resonance imaging (fMRI) data on the Grid'5000 platform. They also developed a custom Web portal to integrate the workflow editor, execution management, and monitoring tools for the Gridbus workflow management system.

The NeuGRID project [120] ported various brain imaging analysis pipelines into a grid infrastructure and developed high-level services to ensure a generic and extensible infrastructure. They developed a custom Web portal to provide a single point of access and to hide the complexity of the underlying infrastructure. The neuGRID system uses the Laboratory Of Neuro Imaging (LONI) pipeline and Kepler workflow management systems.

The WeNMR project [16] offers a user-friendly infrastructure to perform data analysis for researchers in the field of Nuclear Magnetic Resonance (NMR), Small Angle X-ray Scattering (SAXS) and structural biology. Access to the infrastructure is provided through a portal that integrates commonly used NMR applications and grid technology. The WeNMR grid-enabled portal is based on a custom framework.

The Virtual Imaging Platform (VIP) portal [131] supports execution of medical imaging simulation workflows. It helps users to retrieve their personal proxy certificate from a MyProxy server and then submits MOTEUR workflows [59] to the grid and/or a private cluster through the Distributed Infrastructure with Remote Agent Control (DIRAC) [30] pilot-job framework. It complements grid data management with server-side storage used as a fail-over mechanism in case of file transfer errors. The VIP portal is based on the GWT.

The Distributed Application Runtime Environment (DARE) framework [79] is based on Simple API for Grid Applications (SAGA) [62] and provides the key functionality of job and data management on heterogeneous distributed resources. The Pylons Web application framework is used to build gateways for life science applications on top of the DARE framework as proof of concept.

Several community-specific grid portals have been developed using the P-GRADE grid portal family. Their users should own a personal grid certificate in order to utilize the grid resources through these portals. For example: Molecular Simulation Grid (MoSGrid) portal [17] offers access to molecular simulation codes in quantum chemistry, molecular dynamics, and docking domains. The ProSim Science Gateway [81] supports the bio-scientist research community with high-level and easy to use integrated environments to execute and visualize the results of complex parameter sweep workflows for modeling carbohydrate recognition. The SHaring Interoperable Workflows for large-scale scientific simulations on Available DCIs (SHIWA) portal [84] enables cross-workflow and inter-workflow exploitation of available DCIs.

4.4 System Architecture

The e-BioInfra Gateway system architecture and the e-BioInfra components that support it are illustrated in Figure 4.2. The following color scheme is used throughout this chapter: dark green for client tools, light green for e-BioInfra Gateway components, blue for e-BioInfra components, light orange for grid middleware components, and dark

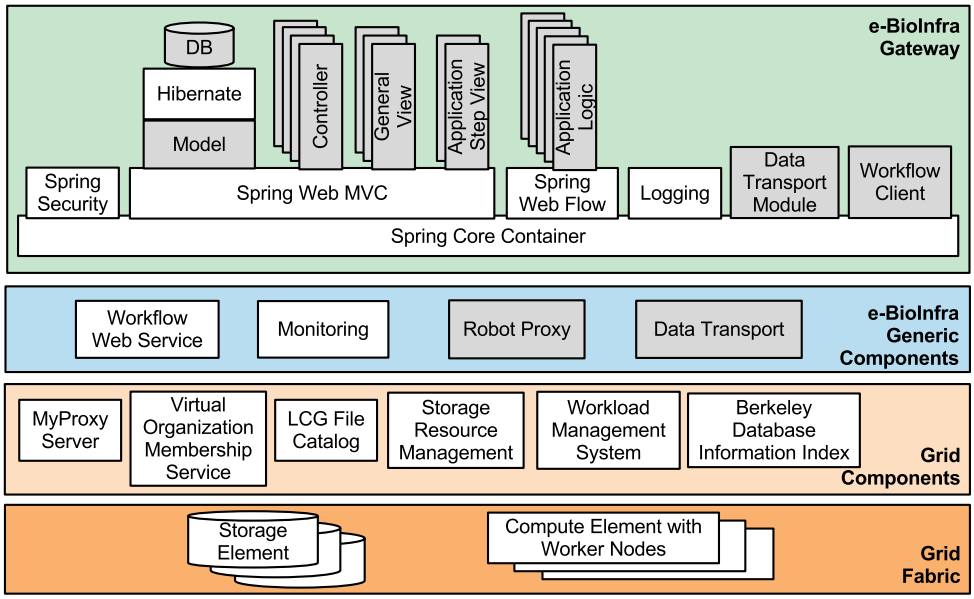


Figure 4.2: e-BioInfra layered architecture: e-BioInfra Gateway and e-BioInfra generic components, grid components, and grid fabric provided by the Dutch grid. The gateway (topmost layer) is built on top of the Spring framework and several third-party modules. Grey boxes denote components that were added to implement the gateway.

orange for the grid fabric components. The main layers and components are described below in bottom-up order.

4.4.1 Grid Fabric and Grid Components

The grid fabric layer is composed of *compute* and *storage* resources provided by the Dutch e-Science Grid (BiG Grid) project [139]. Access to the resources is granted via membership to the *vlemed* Virtual Organization (VO), using Grid Security Infrastructure (GSI) authentication with X509 certificates.

The grid components layer includes Lightweight Middleware for Grid Computing (gLite) [149] middleware components that are also operated by the BiG Grid project. For grid security, *MyProxy* [14] provides a credential repository from which the grid proxy certificate is retrieved securely when needed, and the *Virtual Organization Membership Service (VOMS)* [2] manages authorization within multi-institutional collaborations. Grid file management is supported by the *LCG File Catalog (LFC)* [135], which maps between logical file names and the physical files (including replicas) on the grid storage resources. The *Storage Resource Management (SRM)* [135] manages physical storage resources transparently. The *Workload Management System (WMS)* [95] schedules, distributes, and manages grid jobs across grid compute resources, and the *Berkeley Database Information Index (BDII)* [15] holds information about the grid resources.

4.4.2 Generic e-BioInfra Components

The generic e-BioInfra components layer provides components that intermediate communication with the grid components, as described in [112].

Workflow Web Service

The *Workflow Web Service* enables workflow submission using a generic Application Programming Interface (API) that can be used with different workflow management systems, and also provides status information about submitted workflows. The current implementation is based on the MOTEUR workflow management engine [59], which is used to enact workflows described in the GWENDIA language [105] on the grid resources. MOTEUR parses the workflow description, determines the tasks to be performed on the given inputs, and dispatches tasks to the DIANE pilot job framework [106]. As illustrated in Figure 4.3, MOTEUR and DIANE are installed and operated as e-BioInfra generic components, which are wrapped by the Workflow Web Service. DIANE uses a master-worker model that creates a master for each user, which then submits worker agents as grid jobs to the gLite WMS. When the worker agent is running on a Worker Node (WN), it calls back the master and requests for a task, which is the actual workflow task passed to DIANE by MOTEUR. Each submitted workflow gets a unique identifier that is used to link individual grid tasks to the workflow owner, manage and monitor its execution, or retrieve all related information for support or debugging purposes.

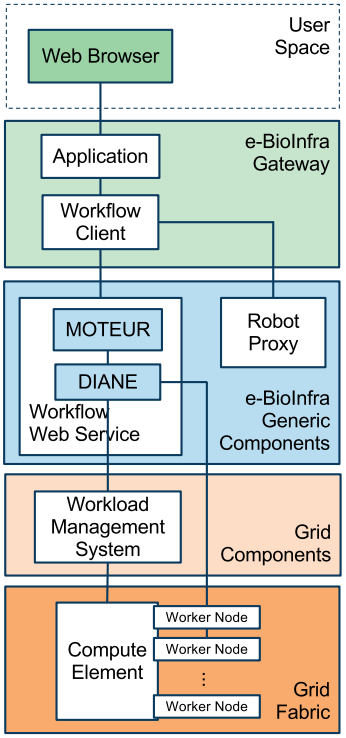


Figure 4.3: Components involved in the execution of grid applications/workflows: e-BioInfra Gateway uses the Robot Proxy and the Workflow Web Service to run tasks on grid using the DIANE pilot job framework.

Monitoring Component

The *Monitoring* component provides information about grid activity, for example about all submitted workflows, the workflow tasks and jobs, DIANE agents, available storage and resource status for provisioning and support purposes. It is possible to search on the database to filter on date, user, etc.

New generic components described below were introduced in the e-BioInfra to better support the gateway functions.

Robot Proxy Component

This component enables researchers who do not have a personal grid certificate to use the e-BioInfra Gateway and the grid infrastructure. It generates a grid proxy certificate from a physical key (hardware token) owned by the gateway, and not an individual person. The hardware token is installed in a secure server owned by BiG Grid and it contains a robot private key that is used periodically to generate and delegate a robot proxy certificate on the BiG Grid MyProxy server. This method has been developed by BiG Grid [168]. A short lived robot proxy is then fetched from the MyProxy server by the gateway components that need to authenticate with grid resources, for example, to transfer data or to submit workflows.

Note that the Robot Proxy component is only accessible by certain e-BioInfra components (i.e., users cannot use it directly). For administrative purposes the robot certificate is linked to a real person who is responsible for grid activities linked to the robot's identity. The gateway keeps records of grid activities performed with this certificate, including individual statistics for each researcher (e.g., workflow identifiers), and reports these to the Dutch or European grid authorities upon request.

Data Transport Component

This component facilitates the transfer of files between the researcher's (local) storage facilities and the grid storage. It is meant to simplify gateway usage for novice users by hiding from them the complexities of grid authentication, file transfer using grid enabled protocols, and file management in large experiments. Figure 4.4 presents how the Data Transport component is related to other e-BioInfra components. Users can choose between two options: to directly transfer files to the Data Transport component using an FTP client, or to upload all files via the Web interface. The first option is more efficient because it avoids transferring all data through the Web server. The second option is simpler to use, in particular for small or few files or one-time experiments. In both cases the files are copied to the grid resources automatically by the component using the robot certificate.

Note that all files transported by this component are assumed to be anonymized to satisfy privacy regulations. As an additional security precaution required at our hospital, in the current implementation the Data Transport component runs within the hospital network and it initiates all transfers, for both incoming and outgoing files. This restriction could be removed for less sensitive data with the deployment of other instances of the Data Transport component.

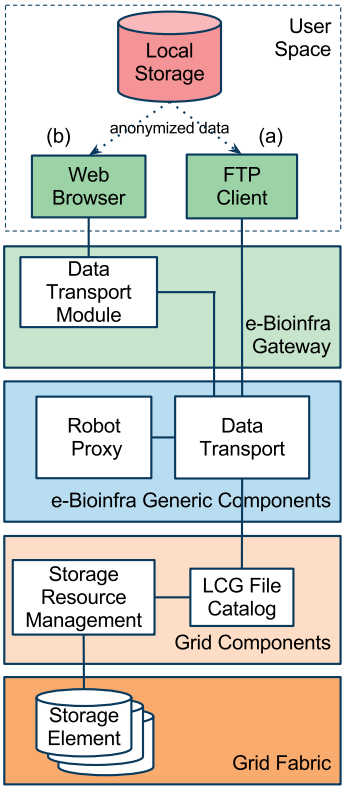


Figure 4.4: The Data Transport component transfers data automatically to/from the grid storage. Users have two options: (a) to directly transfer files through a FTP client or (b) to upload them via the Web interface.

Every user owns a separate directory structure that enforces separation between input and output files, different applications, and experiments – see Figure 4.5. The user is directed to the right directory where to upload the input files for each application and to download output files of each experiment. Users can use a flat or hierarchical directory structure inside the application input directories to organize their files. The Data Transport component is tightly coupled with the gateway, being configured to transfer files to and from pre-configured locations on the grid storage resources. The directory structures on the Data Transport component and on the LFC are identical. The basic principle is that all files put by the user in the "inputs" directory are automatically mirrored to the directory with the same name on the LFC. Similarly, all files resulting from grid workflows are stored in the "results" directory and mirrored back by the Data Transport component automatically. Note that the Data Transport component is only an intermediate stepping stone to transfer files incrementally between the local storage and the grid storage. This component is hidden to the grid resources. This means that running applications still access data on the grid storage, therefore the Data Transport component would not cause a bottleneck during execution. All files on the Data Transport component are considered volatile in the sense that they will be removed upon receiving the notification from the user when the data analysis for a particular research study has been completed, and the results are published, which can last from

days to months or even years.

```
chapter.4
username-n
    |_ inputs
    |    |_ application-1
    |    :       :
    |    |_ application-m
    |            |_ input files...
    |_ results
         |_ application-1
         :       :
         |_ application-m
                 |_ experiment-id
                 :        :
                 |_ experiment-id
                          |_ output files...
```

Figure 4.5: Directory structure in the Data Transport component and on the grid storage.

4.4.3 e-BioInfra Gateway

The gateway (topmost layer in Figure 4.2) is a Java-based Web application developed on top of the Spring framework [161], which is an open source application framework for the Java platform. We chose the Spring framework because it contains minimal yet effective set of components. It provides several generic modules, for example, an authentication and role-based authorization module, and a Model–View–Controller (MVC) [103] based Web application framework. Additionally it allows to port the gateway functionalities and applications to a Java-based portal framework, such as Liferay or WS-PGRADE, with minimal effort in the future.

The *Spring Core Container* manages and connects together the e-BioInfra Gateway modules. The most important gateway modules are briefly described below.

The *Spring Security* is used to handle the user session management. User roles specify which applications and functionalities of the gateway are available to the user. Other gateway modules query the Spring security framework for user's roles for authorization purposes, and adapt their behaviour accordingly.

The *Spring Web MVC* is an MVC-based Web application framework that executes the respective controller based on the user request. The controller uses the model and other gateway modules to perform the respective function, and redirects the results to the respective "general view" to show the outcome to the user. An example: when a user searches for experiments, the controller in charge enquires the Spring security framework for the user roles. Based on the search criteria and the user roles, it performs the search on the model. Finally it redirects the results to the respective view to show them to the user.

Hibernate is an Object-Relational Mapping (ORM) framework [151] used here to perform automatic mapping between the persistent objects in the gateway model and

the relational database (see Figure 4.6). Note that the gateway model also includes non-persistent objects which are not stored in the database.

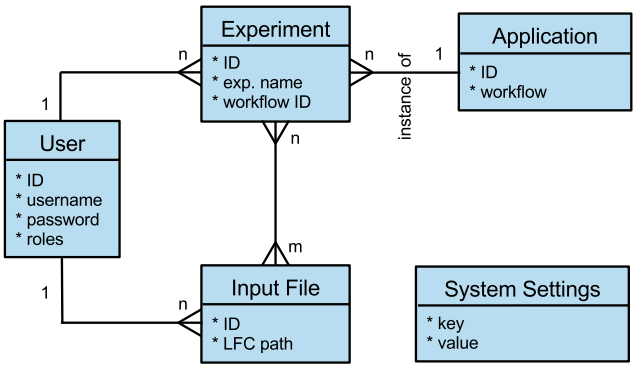


Figure 4.6: Simplified database schema of the e-BioInfra Gateway where only most relevant table and columns are shown.

The *Spring Web Flow* is configured to act as controller for data analysis applications, facilitating the addition of new applications to the gateway (see Section 4.5.1). The business logic of each application is defined as an XML file that specifies the series of the steps that should be taken to start a new experiment, also known as a *flow*. These steps usually include environment initialization for the current user, uploading input files (if there is no input file available) and selection, configuration of experiment parameters, workflow submission, and recording of the experiment metadata in the database. The Spring Web Flow presents each application step to the user by one of the "application step views".

The *Logging* module is used by all gateway modules to log all system events (including intra- and inter-component events) and user activities in a well-structured form. It is implemented using the Simple Logging Facade for Java (SLF4J) API and Log4j logging libraries.

The *Data Transport Module* implements an interface between the gateway and the Data Transport component in the e-BioInfra generic components layer. It also provides the grid storage related functionalities such as directory listing and file transfer.

The *Workflow Client* invokes the Workflow Web Service to submit workflows and query status. Workflow execution uses the Robot Proxy and the underlying generic components as illustrated in Figure 4.3.

4.4.4 Component Interactions

Figure 4.7 illustrates the messages passed between the system components when a researcher/user performs a data analysis experiment using the e-BioInfra Gateway:

(i) The gateway and the Data Transport component periodically obtain a valid robot proxy.

(ii) The user authenticates and chooses an application.

(iii) The user uploads input files to the Data Transport component.

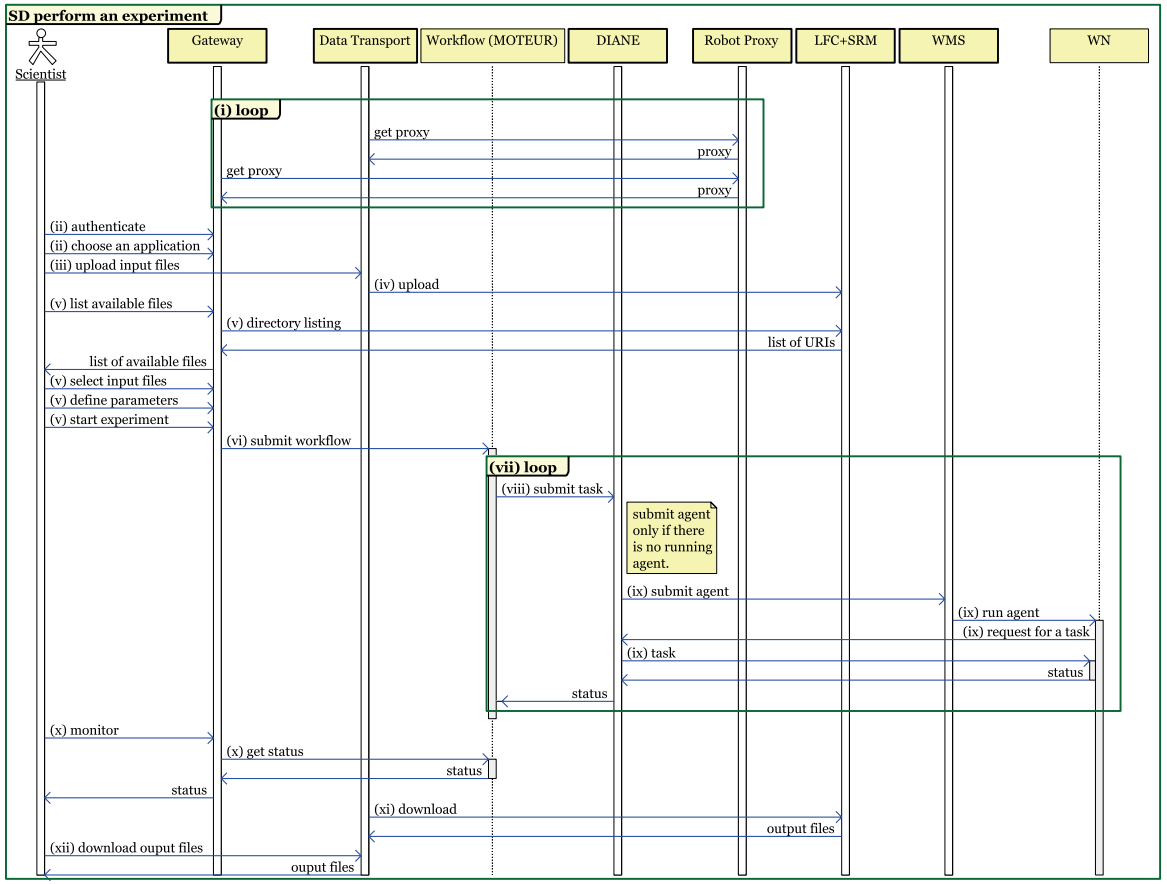


Figure 4.7: Simplified sequence diagram which illustrates the messages passed between the system components when a researcher performs an experiment.

(iv) The Data Transport component uploads the input files to the grid storage and registers them in the LFC.

(v) The user selects input files from the list of his/her available files, defines parameters for a new experiment, and starts it.

(vi) The gateway Workflow Client submits the workflow to the Workflow Web Service.

(vii) The Workflow Web Service launches MOTEUR to enact the workflow, which generates tasks for each combination of process/input.

(viii) MOTEUR submits tasks to DIANE.

(ix) DIANE submits worker agents to the WMS. The worker agents call back the master and request for tasks.

(x) In the meantime the user monitors experiment execution. The gateway retrieves the workflow status information from the Workflow Web Service to enable this.

(xi) The Data Transport component automatically downloads the results from the grid storage.

(xii) When the workflow finished successfully the user downloads the results from the Data Transport component.

4.5 Results

In this sections we present the results obtained with the gateway implemented using the system architecture described in Section 4.4. The following aspects are taken into account: how e-BioInfra developers extend the gateway with new applications, how biomedical researchers use it, and how administrators keep it up and running. Researchers and administrators interact with the gateway using the Web interface. The only necessary software are a commodity Web browser and an FTP client, which are usually pre-installed on every user workstation. Developers use programming interfaces instead of Web interfaces. Gateway usage statistics are presented in Section 4.5.4.

4.5.1 Extending the Gateway with New Applications

When adding a new workflow as an application to the gateway, minor modifications are necessary to the workflow parametrization to enable configuration of a grid directory in which the results are to be written. This is necessary to enable coupling between the gateway and the Data Transport component, where the applications and experiments use different pre-defined directories for the input and output data. This result location is given automatically at runtime and transparently managed by the gateway.

Subsequently, the following steps are needed to integrate a workflow as a new application in the gateway: Firstly, e-BioInfra developers specify the sequence of Web-based forms to be filled by the users to start a new experiment, which is the application

logic described in the Spring Web Flow XML. Secondly, they define the presentation of each form (application step view). There are generic application step views available for common steps, such as input files upload and selection. Finally, they implement the Java classes that implement the application-specific extensions to the model. These extensions are not stored in the gateway database (non-persistent objects) but they are required, for example, to hold experiment parameters as specified by the users. Implementing these steps requires basic knowledge of Java, JSP, HTML, and XML. When a new application is implemented and added to the gateway, it should be compiled, packaged, and deployed on the server.

The gateway implements common functions such as user and role management, workflow submission, grid directory listing, and data upload/download (see also Section 4.4.3). The gateway source code also provides developers with templates and examples that can be adapted to implement new applications.

The gateway has been extended with seven applications, four for medical imaging and three for next generation sequencing data analysis:

- Freesurfer: implements segmentation of structural Magnetic Resonance Imaging (MRI) data with the Freesurfer toolbox [48].
- DTI-preprocessing: format conversion and quality control of Diffusion Tensor Imaging (DTI) data [40].
- DTI-atlas: registration of DTI data for the construction of an average brain (atlas) for a set of scans [25].
- BEDPOSTX: local modeling of diffusion parameters with the FMRIB Software Library (FSL) Bayesian Estimation of Diffusion Parameters Obtained using Sampling Techniques for modeling crossing fibers (BEDPOSTX) [50].
- BLAST: DNA sequence alignment with Basic Local Alignment Search Tool (BLAST) [91].
- T/B-cell Variation: analysis of variation in a specific cell type in different organisms [82];
- Genome Compare: comparison of genomes of related species based on the occurrence of common genes [54].

The last two of these applications have been integrated by developers who were not involved in the gateway or e-BioInfra development, nor have experience with grid software development. It took them less than a week to study the gateway application development documents, implement and test a new application on the gateway. These examples show that the gateway can be extended with additional applications in a straightforward manner, with limited support from the gateway development team.

4.5.2 Using the Gateway

Username and password are used for authentication, and authorization is based on predefined *roles*. After successful authentication, researchers can transfer files from their

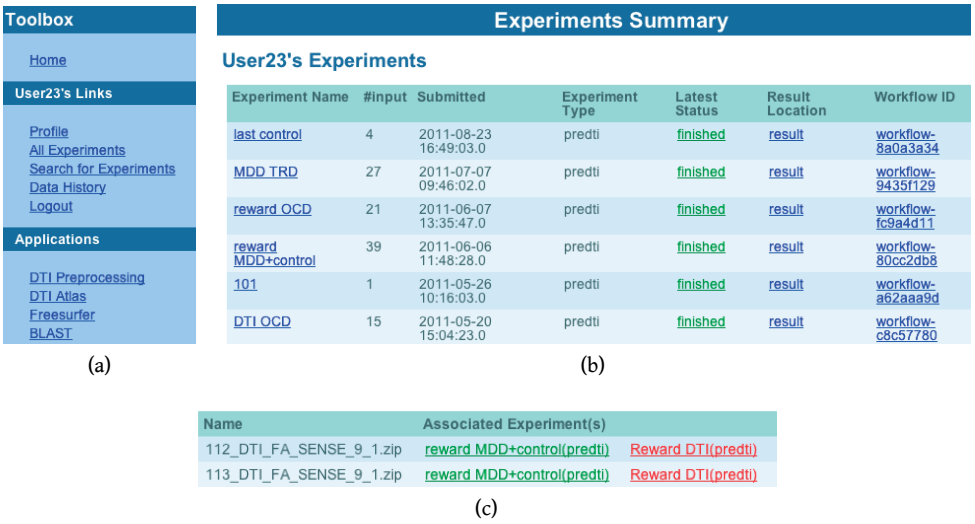


Figure 4.8: Screenshot of the e-BioInfra Gateway GUI for a biomedical researcher. *(a)* Tool bar with links to different gateway functions. *(b)* Summary of all experiments performed by one user. *(c)* Data history summary, showing in red failed experiments and in green successful ones.

workstation or some server to the Data Transport component. Files are automatically transferred to the grid resources, and can be used by workflows for processing. The researcher can choose one of the available applications (see Figure 4.8a), and execute it after specifying input files and setting parameters. Experiments can be tagged with meaningful names for future reference. All activities are recorded by the system, and can be used by the researcher to monitor experiments. The summary pages include number of input files, submission date, application type, status, etc., using color coding to facilitate reading (see Figure 4.8). A link to the monitoring page of the workflow execution is also available for advanced users. After an experiment finishes successfully the researcher downloads the resulting output files from the Data Transport component into his/her local environment for further analysis.

Researchers can also search for experiments based on name, date, status and application type, or additionally inspect how the data uploaded to the Data Transport component has been used as input in various experiments.

4.5.3 Maintaining the Gateway

Administrators are users with a special role that enables them full access to the e-BioInfra Gateway functionalities. After authentication, administrators are provided access to system-wide search, monitoring information, experiment summary and data history for all users and experiments. Additionally, the administrator can use the Web interface to manage users and roles, configure application settings, and modify system

configurations, e.g., the location of generic e-BioInfra components.

When a user registers to the gateway, a profile is created and an administrator is notified per email; the account is kept as "inactive" until the user has been authorized manually by the administrator. The administrator reviews the new user's request information, which includes intentions of usage containing data volume and application. The directory structure for the new user is set in the Data Transport component and on the grid LFC, which is done semi-automatically by a script. Finally the administrator activates the new account after assigning specific roles to it, for example, authorizing the execution of selected applications. The new user needs to agree with the usage policies before getting access to the gateway.

In principle the execution of experiments by a researcher is performed automatically, and the administrator in charge is notified via email only when an error occurs. For example, if the workflow client of the gateway cannot connect to the Workflow Web Service, or the gateway cannot retrieve a fresh robot proxy, the administrator gets a report of the system events led to the problem. The administrator investigates the problem by looking further into the gateway logs and tracing the problem down into the generic components layer.

4.5.4 Usage Statistics

The e-BioInfra Gateway has been in operation for 19 months at the AMC. It has been used in several biomedical research projects, as well as for educational purposes, in teaching medical students the principles of data analysis. Below we present the statistics and analysis of its usage in the period of 01 Jan 2011 to 31 Jul 2012. We exclude data related to gateway activity by users with the goal of development (e-BioInfra developers) or teaching (student accounts).

A summary of usage statistics is presented on Table 4.1. The total execution time indicates the duration of all grid jobs executed as workflow tasks via the gateway, including failed tasks for which logging information was available. Note that the workflows performed via the gateway produced 10 times more data than the input, including both the final and intermediate results.

Table 4.1: Summary of the usage statistics of the e-BioInfra Gateway.

Number of Active users	18
Number of Submitted workflows	926
Total execution time	34,891.7 (h)
Total size of input data	67.7 (GB)
Total size of output data	510.2 (GB)

To better understand the impact of the gateway in the e-BioInfra activity, we compare the total execution time of workflows started via the gateway and via other interfaces. This includes all workflow submissions using the robot certificate and any *vlemed* VO member personal certificate respectively. Figure 4.9 illustrates the execution

time of all workflows started by all users who computed a minimum of 50 hours on the grid during the period and excluding the e-BioInfra developers. Users 02, 04, 09 and 11 are advanced users who perform data analysis as well as workflow development. Users 12 and 14 executed only a limited set of workflows which have been recently integrated into the gateway. Only Users 04 and 11 also have accounts on the gateway, and both of them were involved in designing the gateway as representatives of their research communities.

Note that 32% of all workflow execution time was spent in workflows submitted via the gateway by researchers who had never used a grid infrastructure before. These novice users were attracted by the low entry-point provided by the gateway to perform large computations on the grid infrastructure. For example, Peters et al. [118] and Wingen et al. [175] have already published results based on the data analysis performed via the gateway.

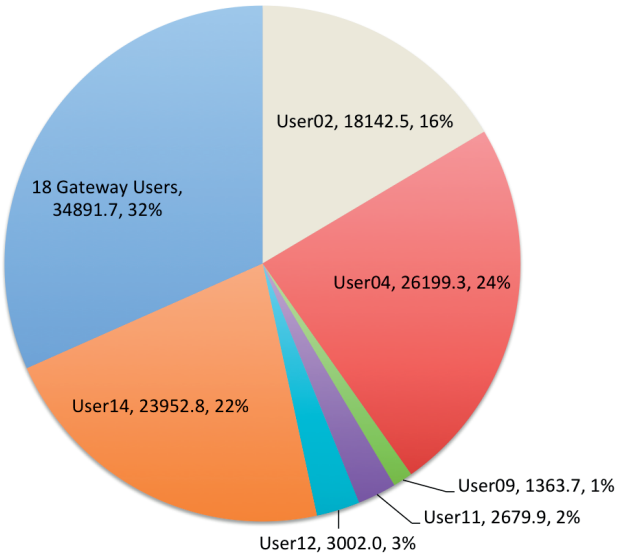


Figure 4.9: Total execution time (hours) of all workflows executed via the gateway (collectively shown as Gateway Users) as well as by *vlemed* VO members (UserX).

Table 4.2 presents workflow execution time for each of the applications. Together, the medical imaging applications of the gateway consumed almost 98% of the total execution time.

4.6 Discussion

Service provision through a Web interface proved to have various advantages. First, it eliminates operating system compatibility issues between the user's workstation and the e-BioInfra platform. Secondly, it removes the need for installation and configu-

Table 4.2: Total execution time (hours) by e-BioInfra Gateway users per application.

Application	Total time (h)
Freesurfer	27,892.6
DTI-preprocessing	3,659.1
BEDPOSTX	1,513.6
DTI-atlas	1,428.9
BLAST	214.6
T/B-cell Variation	178.5
Genome Compare	4.3

ration of specific clients from the user's shoulders. Both have been experienced as a burden in the case of the VBrowser, in particular by novice users. And finally, the Web interface is much simpler for them to use than conventional CLIs and dedicated applications such as the VBrowser. Additional features introduced in the gateway, such as the Robot Proxy and Data Transport components, further contributed to facilitate access to the e-BioInfra for novice users. These researchers now use the platform regularly, being responsible for 32% of the *vlemed* VO activity in 19 months (around four CPU years) and demonstrating good user acceptance. Advanced researchers continue to develop new workflows, and the most popular can be integrated into the gateway in a straightforward manner. The portfolio of applications however is still very limited, and needs to be significantly extended to better address the AMC researcher needs.

Although the researchers in principle can use the e-BioInfra Gateway autonomously, our experience shows that involvement of a field expert for application-specific training and troubleshooting support is essential. These domain experts can also write application-specific documentation, help in input data preparation and monitoring, sort out data formatting constraints, etc. In particular troubleshooting is still a challenge for the biomedical researchers, since it is still difficult to distinguish errors caused by malfunction of the generic components (e-BioInfra, grid), by inadequate data, or failure in a method to run correctly for a particular dataset. In many cases it requires detailed information and knowledge about the system components and services, which are more easily understood by developers and administrators. Troubleshooting therefore involves intervention of various persons with complementary expertise.

The usage statistics in Section 4.5 show that neuroscientists running medical imaging applications form the largest user community of the gateway, in contrast to bioinformaticians who used it infrequently. This can be explained by the fact that bioinformaticians usually have more technical expertise (e.g., they know how to program, compose workflows, operate Linux systems, etc.) and therefore the learning curve of using more flexible and more sophisticated interfaces is less steep for them. Moreover, the medical imaging applications capture complex pipelines that have been developed along many years, which are now considered stable and require no or minimal adaptation. This seems to be different for sequencing applications which still need to be adapted

to perform some particular data analysis task. This means that the functionalities currently provided by the gateway are not satisfactory for these advanced researchers because they require more flexible interfaces to manage their applications, data, and experiments.

Finally, a remark about the software architecture and technologies. Whereas there is a clear trend to use portal frameworks to implement science gateways (see Section 4.3), such technology was not our first choice at the time due to the design considerations mentioned in Section 4.2.4. Instead we chose for an intermediate, light-weight, and effective solution to implement the gateway. The usage of a portal framework creates an extra layer in the software stack that requires mastering for management and maintenance. Moreover, the programmer needs to learn at least one more technology, which is the portlet API, in addition to the technologies required for Web application development, for example, MVC and ORM frameworks. The gateway was also designed to facilitate porting its tools and applications to a portal framework with minimal effort. We are currently in the process of porting the gateway to the Liferay-based WS-PGRADE [75] portal framework.

4.7 Conclusion and Outlook

In this chapter we presented our effort to reduce the roadblocks for running biomedical data analysis on the Dutch grid by hiding the complexity of the underlying infrastructure from the researchers. The e-BioInfra Gateway is now used in several biomedical research projects. By means of the gateway, biomedical researchers have shortened the "raw data to results" time, since they no longer needed to manually process each individual dataset. Both of these usage stories show the success of this approach in decreasing the learning curve and increasing the pace of biomedical data analysis.

Currently users cannot submit their own workflows through the gateway, mainly because the gateway is bound to the robot proxy and it is against grid regulations to use the proxy in an uncontrolled environment. We plan to extend the gateway functionalities to accept personal grid proxies and to support custom workflow submissions without compromising the user-friendliness of the gateway.

Another problem we face today is that complexities hidden behind the gateway become visible when errors occur, and require intervention of the support team. In a recent effort we are looking into redesigning the interfaces between the current system components to make them more aware of each other to improve fault detection and propagation in context. We are also looking into exposing just enough information in a well-structured way according to each user's role, skills and preferences.

Finally and most importantly we are planning to improve data management via the gateway by providing better connectivity with the data servers owned by the researcher communities/labs. In this way we will attempt to make the e-BioInfra Gateway the virtual space not only for processing data on the grid infrastructure, but also the front-end to enable collaboration among the researchers that can share data, methodology and expertise [128].

Acknowledgments

We are grateful to: C. Byrman and P. de Boer for software components that are used in the gateway; M. Caan, V. Korkhov, A. Luyf, S. Madougou, Y. Mohammed, B. van Schaik, and M. Willemsen for developing workflows and/or extending the gateway; J.J. Keijser, T. Visser, and other members of BiG Grid for providing support; And the anonymous reviewers for their valuable comments and suggestions that improved the quality of this paper.

This work is financially supported by the AMC ICT innovation fund, the BiG Grid programme funded by the Netherlands Organisation for Scientific Research (Dutch: Nederlandse Organisatie voor Wetenschappelijk Onderzoek) (NWO), and the COMMIT/ project "e-Biobanking with imaging for healthcare". This work used resources of the BiG Grid project, the Dutch e-Science Grid.

CHAPTER 5

A Data-Centric Neuroscience Gateway
Design, Implementation, and Experiences

In Concurrency and Computation: Practice and Experience, 27(2):489–506, 2015.

Shayan Shahand, Ammar Benabdelkader,
Mohammad Mahdi Jaghoori, Mostapha al Mourabit,
Jordi Huguet, Matthan W.A. Caan, Antoine H. C. van Kampen,
Sílvia D. Olabarriaga

Abstract

Science gateways provide user interfaces and high-level services to access and manage applications and data collections on distributed resources. They facilitate users to perform data analysis on distributed computing infrastructures without getting involved into the technical details. The e-BioInfra Gateway is a science gateway for biomedical data analysis on a national grid infrastructure, which has been successfully adopted for neuroscience research. This chapter describes the motivation, requirements, and design of a new generation of e-BioInfra Gateway, which is based on the grid and cloud user support environment (also known as WS-PGRADE/gUSE framework) and supports heterogeneous infrastructures. The new gateway has been designed to have additional data and metadata management facilities to access and manage (biomedical) data servers, and to provide data-centric user interaction. We have implemented and deployed the new gateway for the computational neuroscience research community of the Academic Medical Center of the University of Amsterdam. This chapter presents the system architecture of the new gateway, highlights the improvements that have been achieved, discusses the choices that we have made, and reflects on those based on initial user feedback.

DOI: 10.1002/cpe.3281

5.1 Introduction

Science Gateways support scientists in e-Science endeavors. De Roure et al. [32] described the requirements of e-Science environments as a spectrum with two ends. One end is characterized by automation, virtual organizations of services, and the digital world, and the other end is characterized by interaction, virtual organizations of people, and the physical world. Orthogonal to these requirements at both ends is the issue of scale, for example, of virtual organizations, computation, storage, and the complexity of relationships between them. Increasing scale demands automation and, as highlighted by Hey and Trefethen [68], computer scientists have the research challenge of creating high-level intelligent services that genuinely support e-Science applications. Such services, e.g., Science Gateways (SGs), should go beyond straightforward access to computing resources, and also include support to construct and manage virtual organizations, as well as to manage the scientific data deluge in the scholarly cycle including hypothesis, experimentation, analysis, publication, research, and learning.

A large number of communities are therefore facing the challenge of building SGs. A recent collaboration resulted in the European Grid Infrastructure (EGI) Science Gateway Primer [43], where issues involved in SG design, implementation and operation are presented and discussed. According to this Primer, SGs are desktop or Web-based interfaces to a set of applications and data collections. SGs comprise front-end and back-end components, and they offer services that facilitate access to computing and storage resources, as well as services provided by Distributed Computing Infrastructures (DCIs). Moreover, SGs support collaboration between researchers through exchange of ideas, tools and datasets. From a functional perspective, there are *SG frameworks* and *SG instances*. SG frameworks implement generic functionalities such as security, workflow and data management, and DCI access; examples are Web Service – Parallel Grid Run-time and Application Development Environment / grid and cloud User Support Environment (WS-PGRADE/gUSE) [77], Catania SG [145], Hubzero [100], and Distributed Application Runtime Environment (DARE) [92] frameworks. SG instances are community-specific science gateways, with tailored interfaces and services for a specific application domain. SG instances can be be built using SG frameworks or with custom software stacks. There are currently a large number of SG instances, which will be discussed in more details in Section 5.6.

The e-BioInfra Gateway [130] was a SG instance for analysis of large scale biomedical data on the Dutch e-Science Grid [162]. It was designed to simplify usage of this infrastructure by biomedical researchers by providing services such as community grid certificate and semi-automatic file transport to and from the grid resources. It was implemented based on a custom framework and deployed at the Academic Medical Center (AMC) of the University of Amsterdam (UvA), The Netherlands. Since its deployment in production (early 2011), researchers have successfully performed large computations on the Dutch grid infrastructure via the e-BioInfra Gateway with minor help from the support team. For example, Peters et al. [118], Wingen et al. [175], Rienstra et al. [121], and de Kwaasteniet et al. [87] have already published results of their neuroscience research based on the data analysis performed via the e-BioInfra Gateway. The gateway structured the system information and allowed for extensions with new data analysis methods. This enabled (external) developers to extend it with

ten applications, six for medical imaging, three for next generation sequencing data analysis, and one as part of the mass-spectrometry tool-chain. Thirty users utilized the e-BioInfra Gateway, most of which were from the AMC Brain Imaging Center (BIC) [137].

Although the e-BioInfra Gateway can be considered a success story, our experience indicated the need for further improvement in particular regarding support for richer data resources and operations. We therefore designed, implemented and tuned a new gateway specifically for the computational neuroscience research community of AMC, coined *AMC computational NeuroScience Gateway (AMC-NSG)*. The most important new features of the new gateway include data and information management, support for metadata that is used and generated during the execution of complex data processing; and additional functionalities for system operation and administration.

In this chapter we discuss the motivation for a new gateway in Section 5.2, and describe a requirement analysis for it in Section 5.3. The system architecture is explained in Section 5.4. Initial user feedback is presented in Section 5.5, followed by comments about related work in Section 5.6. Finally, a discussion and conclusions are presented in Sections 5.7 and 5.8.

5.2 Motivation for a New Gateway

In this section, the functionalities of the previous gateway are briefly explained and the experiences gained during its operation are reflected upon. These experiences motivated the design of the next generation of the e-BioInfra Gateway.

5.2.1 The Previous Gateway

A detailed description of the previous e-BioInfra Gateway can be found in [130]. In a nutshell, the user could authenticate with username and password, select the application to run, select the input files and other parameters, and start a so called *experiment*. Then she/he could monitor the experiment and, when finished, retrieve the results.

The processing on grid resources was handled by the MOTEUR [59] Workflow Management System (WfMS) and the DIANE pilot job framework [106]. Minimal provenance information was recorded, namely the names of the files used in an experiment and the history of changes in an experiment status. More detailed provenance information about workflow execution was collected and displayed by a separate system after the processing was completed [94].

Because medical imaging data files are typically large, their transport was not done directly via the e-BioInfra Gateway Web interface, but via an FTP server that was located in the trusted network of the hospital and synchronized automatically with a directory on the grid storage resources. Therefore, the user had to upload the data to the server before performing the steps above, and could retrieve the results from the same FTP server when the experiment was completed.

5.2.2 Reflections about the Previous Gateway

The functionalities offered by the previous gateway mainly included: transparent authentication and authorization with grid resources using a robot certificate; semi-

automatic data transfer between gateway and grid storage workflow processing management, including logging and monitoring; and an extensible set of applications for various biomedical domains. These functionalities were mainly organized around applications, underlying resources, and their frameworks.

In almost two years of experience with gateway operation and user support, we faced challenges discussed below that made us realize that the gateway should be organized around *data* instead of *applications*.

In the previous gateway, a large number of errors were caused by invalid input data, which are referred to as *data-related* errors. Users typically had difficulty to prepare files for processing with the gateway applications, which involved the steps for file (re-)formatting, naming, and transport. Also, they were not aware of the types of data that can be processed by each application. Although the data-related errors were significantly reduced after training or reading the user manual, we realized that the data preparation and transport process should be improved with further automation.

Whenever the errors were not data-related, they were mostly related to the changes or maintenance operations performed on the grid infrastructure, which are referred to as *computing-related* errors. Exposing users to the computing-related errors turned out to be both unnecessary and overwhelming for the users. A system administrator usually could fix those errors by simple actions such as resource blacklisting and resubmitting the failed experiment. However, it was not straightforward in the old gateway to resubmit parts of the experiments on behalf of the users.

Another necessary improvement was motivated by the evolution in the computing infrastructure. Originally the e-BioInfra Gateway was meant to facilitate access to grid resources. However, in the past years other resources have become available for research, such as local clusters at the AMC and a national High-Performance Cloud. Another solution was required to exploit these additional resources.

Finally, the need for adopting a more sustainable software stack was evident. Although our custom framework fulfilled the needs at first, as a small research group it was difficult to maintain and extend it. In particular, keeping up with all the developments related to DCIs requires significant effort and expertise that can be achieved by utilizing a SG framework.

5.2.3 Preconditions for a New Gateway

Recently the neuroscience research community of AMC has decided to adopt a data server for their research scans. The data is generated by the scanner and directly imported into the data server, which keeps both the raw data and the metadata. To facilitate research data processing, the Radiology department decided that this research data server could be connected to the gateway.

Due to security regulations for processing medical research data, the data server is hosted inside the AMC firewall. The in-house computing clusters and national grid computing resources that are used for data processing are located inside and outside the AMC network respectively. The AMC-NSG server is located in the DeMilitarized Zone (DMZ) of the AMC network, which means that only some of the gateway services are visible from outside the network and, similarly, only some of the data server and in-

house computing cluster services are visible from the DMZ. Figure 5.1 illustrates the resources related to the AMC-NSG and their network location.

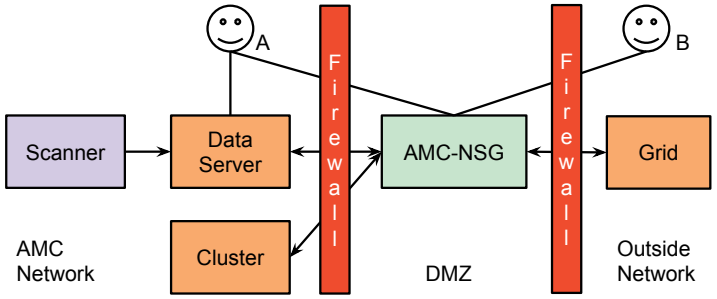


Figure 5.1: The resources related to the AMC-NSG and their network location: inside or outside the AMC network, or at the demilitarized zone. User *A* is inside the network and can access the data directly. User *B* is outside of the network and therefore only has access to the gateway and limited metadata.

The envisioned usage scenario for this system is as follows. The users inside the AMC network can import their data into the data server either directly from the scanner or by uploading it to the data server. The data is automatically preprocessed according to pre-defined rules, for example, it is pseudonymised and converted to a more compact format, and its metadata is extracted from the Digital Imaging and COmmunication in Medicine (DICOM) headers [37]. The users, both inside and outside the AMC network, are able to query and filter the data based on its metadata, initiate and monitor data processing, regardless of their location. After the data processing is completed, users should be able to download the results from the data server only if they are inside the AMC network. The system administrators, who are also located inside the AMC network, should be able to monitor all data processing activities and inspect them in more detail if any error happens.

5.3 Requirements Analysis

In a previous work, we described in detail the typical phases of computational neuroscience studies in [128], namely *study design, data acquisition, data handling, processing, analysis, and publication*. Based on the analysis of these phases, the actors who are involved in each phase, and the tasks that they perform, in that chapter the properties and functionalities of SGs to support computational neuroscience research communities are identified. In summary, the required properties and functionalities include: sharing of data and methodology; satisfying security and privacy regulations; scalable, transparent, and flexible management of storage and computing resources; literature discovery; collaboration support; metadata, data, workflow, and provenance management; and visualization.

The design of the new gateway presented in this chapter takes into account these desired properties, as well as the experiences and preconditions presented in Section 5.2.

In particular, we focused on additional functionalities that would put *data* at the center of interaction between the user and the gateway. A *data-centric* gateway should provide necessary tools and services to the users to interact with their data, for example, for data discovery, exploration, preparation and processing.

The following functionalities should be provided by the new data-centric gateway:

1. Unified, secure, and easy access to data and related metadata stored on distributed and heterogeneous data servers. Users should be able to transparently query, explore, process, and analyse data from a single interface, without bothering about the data location or format, or how the data is retrieved and transported for further processing.

2. Automatic and interoperable file transport and processing on different infrastructures (e.g., data servers, cluster, grid). Low level technical details should be hidden from the users, such as different communication protocols, middleware services, and authorization mechanisms.

3. Assistance for users to choose the correct data processing method based on metadata.

4. Automatic provenance information collection about the methods, parameters and input files used for processing. This provenance information can be used in troubleshooting, to track the data lineage, and for statistics.

5. Single sign-on facility to authenticate and authorize transparently to various computing and storage resources using user or community credentials.

6. Streamlined operations of the gateway by its system administrators. They should be able to access log files easily, communicate the causes of errors with the users, and restart the faulty data processing on their behalf.

In addition to these functionalities, the new gateway should be:

1. *extensible*, to easily connect to new data or compute resources, and accommodate new data types, applications, and user groups;

2. *customizable*, to support preferences and configurations for both end-users and system administrators;

3. *scalable*, to gracefully support the growth of user community and its needs for resources, as well as infrastructures capacity and heterogeneity; and

4. *sustainable*, to be able to maintain the gateway software with minimal costs, while its underlying infrastructure changes.

5.4 System Design and Implementation

Figure 5.2 illustrates the layered architecture of the AMC-NSG. At the bottom, the Resource layer (dark orange) with several DCIs (i.e., local clusters and grid) and data resources (i.e., Radiology research data server). These resources are utilized through Middleware Services contained in the second layer (light orange). High-level Services contained in the third layer (blue) provide an abstraction to interact with the middleware, such as workflow management and data transport. Finally, the Presentation layer (green) contains the interfaces for user interaction. The two topmost layers (green, blue) are implemented using generic SG framework components provided by WS-PGRADE/gUSE (at the right), and components developed for the new gateway (at the left). The components of the new gateway complement the functionality of WS-PGRADE/gUSE framework for the specific case of the AMC-NSG.

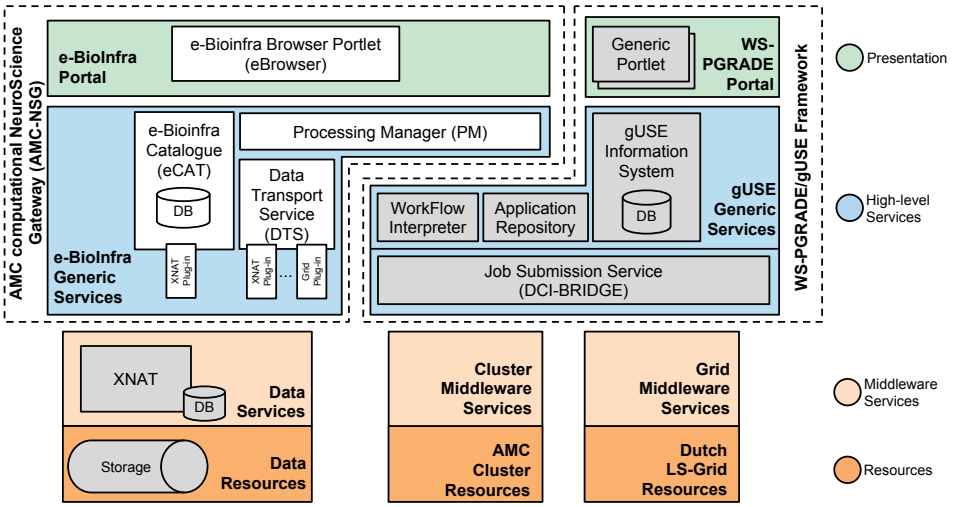


Figure 5.2: Layered architecture of the AMC-NSG based on the WS-PGRADE/gUSE SG framework. Grey boxes represent existing third-party components, and white boxes denote new components. See text for more details.

The core of the AMC-NSG is made of the following components: e-BioInfra Catalog (eCAT), Data Transport Service (DTS), Processing Manager (PM), and e-BioInfra Browser Portlet (eBrowser). They are loosely coupled and communicate via well-defined Application Programming Interfaces (APIs), an approach that paves the road towards a service-oriented architecture and facilitates their reuse to build other gateways for different scientific applications. These components also utilize the API of the WS-PGRADE/gUSE components to implement the functionalities of the AMC-NSG.

Below the components that are more relevant for a data-centric SG, namely data services and the core components (white boxes in Figure 5.2) are presented in further detail. For completeness the WS-PGRADE/gUSE SG framework is also introduced briefly.

Finally, the interactions between these components are illustrated based on a use case.

5.4.1 e-BioInfra Catalog

The e-BioInfra Catalog (eCAT) has been designed to facilitate the data and metadata management functionalities at the gateway. It is a central store for user and system-level information that uses and implements a data model with the following main entities: `User`, `Project`, `Data`, `Meta-data`, `Resource`, `Credential`, `Application`, `Processing`, `Submission`, and `Submission Status`. The main relationships between these entities are illustrated in Figure 5.3.

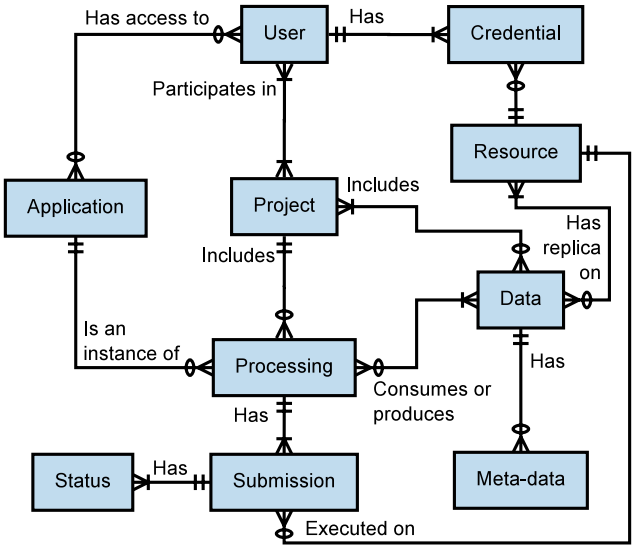


Figure 5.3: Simplified entity-relationship model of the information stored in the e-BioInfra Catalog (eCAT).

In the eCAT data model (Figure 5.3), a `User` participates in `Projects`, which provides the scope for access control. `Data` entities are included in, and are processed within, the scope of Project entities. Each User has one or many `Credentials` that are used by the gateway to access resource(s) on the user's behalf. A `Resource` can be a computing resource, a storage resource, or both. A data server is an example of a data resource, and grid or clusters are examples of resources both for data and computing. Each Data has at least one replica on a data resource and has metadata attached to it. `Meta-data` is represented by a key-value pair. Users have also access to `Applications` consisting of validated and ready-to-use workflows that wrap some legacy code for data analysis. Applications have inputs and generate outputs; they also have affinities with particular data types and formats. The outputs of applications are also stored as Data entities that are annotated with provenance information about the applications that generated them. When a user processes a certain Data with a specific Application, the information about this activity is captured by eCAT as a `Processing` entity. Each Processing includes one or more workflow `Submissions`, depending on the cardinality of input data. A workflow is executed on a computing Resource. The provenance information about the Data consumed and produced during a Processing, the parameters, and the history of

`Submission Status`, are also stored in the eCAT database as relationships and attributes of these entities.

Note that eCAT is not meant to duplicate metadata that is already stored on data servers; instead, it only stores pointers to information on the data servers. The only exceptions are some types of metadata that are specific to user activities on the gateway, and which are not possible, nor of direct interest of research communities, to store in their data servers. eCAT retrieves and stores metadata on heterogeneous data servers through *Plug-ins*, which are software modules attached to eCAT to enable programmatic communications with a specific data server.

5.4.2 Data Servers

A data server can be as simple as an FTP server that contains the data in a hierarchy of directories. However, management of biomedical research data, with its growing size and complexity, requires domain-specific *Information Management Systems (IMSs)* with structured metadata. There are several IMSs for management of biomedical research data and metadata, electronic data exchange, archival, and security, and the research communities start to adopt such systems routinely. Additionally, every community has its own procedure to implement rules and regulations regarding the protection of biomedical research data, as well as policies for data sharing and archiving. Therefore, instead of replicating such efforts, we decided to rely on existing, external, biomedical research data and metadata resources, as well as on their own security mechanisms and policies. In this way, the research community itself provides and manages the IMS, defining data ownership, access policies, and regulating data confidentiality and data privacy.

A popular IMS for medical imaging data and metadata is the eXtensible Neuroimaging Archive Toolkit (XNAT) [96]. XNAT is an open source IMS that offers an integrated framework for storage, management, electronic exchange, and consumption of medical imaging data and its complementary metadata. XNAT provides a rich communication layer based on a RESTful[1] API of resource-oriented Web Services. Due to these qualities, XNAT has been deployed at the Radiology department of AMC to implement a research data server. The XNAT server is connected to the AMC-NSG by agreement between the neuroscience community and the gateway providers. The data becomes available for processing at the gateway for authorized users only. Gateway users should provide their XNAT credentials before they are able to access data and metadata on the XNAT. All of the API calls from the gateway to the XNAT are performed with user credentials.

XNAT implements an extensible data model that also has some fixed entities. In summary, XNAT "users" have access to "projects" that contain "subjects" (i.e., people who have one or more scans) and their "image sessions". Each image session includes one or more "scans" and each scan has a many-to-many relationships with a specific entity called "reconstructed image". a reconstruction image is the result of any processing software. The most relevant XNAT entities for our case are "projects" and "scans", which are mapped respectively to the `Project` and `Data` entities in the eCAT. The "reconstructed image" entity of XNAT is used to store the processing provenance

[1] REST stands for Representational State Transfer.

information for entities generated by the SG. In the new gateway, we developed an eCAT plug-in for XNAT using its RESTful API. This plug-in maps the eCAT data model into the XNAT data model by generating queries and parsing responses between them.

5.4.3 Data Transport Service

The Data Transport Service (DTS) transports data between data servers and storage resources on DCIs. This service contacts the eCAT to retrieve information needed to access the data resources, such as hostname, type of resource and credentials. From this information it determines how to authenticate with the data server on behalf of the user, as well as how to authenticate with the storage resources of the DCI, typically with community credentials. It autonomously performs the data transfer using third-party mechanisms as much as possible to avoid bottlenecks. Similarly to the eCAT, DTS also has *Plug-ins* that implement the necessary functionalities to enable data transfer between resources. Currently plug-ins are available for the XNAT data server and the Lightweight Middleware for Grid Computing (gLite) grid middleware. If some data has been replicated on a DCI, the location of that replica is stored in the eCAT and it can be retrieved later to avoid transporting the file again in the future.

5.4.4 WS-PGRADE/gUSE SG Framework

Web Service – Parallel Grid Run-time and Application Development Environment / grid and cloud User Support Environment (WS-PGRADE/gUSE) SG framework [77] is an open source, workflow- and service-oriented framework that facilitates development, execution, and monitoring of scientific workflows on DCIs. It comprises the Web Service – Parallel Grid Run-time and Application Development Environment (WS-PGRADE) portal and the grid and cloud User Support Environment (gUSE) services. WS-PGRADE is based on the Liferay portal framework, which provides rich facilities for community management and customizable user interfaces. gUSE provides high-level services to access various DCI resources. These qualities motivated the choice for this SG framework to implement our gateway.

The most relevant gUSE services used by our gateway are:

- *Job submission service or DCI-BRIDGE:* provides flexible and versatile access to a large variety of DCIs such as grids, desktop grids, clusters, clouds and service-based computational resources. It also handles authentication and authorization to the configured DCIs transparently.
- *Workflow Interpreter:* parses workflows, submits jobs to the DCI-BRIDGE, and retrieves their status for monitoring and fault-tolerance.
- *Application Repository:* stores ready-to-use tested and configured workflows. These workflows are exported to the application repository by workflow developers, from where they are imported into user space for execution.
- *gUSE Information System:* stores configurations of gUSE services and workflow related information such as workflow executions and their jobs status.

Additional facilities offered by gUSE are also very important for the implementation of our SG. The first is support for community credentials (robot certificates). The other is functionality to pause and resume workflow execution, which is used by the administrator.

The WS-PGRADE/gUSE framework also provides two APIs to create SG instances. We used the *gUSE Application Specific Module (ASM)* API to utilize gUSE services, more specifically the *Application Repository* and the *Workflow Interpreter*.

The WS-PGRADE portal also offers a set of generic portlets to interact with gUSE services via Web-based graphical user interfaces. These portlets are only visible to the developers and administrators of the AMC-NSG. See [77] for the complete description of WS-PGRADE/gUSE services and portlets.

At the time of writing, the WS-PGRADE/gUSE framework did not have any facility to connect to data servers. Moreover, its data transport facilities were also limited. The additional components described above are meant to bridge this gap.

5.4.5 Processing Manager

The Processing Manager (PM) takes care of preparation, submission and monitoring of data processing applications that are executed on a given set of input files. All the details needed to run the application are obtained by querying eCAT, such as the gUSE workflow, DCI, as well as the input ports, the output ports, and the relationship between them.

The steps carried out for each processing started by the user are the following; they are collectively called a Submission in the eCAT data model. First, the PM instructs the DTS to transport input files from the data server to the storage resources of the DCI on which the data processing will be performed. Then, it imports the workflow from the gUSE Application Repository and configures it with the physical location of input data. All workflows are configured to run with community credentials, such as in the previous gateway. The configured workflow is submitted to and executed by the gUSE Workflow Interpreter. The workflow execution is monitored by the PM autonomously. When the workflow execution is completed successfully, the PM instructs the DTS to transport the results back to the data server.

Note that each processing started by the user can generate one or more submissions. This depends on the number the input data files and the relationship between input data and results for each application. In most applications, there is a 1-to-1 relationship between input data and results, that is, one result is generated for each input. In these cases the processing consists of n workflow submissions, one for each of the n input data files. In other applications, a single result is generated for a collection of input data files, and therefore a single workflow is submitted. Submitting one workflow for each processing result, instead of using parameter sweep capabilities of the WfMS, is motivated by the need for fine-grained control and monitoring of workflow execution. It also facilitates linking the results generated at the output ports of a workflow to its inputs, which is necessary for provenance collection. Note that the multiple workflow submissions for a processing are hidden from the user, whereas he/she can obtain progress information about the individual processing tasks transparently.

Each individual submission goes through the states illustrated in Figure 5.4, which correspond to status information shown to the users and system administrators. It is first in the `In Preparation` state, during which the input data is transferred from the data server to the target computing resource, and the workflow itself is imported, configured, and submitted through the gUSE ASM API. After successful submission, it reaches the `In Progress` state, during which the workflow is executed by the WfMS on the target computing resource. When the workflow execution completes successfully, the Submission moves to the `Uploading` state, during which the results are uploaded to the data server. Finally, when all the previous steps were completed successfully, the status changes to the `Done` state, and the results became available for the user via the interface. The user can also abort the submission at any time.

If any problem is detected during any of the operations performed for preparation, submission, workflow execution and data transfers, the Submission moves to `On Hold` state, and a notification is sent to the system administrator. He/she then investigates and troubleshoots the error using information about that particular Submission that is presented on the administrator's dashboard (see Figure 5.7). If the error can be fixed, the workflow is *resumed* and gUSE continues execution from the last successful job. This is often the case of errors related to the DCI, for example a failed job or unavailable file. Otherwise, the administrator *aborts* the submission, which causes it to move into the `Failed` state. At this point a message can be written to the user providing high-level information about the cause of the error and the actions to take. Typically these are data-related errors, as all the DCI-related errors will be handled automatically by gUSE or manually by the administrator.

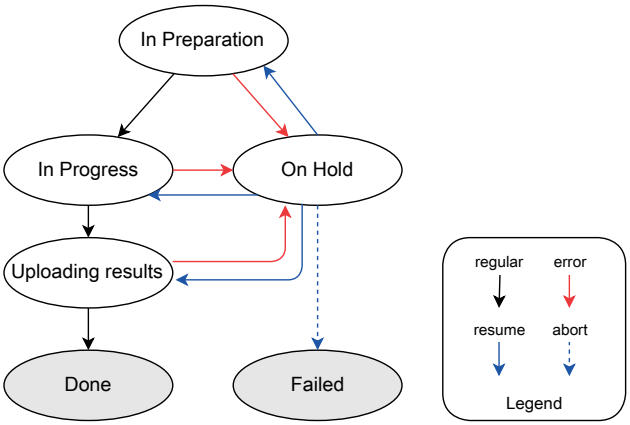


Figure 5.4: Submission state diagram. Red transitions occur on error; blue transitions occur by manual intervention of the administrator.

5.4.6 e-BioInfra Browser Portlet

The e-BioInfra Browser Portlet (eBrowser) is part of the presentation layer. It provides a Web-based user interface to interact with the e-BioInfra generic services. Instead of contacting the services directly, eBrowser retrieves information from eCAT to provide a homogeneous view to the users and system administrators to browse data, projects and

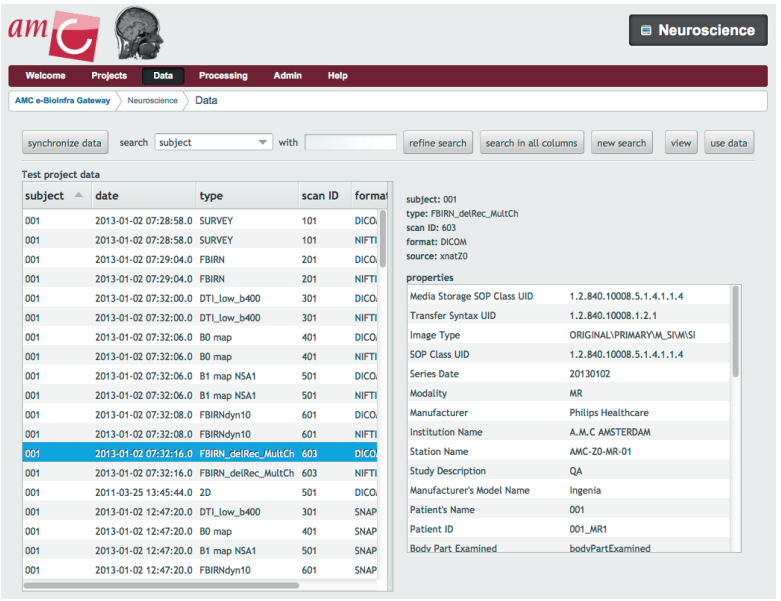


Figure 5.5: AMC-NSG eBrowser user view for data browsing: data list on the left; metadata of selected item on the right.

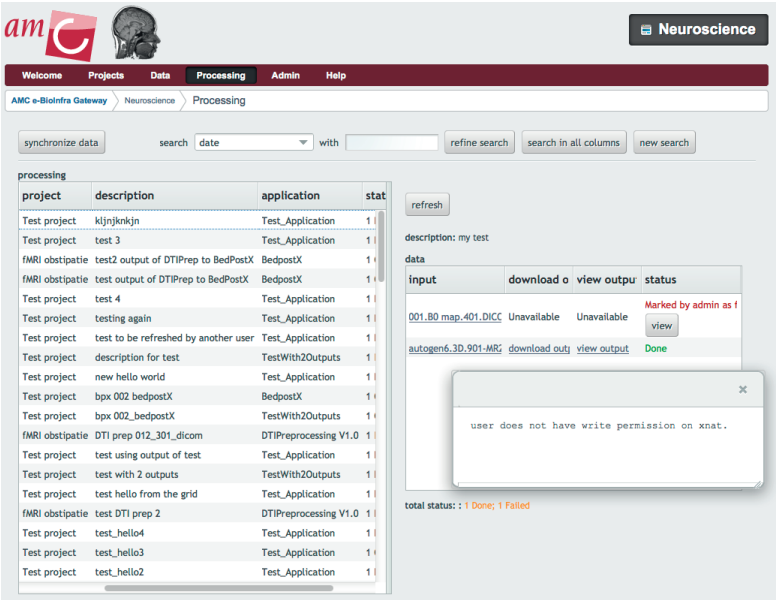


Figure 5.6: AMC-NSG eBrowser user view for processing monitoring: list of processings on the right; progress information about selected processing on the left; box shows error message sent by administrator.

data processing instances. The eBrowser interfaces are adapted based on the roles that are assigned to the user profile: neuroscientist, called user here, and administrators.

eBrowser essentially enables users to start, manage, and monitor data processing. Figures 5.5 and 5.6 depict these user interfaces. When a user selects one or many data items to process, the eBrowser only displays the applications that are compatible with the selected data to the user based on the metadata and application specifications.

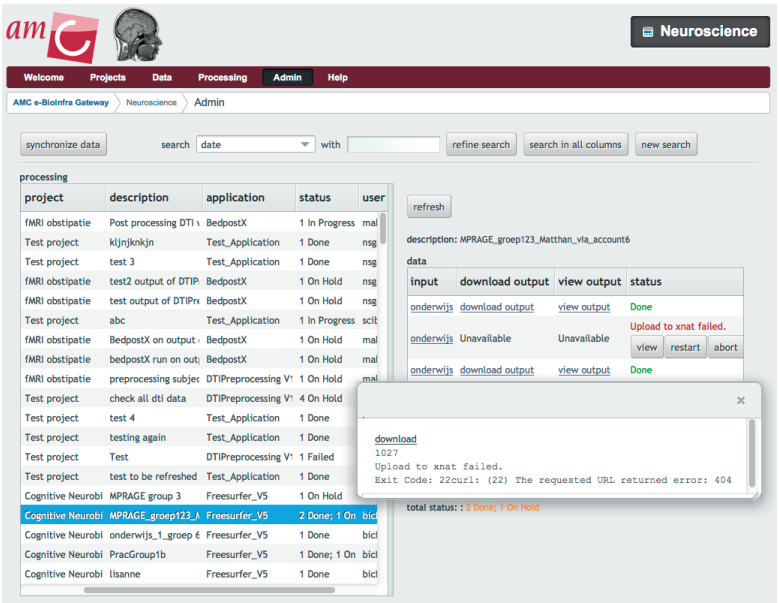


Figure 5.7: AMC-NSG Administrator processing view: all processings on the right; detailed information about selected processing and its Submissions on the left; box shows detailed error message.

The eBrowser also provides interfaces for system administrators. The administrator's dashboard displays monitoring information about all of the user data processing activities, and enables intervention on error. For example, in case of a failure during the execution of a workflow, a brief error message is displayed at the dashboard, and more details can be obtained by clicking the *View* button (see Figure 5.7). The administrator can choose to *Resume* or *Abort* the execution of the workflow, and to send a high-level message for the user (e.g., "The input file is corrupted.").

5.4.7 Component Interactions

Figure 5.8 illustrates the simplified use case of the science gateway, and depicts the functionalities and services provided by the gateway components. User actions are expressed via the eBrowser and trigger interactions between other high-level components (i.e., PM, DTS and eCAT) and lower-level components (i.e., gUSE and XNAT). *User A* is

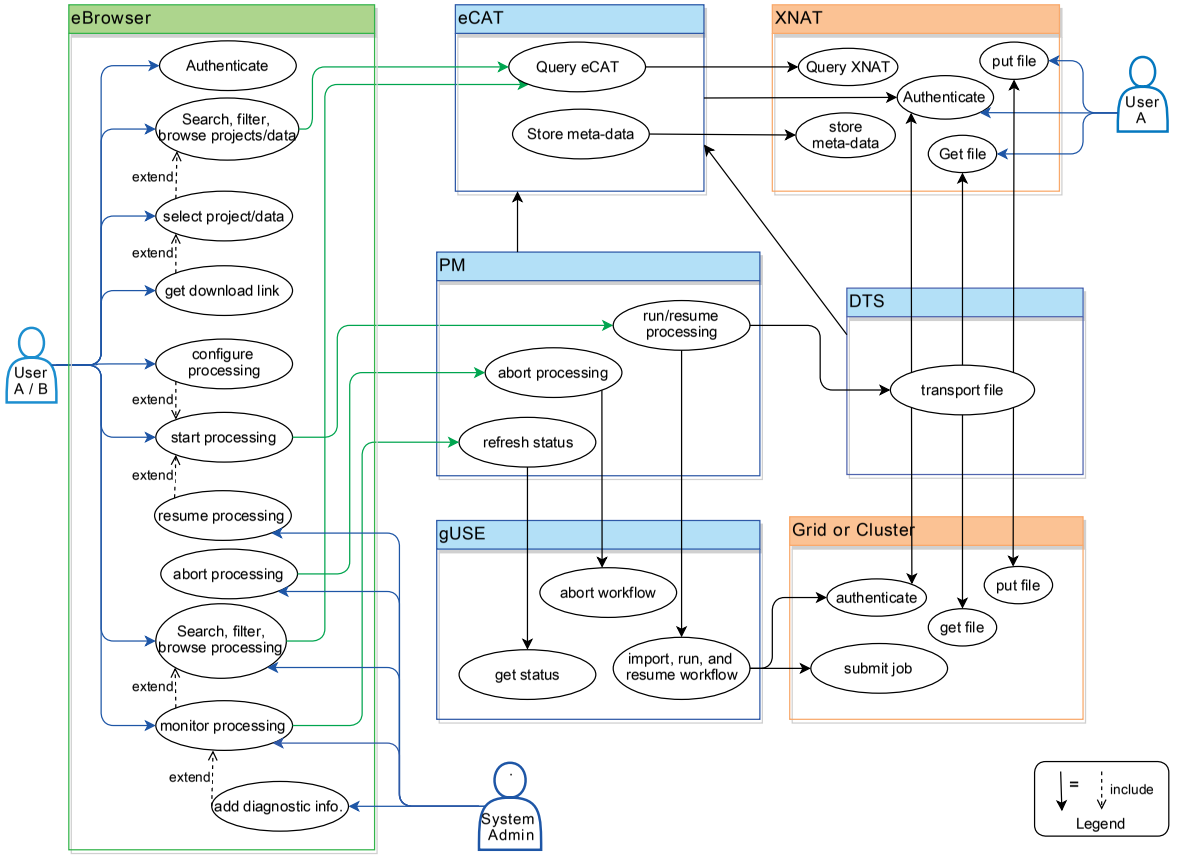


Figure 5.8: Simplified use case diagram illustrating the interactions between the users and the various components of the AMC-NSG. For simplification, a link that is attached to a component means that it is attached to all actions within that package. The color code is the same as it is used in Figure 5.2.

inside and *User B* is outside the AMC network. Details of these interactions are presented below.

Upon successful authentication with the gateway, the user gets access to the eBrowser portlet. New users need to configure an XNAT endpoint by providing their credentials. These configurations are collected by the eBrowser and sent to eCAT for validation and storage. After this configuration step, the following takes place when the user logs into the AMC-NSG.

At first the user sees a list of her/his projects. To display this list, eBrowser sends a request to eCAT, which authenticates on behalf of the user to the XNAT and generates a list of all projects that are accessible by that particular user. Similarly, when the user selects a project, the eBrowser sends a request to eCAT, which queries metadata on the XNAT to produce the list of all data entries in that project. Note that data and metadata should be inserted into the XNAT prior to its retrieval from the gateway.

The user then selects data entities for processing, and browses for available applications. The eBrowser retrieves and displays the list of applications that can be executed by the user, only showing applications that are compatible with the selected data type and format. The user selects an application and the eBrowser displays configurations (application parameters) for that application. The user configures the application and starts a new data processing. The eBrowser collects the provided configuration and submits a processing request to the PM. The PM consults eCAT to find the details of the selected application, namely the DCI to run it and the arguments that need to be configured for its execution (e.g., input files and parameters). The PM creates a new processing entity in eCAT, from which the eBrowser can later retrieve and display to the user for browsing, management, and monitoring purposes.

The PM further instructs the DTS to move the required input data to the target DCI. The DTS contacts eCAT to determine if those data already have a replica on the target DCI. If no replica is available, the eCAT provides DTS with the XNAT endpoint configurations (including authentication token) and location where it can retrieve the input data. The DTS then uses this information to authenticate on behalf of the user to the XNAT and download the input data to the gateway server. Similarly, it retrieves user authentication tokens for the target DCI to upload input data. Finally the DTS registers in eCAT the location of the file replica in the DCI and returns it to the PM.

After all data have been staged to the target DCI, the PM imports the application from gUSE via the ASM API, and configures it with the location of input data and user-specified parameters. Having everything in place, the PM starts the data processing by submitting the configured workflow to gUSE via the ASM API, and updates the processing status in eCAT. The gUSE Workflow Interpreter parses the workflow, generates corresponding jobs, and submits them to the DCI-BRIDGE. This service then retrieves user- or community-specific authentication tokens for the target DCI to submit jobs on behalf of the user to the target DCI.

The PM periodically updates the information in eCAT based on the status reports from gUSE. The user browses, manages, and monitors the processing via the eBrowser. eBrowser contacts eCAT to get information about processing entities, including status. Different levels of details (views) are shown to the user and to the system administrator.

Typically, each processing consists of multiple data to be processed. When the processing of a data item is finished, the result is immediately stored in the XNAT via

the DTS. Provenance data is associated to the results to identify the application that generated it and the input data. Thereby the user can check results even before the entire processing is complete. The links to the results on the XNAT are displayed by eBrowser.

5.4.8 The AMC computational NeuroScience Gateway

The AMC computational NeuroScience Gateway (AMC-NSG) is available on-line via http://neuro.ebioscience.amc.nl. The links to its source code, developer, and user documentations are also available on the SCIentific gateway Based User Support (SCI-BUS) portlet repository [6].

Currently, the following medical image processing applications are available to process Diffusion Tensor Imaging (DTI) and structural Magnetic Resonance Imaging (MRI) data: a) *Freesurfer*: implements segmentation of structural MRI data with the Freesurfer toolbox [48]; b) *DTI-preprocessing*: performs format conversion and quality control of DTI data [40]; and c) *BEDPOSTX*: performs local modeling of diffusion parameters with FMRIB Software Library (FSL) Bayesian Estimation of Diffusion Parameters Obtained using Sampling Techniques for modeling crossing fibers (BEDPOSTX) [50].

5.5 User Feedback

The new gateway has been only recently released, in November 2013, for widespread usage at the AMC; therefore, is has not been possible so far to carry out a significant user study. Nevertheless, in this section we describe our initial attempts to collect user feedback about the new gateway in various opportunities.

The new gateway has been first released for AMC users in July 2013. It has been thoroughly evaluated by two power users from the AMC radiology department for a few months (July-September), during which extensive feedback was provided and the necessary improvements were implemented in the system.

After that, the new gateway was used in the "Advanced Cognitive Neurobiology & Clinical Neurophysiology" course of the Biomedical Sciences Master program of the University of Amsterdam. After one hour of introduction about e-Science and the gateway, the students used the gateway during another hour to perform high level tasks such as "find the scan of your brain", then "run Freesurfer on it". During this course 17 students used the gateway simultaneously through 6 user accounts. All students were able to successfully complete the data analysis tasks.

After the course, the students were asked to answer a questionnaire as the first external users of the system, and to give feedback about their experience with the new gateway. They answered the following questions using five multiple choices ranging from *very negative* to *very positive*:

- Overall, how *satisfied* or dissatisfied are you with the Neuroscience Gateway?
- How likely are you to *recommend* the Neuroscience Gateway to a friend or colleague?
- How *capable* is the Neuroscience Gateway in supporting your needs?

- How *easy to use* do you consider the Neuroscience Gateway?
- How *visually appealing* or unappealing do you consider the Neuroscience Gateway?

Figure 5.9 summarizes the responses of the students in a radar chart. Although these responses can only serve as a very initial assessment, they show no extremes. Note that, although most of the students found the gateway not easy to use, almost all of them indicated that they are likely to recommend it to others. We recall that the students, who were absolute beginners in the topic and the usage of e-Science environments, were able to complete the assignments successfully. This indicates that the gateway is easy to use for management of computational neuroscience data analysis and hides the complexities of underlying framework from the end-users. Additionally, on average they were neutral about how satisfactory, capable, and visually appealing the gateway is.

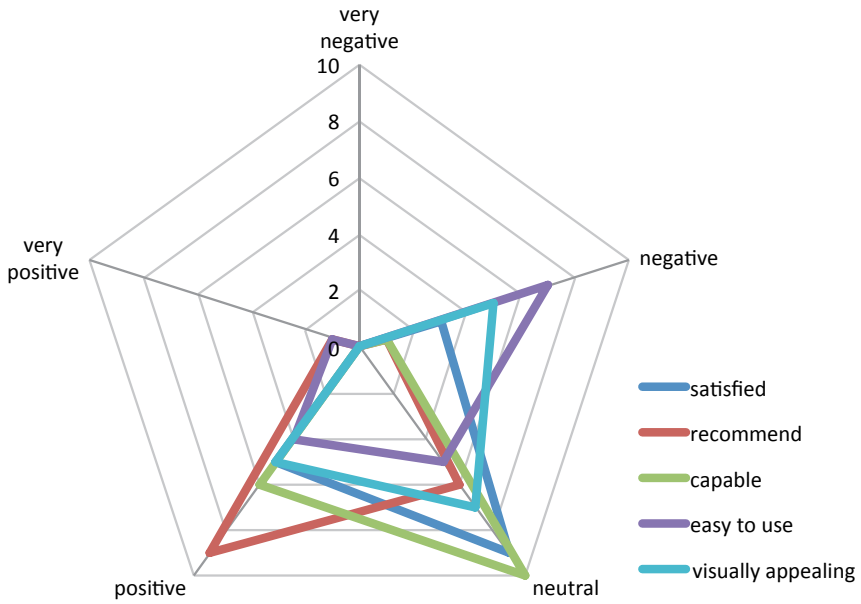


Figure 5.9: Summary of the feedback received from the students of the "Advanced Cognitive Neurobiology & Clinical Neurophysiology" course after one hour usage.

Finally, in November 2013 the new gateway was officially released to the Brain Imaging Center of the AMC. During that event the potential users pointed out that the interface was indeed looking better, but that hands-on experience with a real scientific task would be necessary for further feedback. Some of the users also pointed out the need to support other data sources, and to facilitate import of non-DICOM data into XNAT.

5.6 Related Work

Design, development, and usage of SGs have gained interest and attention in the past few years. Several projects and initiatives have been started worldwide to develop SG frameworks and SG instances for diverse user communities [80]. For example, see the list of SGs on the Websites of eXtreme Science and Engineering Discovery Environment (XSEDE) [165], EGI [44], and the SCI-BUS [157] project. In particular, several neuroscience research communities have developed various science gateways to integrate their medical imaging applications and data with access to computing and storage resources. This section presents some of the most recent science gateways for the neuroscience research and how they related to our work.

The neuGRID for you (N4U) SG [155] provides user-friendly access to a large number of tools, algorithms, pipelines, visualization toolkits, data, and resources on various DCIs (grid, cloud, and clusters) for medical imaging research. The goal is exploit these tools towards the cure of neurodegenerative diseases, in particular Alzheimer's, psychiatric, and white matter disease. The N4U Persistency Service registers distributed data from project partners into the N4U Information Base, which are then treated as a single data source.

The CBRAIN portal provides transparent access to remote resources to manage, share, process, and visualize imaging data [52]. The CBRAIN platform links several brain imaging centers to High Performance Computing (HPC) and cloud facilities across Canada and the world, both for data sharing and distributed processing. The data transfer and job submission details are transparently handled by the platform. Gee et al. [53] designed and implemented a data mining platform for neuroimaging data warehousing and processing that aim at brain recovery research. This platform integrates with CBRAIN for data processing and utilizes XNAT for data storage and sharing.

The Laboratory Of Neuro Imaging (LONI) pipeline environment facilitates the integration of disparate data, tools, and services in complex neuroimaging data processing workflows. It supports neuroscientists with visual tools for data management and integration, and workflow development and execution on HPC platforms. It also updates the data provenance automatically during the processing [38].

The Virtual Imaging Platform (VIP) portal [58] is a multi-modality medical image simulation platform that facilitates sharing of object models and medical image simulators. The models are described with semantic Web ontologies and shared in a common repository. VIP portal enables users to run simulations implemented as MOTEUR workflows on the EGI computing resources and in-house clusters. Data is uploaded via the portal Web interface using a dropbox-like approach; it is then stored on the EGI data resources, and indexed in a central logical file catalog.

The Charité Grid portal [177] enables its users to run medical imaging applications on the German grid resources provided by the MediGrid and PneumoGrid projects. It also provides an interface to a Picture Archiving and Communications System (PACS) that contains anonymized medical images. The users are also provided with interfaces to upload data to the portal server. High-level data services upload data from the PACS or the portal server to the grid computing resources for processing, and download the results to their desktop computer.

The Diagnostic Enhancement of Confidence by an International Distributed Envi-

ronment (DECIDE) SG [9] provides high-level services for computer-aided neurological diseases diagnosis and research on the European Research and Education Networks, and EGI. It is based on the Catania SG framework [145] and utilizes a data engine that enables data transfer and sharing on grid storage resources.

The Neuroscience Gateway [132] is based on the CyberInfrastructure for Phylogenetic RESearch (CIPRES) Science Gateway framework [102]. It enables the users to run parallel neuronal simulation tools on HPC platforms in the US cyberinfrastructure. It hides the technical details from the scientists for running jobs and managing data.

These science gateways are usually designed and implemented based on the requirements of the specific research community that they support. Therefore, each one is unique in its own way. However most of them display a few common characteristics that resemble our new gateway: data resources are directly connected to the gateway; a large variety of neuroimaging applications are available for the users; grid and clusters are used for high throughput computing. The major differences lie on the software platform used to implement the gateways, which varies from customized solutions to workflow management systems and SG frameworks. More information about the implementation would be necessary for a proper comparison of these systems; however, such information is normally not presented in the publications accessible to us, and, when they are, they may become obsolete very quickly.

5.7 Discussion

The new gateway is significantly different from the previous one. Table 5.1 highlights the main differences concerning their main features. Although some of the features are similar, in many cases they have been integrated. Moreover, their implementation are totally different in the two generations of e-BioInfra Gateway. The previous one was built based on the Spring framework, MOTEUR WfMS, and DIANE pilot job framework; it only supported the Dutch grid infrastructure; and it lacked facilities for data management, user interface customization or community support. In contrast, the new one is built based on the WS-PGRADE/gUSE SG framework, which itself is built on the Liferay portal framework. Liferay provides facilities for user management, community management, and community support (e.g., on-line forum). Moreover, it also facilitates the construction of customizable Web-based user interfaces that are required to suit needs of each user (community) based on their profile, expertise, and roles. The WS-PGRADE/gUSE SG framework provides high-level generic services to manage workflows, enact them to various DCIs, and monitor their execution. These services allow for functional scalability and interoperability between various DCIs. Additionally, the WS-PGRADE/gUSE framework is an actively maintained, supported and developed open-source project, which allows the development team of the AMC-NSG to concentrate on community-specific features, and makes the gateway maintenance more sustainable.

Currently only XNAT is supported as data server. Several other data management platform alternatives meet the research requirements, although XNAT is of special interest due to its support for medical imaging. More importantly, it is already adopted by the AMC neuroscience research community. XNAT has been particularly designed for

Table 5.1: Main differences between the previous and the new e-BioInfra Gateway (AMC-NSG).

Feature	Previous gateway	New gateway
Supported DCIs	Dutch grid	Dutch grid + in-house cluster
Supported external data servers	Absent	XNAT
Data transport	Semi-automatic & passive (GridSync)	Automatic & active
Search data based on metadata	Absent	Present
Collaboration between researchers	Absent	Data shared within XNAT Projects
Provenance	Only data history	Full support (data, parameters, etc.)
Recovery from failures	External to the gateway	Integrated into the gateway
Communication between users and system administrators	External to the gateway	Integrated into the gateway
Workflow details & error messages	Exposed to the users	Hidden from the users

managing standard medical imaging data as the core of its functionalities. In addition, its archiving and integrating capabilities, data model flexibility, ease of use and the highly active community of users/developers makes it a relevant asset. By connecting the XNAT to the AMC-NSG, the XNAT usage is also improved. Researchers are now able to perform compute-intensive data analysis on the data and receive the results in the same system. An alternative would be to develop XNAT pipelines to send processing jobs to external computing resources. Note however that the AMC-NSG has been designed to support multiple and heterogeneous data servers, and it is not dependent on XNAT.

In this implementation we chose to use an external data server, and keep the access control to this data server completely in the hands of the community administrators. This helped us build trust between the systems, which is a known critical factor to connect them to open infrastructures. Additional advantages of relying on external data servers for data management are flexibility, extensibility, data federation, and transfer of operational responsibilities to data owners. On the other hand, it is challenging because of issues such as connectivity, speed, and synchronization. For example, we experienced difficulties to connect to the XNAT server at the AMC because of the firewall policies. Also, the eCAT caches metadata to reduce the frequency of XNAT

queries; however, keeping the metadata on XNAT and eCAT in sync also turned out to be challenging. Finally, if the external data server is discontinued or is off-line for any reason, the links from the eCAT to XNAT and much metadata (e.g., provenance) present on the gateway become invalid.

We used WS-PGRADE/gUSE as SG framework, which in principle provides the workflow management and portal functionalities needed for our new gateway. After a learning phase, during which the concepts of the framework were better understood by the team, we observed that the usage model of the framework differs from our needs in some cases, which has led us to develop our own *processing manager* component. This has the goal of translating high-level "data processing" commands into low-level data transports calls to the *data transport service*, and to workflow execution calls to the gUSE ASM API. At first sight this introduces small overhead, but at the same time it provides sufficient isolation from aspects regarding this particular WfMS, and allows us to consider other WfMSs in the future. Moreover, this solution handles the data transfer and provenance collection properly, which would be more difficult to implement without this abstraction layer.

The development of eBrowser viewing portlets was also simplified by the decision to have all user interaction to take place using information available on the eCAT. This approach requires all software components to register all activity on the eCAT, but it decouples the viewer from all the other components accordingly. This reduces dependencies between the system components and simplifies its implementation and maintenance. Moreover, it makes eCAT a natural provenance data repository for the activity carried out at the gateway. The provenance is captured during the runtime and the information that is relevant for the research community is stored in their data server as metadata.

Among the motivations for the new gateway was the need to reduce the number of errors and also to handle them in a more elegant way. In the new gateway the data-related errors are not observed by the end-users anymore because the gateway prevents them from processing incompatible data type and formats with its applications. Additionally, in case of an error, the submission is put on hold and the administrators are notified about the error. This allows experts to inspect the error and act upon it (e.g., blacklist a faulty cluster on the grid and resume the submission) without involving the end-user unnecessarily. In the new gateway, the user is exposed to high-level user-friendly error messages only when there is some application or data-related problem that he/she can resolve.

5.8 Conclusions

In the new generation of the e-BioInfra Gateway (AMC-NSG) we tried to reduce the gap between users, data services, and DCIs. In contrast to our previous gateways, here we aimed for a data-centric gateway in which everything is organized around "data" and most importantly "metadata".

Now users can use the gateway to browse their data and metadata, which can be potentially stored on several data servers and described by rich metadata, and to

perform large scale data processing on them using DCIs. This can be done without getting involved into low-level details of the infrastructure.

By making the gateway metadata rich, the execution of applications is also streamlined. Because there are now more metadata available about the input data and the applications, it is possible to assist the user in choosing the correct application. For example, if a data item is not compatible with a specific application, the gateway prevents the user from starting a processing for this combination. Moreover, applications are not isolated from each other any more. The output of one application is transferred to the XNAT with proper metadata, which can be used to match this result to inputs of another application as a subsequent step in the data analysis pipeline. Although in some cases the steps can be linked at the workflow representation, in medical imaging it is still usual to visually check the results for quality control, which hampers full automation.

In the near future, the gateway will be disseminated in more training events, and become open to the whole neuroscience research community of the University of Amsterdam. This step will require inclusion of other data servers, for example other XNAT instances or even other systems, as well as extending the eCAT with federated services for accessing (and/or querying) multiple data servers. Increasing number of users and data will possibly require further development of instruments for strong community support and communication. Moreover, semantic content annotation (ontologies), as well as adding knowledge and integrating it with existing data, could enable further automation of the data processing to reduce even more human intervention in the analysis of large quantities of biomedical data.

Finally, we kept bioinformatics researchers in the loop during the requirement analysis, design, and implementation of the gateway. The goal was to assure a design that is generic enough to support this new community with minimal additional effort. Although in this chapter we focused on the computational neuroscience applications, the same concepts and software components are being used to develop a science gateway for protein docking.

Acknowledgments

We thank the colleagues who participated in the development, deployment and testing of the new e-BioInfra Gateway for computational neuroscience: Mark Santcroos, Hurng-Chun Lee, Luis Font, Paul F.C. Groot, and Gerbrand Spaans.

This work performed computations using resources of the Dutch e-Science Grid, which is financially supported by the Netherlands Organisation for Scientific Research (Dutch: Nederlandse Organisatie voor Wetenschappelijk Onderzoek) (NWO) and by Stichting SURF.

This work is financially supported by the COMMIT/ project "e-Biobanking with imaging for healthcare" funded by the NWO; the SCI-BUS project funded by European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 28348; the ER-Flow project funded by FP7 INFRASTRUCTURES-2012-1 call under contract no 312579; and the HPCN UvA project "Computational Neuroscience Gateway" funded by the University of Amsterdam.

PART III

Insights

CHAPTER 6

Reflections on Science Gateways Sustainability Through the Business Model Canvas

Case Study of a Neuroscience Gateway

In Concurrency and Computation: Practice and Experience, To Appear.

Shayan Shahand, Jeroen van Duffelen, Sílvia D. Olabarriaga

Abstract

The sustainability of science gateways has been a topic of active discussion, because they have been created and supported in the context of temporary research and infrastructure projects. As successful projects come to an end, it is necessary to find (new) models to secure continuous exploitation of products generated by these projects. Taking this step requires business considerations that are not trivial to do from the role of a researcher. This chapter presents our experiences in adopting a methodology from lean business development, the Business Model Canvas (BMC). This methodology enables structured reflection upon the business model, and facilitates exploring alternative ones (pivoting). We have applied the BMC to one of the science gateways designed, developed and operated by the AMC e-Science group: the AMC Computational Neuroscience Gateway. The current gateway BMC is explained in the chapter and used as basis for a reflection to improve its sustainability. Alternative business models are given as examples of BMC iteration or pivots. This exercise helped us to structure the various aspects to be considered when designing or reflecting upon the business model of our gateway. It also facilitated the visualization of the complete business picture and helped the reflection about improvements in the business model towards sustainability. We believe that this methodology could be valuable also for the reflection about sustainability of other science gateways that are growing from academic groups that do not have business training.

DOI: 10.1002/cpe.3524

6.1 Introduction

Science Gateways (SGs) are systems that enable scientists to manage their big data, harness the power of big computers, and collaborate with each other towards a certain scientific goal. SGs are usually accessible via Web interfaces that hide the complexities of the underlying IT infrastructure from the scientists, so that they can focus on answering their main scientific questions. At the e-Science group [138] of the Academic Medical Center (AMC) of the University of Amsterdam, we[1] have been designing, implementing and operating SGs for more than four years now [73, 129, 130]. During this period we have implemented three generations of SGs, and yet another generation is currently under development. These SGs have been used mainly by biomedical researchers for big biomedical data analysis, such as medical image processing, DNA sequence alignment, and protein-ligand docking simulations. Our SGs enabled researchers to perform their data processing fast and easily, for example in the studies reported in [87, 118, 121, 175].

6.1.1 The Problem

As a small research group, sustaining the operation of our SGs has always been an important topic of discussion and much reflection. Our SGs have been built with financial support mainly from various research projects. This means that, in most cases, the goal of these projects is not to operate SGs, but instead it is to research best practices, tools and techniques to design and implement SGs. Therefore, SGs are considered as by-products of our funded research, and it is not in the interest of this funding to support their operation for a long time after validation. On the other hand, the very fact that the products of our research are used in practice is an important indicator of success that we do not want to miss. There have been some small funds from our home institutes to sustain the operation of these SGs, and there is increasing awareness about their benefits and usefulness. More generally, there is also a growing trend in the funding bodies to request from research projects to define a "sustainability plan" or a "business model" for exploitation of products generated during the project after its termination. However, it is still not clear how to keep SGs in operation in the long term. We observed that this is not our exclusive problem.

6.1.2 Related Work

Several other communities also face such challenges and are looking for solutions to sustain future operation of SGs. One of the many examples of community efforts towards answering sustainability questions is the Science Gateway Sustainability Workshop, held jointly with the International Workshop on Science Gateways (IWSG) in 2013 [159]. The participation in the workshop acknowledged the importance of sustainability of SGs and their underlying services and technologies for a wide range of disciplines. Based on the workshop report [123], in summary the main findings of the workshop are: *a*) SG front-ends and user interfaces are important for their acceptance, usage, and success; *b*) SGs should support both compute-intensive and data-intensive tasks – activities

[1] Throughout this chapter, "we" refers to all or a subset of the AMC e-Science group members.

related to data such as reproducibility of publication results are often overlooked; *c*) sustainability of SGs remains challenging – it is specially important to cover operation and maintenance of infrastructures and services, and user support beyond the project-based funds; *d*) several alternatives such as vouchers, consumables, pay-per-use, and centralization, as well as indirect support from education and e-learning services were suggested as funding approaches; *e*) making a software open-source does not remove the costs required to further develop, maintain, and support it; *f*) company and spin-off business models still remain to be proven, especially with respect to funding and marketing strategies that are different from the academic side.

The on-going activities of the Science Gateways Institute (SGI) [158] are another example of effort toward sustainability of SGs. They published a report [174] about the characteristics of successful SG projects that are embraced by a scientific community and change the way they conduct science. This report is published based on the discussions carried out in several cross-disciplinary focus groups. The report highlights some of the challenges faced by SG stakeholders: *a*) academic funding and reward systems are not designed for production and operation of stable software; *b*) although development of SGs can be quite similar across different domains, typically it has been done in isolation by researchers who are unprepared for demands of such projects such as business and technology development, community building, and fund raising; *c*) SGs require national and international foundational building blocks that enable their designers and developers to focus on higher-level and grand-challenge functionalities; *d*) securing sustainable funding for SGs remains challenging especially at the operational or institutionalized infrastructure phases that come after they have graduated from the initial proof-of-concept demonstrators and developed into a stable product for deployment. For various stakeholders in the SG projects (i.e., managers, developers, community members, and funding organizations), they provide recommendations such as: *a*) designing governance models to satisfy specific needs of the project at different stages; *b*) planning for turnover in the future; *c*) recruiting a professional development team that understands both the domain-related and technical issues; *d*) determining strategies to pay for the project after the initial funding; *e*) measuring success early and often; *f*) leveraging the work of others; *g*) being flexible to technological, user, funding, or research changes and to adapt according to input and reactions from other stakeholders; *h*) identifying a specific community and a clear goal and *i*) engaging the community members throughout the project. Based on these recommendations, SGI envisioned to offer a range of services and resources to support SG development: *a*) SG incubator service that include development facilities, consulting (for business development, marketing, project management, etc.), hosting services, and software recommendations; *b*) SG support team that collaborates with research teams to transfer knowledge and help them build, enhance, operate, and maintain SGs; *c*) SG framework with a modular and layered architecture that allow SG developers to pick and choose the components they need; *d*) SG forum to gather people who are interested in SGs that provides networking, knowledge exchange, feedback gathering, and dissemination opportunities; *e*) SG workforce development to train professionals with dynamic and multi-disciplinary expertise that is required for SG design and development.

There are a few scientific communities who are already following some of these recommendations. For example, the VisIVO gateway [124] tries to achieve sustainability

by making the project open source and governing it transparently with the aim of attracting maximum contribution from general public and other scientific communities [163]. The Molecular Simulation Grid (MoSGrid) gateway [55] is another example of a SG that gained sustainability through the support and interest received from users. This has led to more funds from EU and US research agencies to continue and extend the SG.

On a more general scope, there is the "CyberInfrastructure Software Sustainability and Reusability report" [134] that describes the findings and recommendations based on workshop discussions to examine the general issue of software sustainability. Another example is the "Working towards Sustainable Software for Science Practice and Experiences" [176] workshop series that provides a forum to discuss and share best practices for sustainable scientific software.

6.1.3 Methodology for Approaching Sustainability

But.. what does all this sustainability and business talk mean in practice? As researchers, typically without business training, at the AMC we are facing the challenge of taking a step into the economics of SGs as a means to assure their continuation, exploitation, and further development. In response to this challenge, recently two members of the AMC e-Science group participated in a program offered by the Amsterdam Center for Entrepreneurship (ACE) to learn techniques required for building a start-up. A start-up is a company, a partnership or temporary organization designed to search for a repeatable and scalable business model. In lean business development, multiple business models are designed and tested by the start-up. A popular tool used for this purpose is the Business Model Canvas (BMC), which is a template that facilitates mapping, describing, designing, and inventing new business models. During this program we used the BMC to study the case of the AMC Computational Neuroscience Gateway, which is one of the various SGs we designed, implemented and operated for users of the AMC.

In this chapter we describe how we applied the BMC in the context of the Neuroscience gateway. In this exercise we considered the SG as a business which would be developed via a start-up. We described the current situation, and then designed a new business model that might be more sustainable. We use this as an illustration of how BMC can be used to organize and visualize business activities of a SG. Note that this tool is meant as an evolving picture of the AMC computational NeuroScience Gateway (AMC-NSG) seen from a business perspective, therefore it is expected that many of such BMC are drawn along time.

The chapter first briefly presents the BMC concept in the context of SGs. It then presents the case study of the AMC-NSG, and finally discusses the lessons learned in this exercise. We found that the BMC helps considering the various factors involved in the sustainability of the SGs in a structured way. Although in our exercise we considered only one of the SGs, for a particular discipline, we believe that many of our findings could be valuable for other cases.

6.2 Business Model Canvas for Science Gateways

"Lean start-up" is the new methodology for launching companies. It puts emphasis on searching for a repeatable and scalable business model through testing *hypotheses* and collecting early and frequent *customer feedback*, instead of executing finalized business plans and releasing fully functional products. In this methodology, the hypotheses are summarized in a framework called a *Business Model Canvas (BMC)* [115]. It is used in an approach called *customer development*, which emphasizes going out and asking potential customers and users for feedback on all elements of the business model. The emphasis is on agility: to quickly develop the *minimum viable product* and collect customer feedback, then use the feedback to revise the business hypotheses either by making small adjustments, a.k.a. *iterations*, or by more substantial ones, a.k.a. *pivots* [18].

The BMC consists of nine building blocks that describe value proposition, infrastructure, customers, and finances of a business. These blocks are arranged in a canvas in such a way to help organizing, aligning, and illustrating the business activities. Below these blocks are described briefly in the order in which they should be considered during business design and analysis – see more details in [23, 143].

Customer Segments. This block includes all the people and organizations for which the business creates value. Building an effective business model depends on the identification of the customers that the business tries to serve. There are various types of customer segments based on the characteristics of the market, for example, mass, niche, or multi-sided market. In the context of SGs typically one thinks about the end-users: the scientists who directly use the system in their research. However there could be other segments, such as project managers, infrastructure operators, or even developers of SG or new applications and services that are accessible via the SG. For example, when the SG also supports application development via scientific workflows, workflow developers could also be considered as customers. These target customers usually have diverse profiles and expertise, collaborating in the context of research communities. Other segments outside the research community could also be considered, for example, organizations that are interested in the results of research carried out via the SG. Note that if there are several customer segments, there can be multiple and different value propositions for each of them.

Value Propositions. This block describes the collection of products and services that create value for the customers. The value is provided through various elements such as performance, price, cost reduction, usability, customization, availability, innovation, mobility, etc. The rule of thumb is to answer the flowing questions in this block: *a*) Which problems does each customer segment face? *b*) Which services and products are offered to solve those problems? *c*) What does each customer segment gain by using the solution? In the context of SGs the most obvious values for the scientist could be to scale applications up or out to increase performance and reduce time to results; integrated access to data, applications, infrastructure, and collaboration tools; and higher usability and efficiency through customized and streamlined services. When considering a broader customer segment than the scientists themselves, other

values emerge, such as lower cost due to streamlined and coordinated infrastructure management, faster or easier development and maintenance of SG, more efficient operation of SG, etc.

Channels. This block describes how potential customers become aware of the Product and Services (P&S) of a business, how they evaluate value propositions of P&S, how they purchase P&S, how P&S and their value propositions are delivered to the customers, and how they receive after sales support. Effective channels distribute the business' value propositions fast, broadly, and efficiently. In the context of SGs, it usually means *a*) colleagues or community Websites as awareness and evaluation channels; *b*) Web or graphical user interfaces that can be accessed through the Internet or the institute's Intranet using various devices such as desktop, tablet, or smart phones as delivery channel; and *c*) emails, mailing lists, and forums as after sale channels.

Customer Relationships. These are the type of relationships established with the costumers, which are crucial to the survival and success of any business. Examples of various forms of customer relationships include: personal assistance, self-service, community support and training, and co-creation of product and services. In the context of SGs it could mainly mean supporting users (i.e., end-users, managers, developers, etc.) especially for training and troubleshooting. Additionally, it could include extending the functionalities of the system to accommodate new requirements coming from power users, the research communities or the infrastructure. This could be implemented as co-creation, where power users collaborate with the SG team for improvements.

Revenue Streams. This block is about the ways through which a business generates revenue from each customer segment. Examples of revenue streams include: usage fee, subscription fee, licensing, and advertising. SG revenue could be guaranteed individually (pay per use, per researcher) or collectively (organization pays a fee that allows a group of researchers to use the SG). Another possibility is to sell advertisement space on the SG Web portal. Note that revenue generation is a particularly challenging topic in the context of SGs, because research funding is usually the main source of revenue for the users of SGs, coming from projects at international, national, or even institutional organizations. These sources of funding are usually temporary and rarely include sufficient budget for information infrastructure, which is assumed to be provided by persistent organizations. The current trend to treat IT infrastructure as pay-per-use service might change the culture for research infrastructure management and facilitate the payment for SG services in the future.

Key Resources. These are the resources and infrastructure necessary to create and deliver value to the customers. These resources, which could be human, financial, physical, and intellectual, are indispensable assets that are required to sustain and support the business. In the context of SGs it usually includes: *a*) hardware such as computing and storage resources, which are typically distributed and shared among various organizations; *b*) software such as middleware, applications, and third-party

tools and frameworks; and *c*) expertise and intellectual properties of the SG designers, developers, and operators.

Key Activities. These are the most important activities that need to be performed to deliver value to the customers and to have the business perform well. In the context of SGs it usually includes: *a*) the software development activities such as requirement collection and analysis, system design, development, integration, test, and validation; *b*) system operations and maintenance activities; *c*) extending SG functionalities for example by porting new applications into the system; and *d*) other soft activities such as user support and training, marketing, business development, fund raising, and project management.

Key Partners. These are the external organizations that form a buyer-supplier relationship with the business in order to optimize business operations, reduce risks, and focus on the core activities. In the context of SGs it usually includes: *a*) international and national organizations, and institution's departments that manage and provide computing, data, or network infrastructures; *b*) third-party resource and software providers; and *c*) research communities that provide use-cases, requirements, data, specialized resources, etc.

Cost Structure. These are the costs while operating the business. Characteristics of cost structure include the type (fixed or variable costs), and the relationship between product or services and their scale or scope. In the context of SGs it usually includes the personnel's salary and costs of hardware and external software and services. It also could include costs with research and training for development of expertise, traveling costs for establishing new or keeping existing customer relationships, auditing and consulting costs for administrative or security purposes, etc.

6.3 Case Study: AMC Neuroscience Gateway

In this section first a brief overview of the AMC-NSG [129] is presented, then its current business model is described using the BMC. The current situation is analyzed, and finally possible directions to pivot our business model to make it more sustainable in the future are explored.

6.3.1 Overview of the AMC-NSG

This section aims solely to set the background for the business case study – more technical details about the AMC-NSG can be found in [129]. The AMC-NSG provides large-scale data processing services for a few applications for structural Magnetic Resonance Imaging (MRI) and Diffusion Tensor Imaging (DTI) scans. The brain scans are stored at a given image data server based on the eXtensible Neuroimaging Archive Toolkit (XNAT) technology [96]. The following applications provided by third-parties are currently integrated: Freesurfer [48], FSL BEDPOSTX [50] and DTI pre-processing [39]. Users, typically neuroscientists, select the data and the application from a Web interface, and

the AMC-NSG autonomously manages the data transfers and computation on the Dutch grid infrastructure [162]. The users don't see the grid infrastructure at all; we quote how one of our users sees the AMC-NSG: *"...an incredibly user friendly interface to perform the most difficult analyses with several mouse clicks..."* [70]. The AMC-NSG currently uses the Web Service – Parallel Grid Run-time and Application Development Environment / grid and cloud User Support Environment (WS-PGRADE/gUSE) [77] to manage the computation, Liferay [153] for the Web portal, and the Lightweight Middleware for Grid Computing (gLite) [149] as middleware to the grid infrastructure, which is provided by SURFsara [162]. A number of additional high-level services and customized Web-based user interfaces were designed and implemented in-house to address requirements of neuroscientists from the AMC.

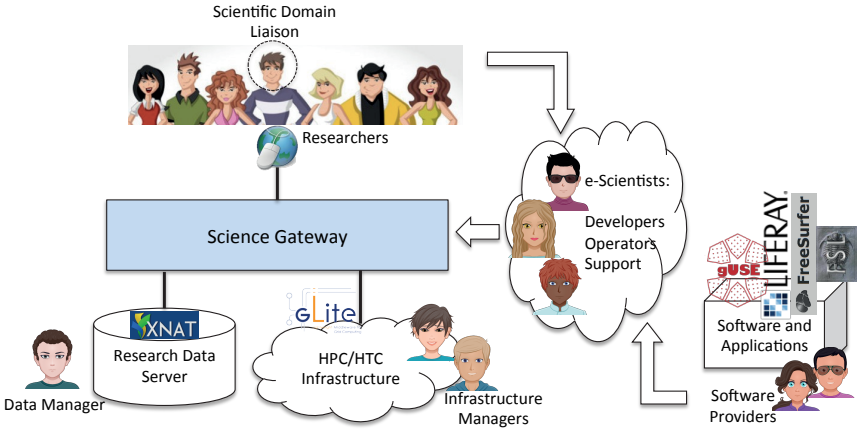


Figure 6.1: AMC-NSG ecosystem showing most relevant actors and services: users, developers, operators, external resources (data, computing, middleware and applications), and collaborators.

Figure 6.1 summarizes the most relevant components of the complex ecosystem behind the AMC-NSG. E-Science developers extract requirements from the community, typically via liaisons or power users, and utilize and customize available software and applications provided by third-parties to address the community requirements and realize the SG. The SG also exploits High Performance Computing (HPC) and Hight Throughput Computing (HTC) infrastructure and research data servers for computation and data storage respectively. This is done via middleware services and APIs, which in turn might require development of high-level services. Finally a customized Web-based user interface is developed according to the concepts, desired functionality, and vocabulary of the target community. After deployment, the e-Science operators make sure that the system is up and running. User support is also offered, in particular initial training and troubleshooting. At this stage small interventions are required to accommodate applications, data and computing infrastructure evolution. Similarly, problems arise during the operation that require investigation and debugging by the support team. E-scientists maintain contact with all involved parties, i.e., the scientific

community, software providers, and infrastructure and data managers, to keep the SG operational and facilitate its usage.

6.3.2 Current Business Model

Figure 6.2 Illustrates the current business model of the AMC-NSG in a BMC. Below we present more details and discuss the AMC-NSG BMC blocks.

Customer Segments. Currently the AMC-NSG supports the research activities of the AMC Brain Imaging Center (BIC) [137] and is only available to its members under the "terms and conditions" that are available on the gateway. BIC currently consists of roughly 50 researchers with varied expertise, background, and profiles who are involved in computational neuroscience research projects. The group contains eight principal investigators, four postdocs, 25 PhD students, 11 master students (interns), and two technicians. PhD and master students typically perform data processing and analysis, whereas principal investigators and postdocs have a supervising role. These projects typically include the following phases [128]: study design, data acquisition, data preparation, data processing, data analysis, publication and data archival. The AMC-NSG covers the data preparation (partially), data processing, and data archival phases.

Value Propositions. Computational neuroscience researchers require big processing power to process medical imaging data with image processing applications. They can utilize grid infrastructure, which is free for academic research projects, however the researchers need to go through the paperwork to get access to these resources. Moreover, they need to have technical knowledge or training to be able to harness the full power of these resources. The AMC-NSG addresses these challenges by facilitating large-scale data processing for these researchers without the necessity to go through the technical training and paperwork. It offers: *a*) a set of popular and predefined neuroimaging applications executable via a few mouse clicks; *b*) a customized Web-based and easy-to-use Graphical User Interface (GUI) designed to address the requirements and desired functionality of BIC community members and projects; *c*) streamlined data management and processing by the integration of (in-house) data servers and remote computing resources; and *d*) scalable and transparent data processing on the Dutch grid infrastructure. These services enable the AMC-NSG users to perform large data processing in less time (i.e., from months to days) with minimal effort (i.e., a few mouse clicks to select the data and start the computation), enabling researchers to scale out their experiments and improve their efficiency. The AMC-NSG can also be used in teaching and training activities offered by BIC members. Additionally, the AMC-NSG removes the burden and costs of systems utilization, integration, and operation from the shoulders of the researchers.

Channels. Currently the main channel to reach the new customers (awareness) is through the BIC Website and word of mouth (e.g., presentations, meetings, courses).

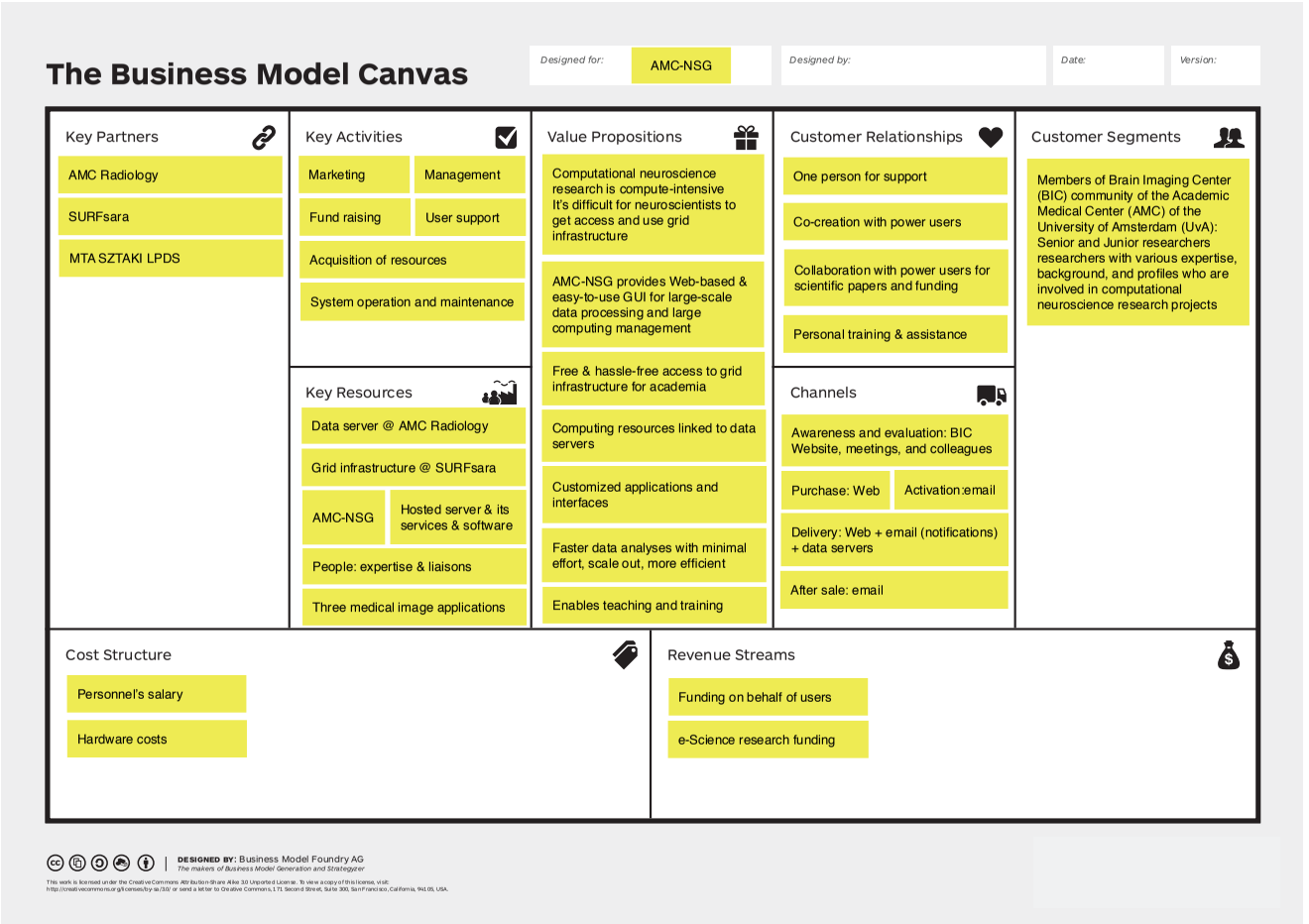


Figure 6.2: The Business Model Canvas applied to the AMC computational NeuroScience Gateway (AMC-NSG).

Researchers consider using the AMC-NSG when they want to run the offered applications on large datasets. Their initial evaluation, which takes them to decide upon using the AMC-NSG or not, is typically based on the experience of others, or on a face-to-face meeting with the support team. The services are delivered through Web-based interfaces and email. Account registration (free purchase) and activation are done via email communication with the support team. The input and output data are delivered via the in-house data servers. Additionally, email notifications are used to inform the users about the latest system events, for example, the status of their running data processing. Finally, individual email and mailing lists are used to provide after-sale support.

Customer Relationships. One person is dedicated for activities related to customer relationships, such as user assistance, support, and personal training. The development team keeps a close contact with the BIC community members through "power users" or liaisons to develop and operate the AMC-NSG based on their (new) requirements. In addition to co-creation, the power users are also involved in co-publication of scientific papers and submission of proposals to receive funding to sustain the AMC-NSG activities.

Revenue Streams. Currently the AMC-NSG service is available free of charge to the members of BIC. As an e-Science research group we raise funds for researching systems that enable data- and compute-intensive research projects. To report the impact of our research, and also to be able to raise more funds, we need to report our publications and their citations. Therefore, the users reward us by citing the publication about the gateway that they have used. If major design and implementations are needed to address the requirements of a project or community, we set-up a collaboration project to customize the gateway and implement new features. Results of such projects are usually worthwhile to share with the scientific community via additional publications. Additionally, on behalf of users, we receive allocated resources and small funds from our home institute to operate the AMC-NSG and support its users. In summary, we received the following funds: *a*) from a national project to conduct research about the general design and functionalities of SGs [146], *b*) from a European project to develop, deploy, and operate a domain-specific SG based on the WS-PGRADE/gUSE generic technology [157], *c*) from a European project to develop and integrate applications as scientific workflows [148], and *d*) from our home institute to continue operation and user support beyond the lifetime of mentioned projects. The acquisition of this funding so far has been the initiative of the e-Science group.

Key Resources. The followings are considered as key resources because they are indispensable assets required to sustain and support the AMC-NSG: *a*) the computing and data storage resources of the Dutch grid infrastructure and their corresponding middleware services, which are used to process and manage large amounts of data; *b*) the in-house data server and its middleware service to which researchers upload inputs and from which they download the processing results; *c*) the AMC-NSG software (including the WS-PGRADE/gUSE framework) that integrates the computing and data storage resources; *d*) the in-house expertise and intellectual property related to the distributed

systems design and integration; *e*) the community power users or liaisons who provide feedback in order to keep the AMC-NSG relevant for the research community; and *f*) the three medical image applications used for data processing, which are developed by (external) specialized research groups and integrated into the AMC-NSG as scientific workflows. Note that replaceable resources such as hosted servers and non-specialized software such as database or mailing list management systems are considered as commodities.

Key Activities. Before launching the AMC-NSG, the main activities are: requirement collection and analysis, system design, implementation, integration, application porting, and testing. However, after the launch, the focus is on user training and support and system operation and maintenance (e.g., upgrade software and configurations, integrate new services, communicate with resource providers). Marketing activities are also important to attract new users. In order to keep the AMC-NSG operational, activities to raise funds (e.g., writing proposals) or business development and to acquire computational and data storage resources are also necessary. Finally, all of these activities require management to stay relevant and also to report on usage statistics, etc.

Key Partners. The data server is run by the AMC Radiology Department. The computing and data storage resources are provided by the Dutch e-Science research infrastructure (SURFsara [162]). The AMC-NSG is built using the WS-PGRADE/gUSE technology that is provided by the Laboratory of Parallel and Distributed Systems of the Computer and Automation Research Institute of the Hungarian Academy of Sciences (MTA SZTAKI LPDS). This software provider is considered as a key partner because they provide support for the WS-PGRADE/gUSE, maintain its code, and add new features based on new services and requirements.

Cost Structure. Costs mainly include personnel's salary for the key activities and hardware costs for operations and user support. In addition to salary costs, there are also costs for training of personnel and for participation in (inter)national events.

6.3.3 Reflections about the Current Business Model

The BMC enabled us to better understand the large amount of aspects involved in our SGs, as well as the immense possibilities and challenges regarding business models. To support further reflection we collected data about the AMC-NSG usage, as a way to confirm our original value proposition.

The current version of the AMC-NSG was launched in Nov 2013, however at that time the data server was not yet populated, therefore no data analysis was possible. The first usage for data processing by a neuroscientist took place in January 2014, and since then the number of users and activity increased significantly. Figure 6.3 illustrates the number of submitted jobs corresponding to execution of one of the three available applications on one dataset. In summary, 13 neuroscientists submitted around 2,500 jobs via AMC-NSG, which consumed about 90,000 hours (10 years) of CPU time. At the

time of writing this chapter, most of the neuroscientists are in the process of analyzing and publishing their findings. This was obtained without the researchers spending time on requests to use grid resources, learning how to use the grid infrastructure, or payment. These data confirm our value proposition of providing free and hassle-free data processing services for the AMC BIC researchers.

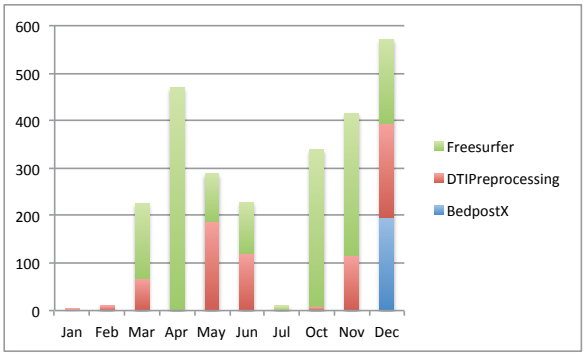


Figure 6.3: Number of jobs submitted by the AMC-NSG users in 2014 for the three available applications. Stacked columns; empty months are removed.

Another finding after applying the BMC is the detection of an imbalance between the customer segment and the revenue streams. The AMC-NSG users in general are not in a position to directly generate revenue to our business. This means that the value proposition needs to be expressed not only at the individual gateway users, but also at a higher level of management. So far our focus has been on the requirements for the end user, and based on the BMC we realize that we should direct our attention to the management layer as well. Or alternatively, we could look into multi-sided markets by searching revenue from organizations instead of from the end users. In the canvas we could also see that key partners can become essential to provide other channels that could facilitate generating revenue (e.g., the ICT department of an organization could be interested in adding the SGs to their services, and selling these services for external parties).

Another finding is that there are a lot of dependencies on the key partners, for example the Dutch e-Science research infrastructure. This infrastructure is currently sustained free of charge for Dutch researchers, however coordinating activities with them requires additional effort on our side. Moreover these partners provide resources only available for the academic research-oriented SG, therefore we must seek other partners should we decide to diverge the business model into industrial applications.

Finally, we also realized that the amount of key resources and key activities, although not all of them are listed here, are rather large, which also involves large costs. One possible solution to sustainability is to look into the *economies of scale*, which is to obtain cost advantages through increasing user base to an optimum amount that will decrease cost per unit of output. Another solution is to search for additional revenue streams, other than those generated by direct users. For example, we foresee that SG innovation will continue to depend primarily on research funding. We believe that these findings and reflections could apply to and be valuable for other SGs and communities as well.

6.3.4 A Business Model Iteration

To demonstrate how an *iteration* on a business model works, a possible iteration on the AMC-NSG BMC is explained here, where the customer segment is extended with an additional segment from the same community. Only the BMC blocks that are changed with respect to Section 6.3.2 are explained.

Customer Segment. Principal Investigators (PIs) of the BIC community research groups are added as an additional customer segment. They usually steer several research projects, for which senior and junior researchers (postdoctoral researchers and PhD students) are hired to do tasks such as data acquisition and processing.

Value Propositions. PIs need to monitor the actual progress of each project and intervene when necessary, which is time consuming to do manually. Additionally, they need to reuse and merge data of several projects when possible. Moreover, the project members are typically hired by short-term contract, which implies when their internal knowledge about the project is lost when they leave. However, because these projects are already using the AMC-NSG, a dashboard can provide to the PI an overview about the progress of all running projects. PIs can dive into details of each project when necessary to guide their teams. This saves time and effort of managers and team members. Additionally, when a project ends the actions taken by its team to process the data are archived with the input data and results, which can be reused in other (future) projects.

By attracting this customer segment to our business the goal is to offer services to the higher level of management which will have the means to pay or to decide about payment for gateway services in the future.

6.3.5 A Business Model Pivot

To demonstrate how a *pivot* on a business model works, a possible pivot on the AMC-NSG BMC is explained here, which explores a multi-sided market where the revenue comes from an organization instead of end-users. Currently the AMC-NSG is not visible to the grant organizations, which fund the neuroscience research projects directly.

Customer Segment. There are two customer segments: neuroscientists, such as the in original BMC presented in Section 6.3.2, and grant organizations that support and fund neuroscience research projects.

For the neuroscientists, the value propositions, channels, and customer relationships remain the same as in the original BMC presented in Section 6.3.2. Thus, the changes in these BMC blocks are explained below with respect to the grant organizations.

Value Propositions. Currently researchers spend much effort to organize data management and processing for large studies, which limits the research achievement that can be obtained in the scope of the funded projects. It is in the interest of the grant organizations to maximize research achievements. Moreover, there are duplicated efforts and expenses across different projects. Finally, grant organizations have difficulty to enforce open and sustainable research data practices. SG services could be hired by the grant organization to address these challenges, and these services would then become available without payment for the researchers that receive grants from the organization. On the one hand, grant organizations can spend funds in an effective and efficient way, by concentrating expenses on research support services that can be shared by many projects. On the other hand, applicants are able to focus on their own research question, with access to adequate data processing services. Additionally, grant organizations can promote and enforce sustainable data management policies for open reproducible research by using our services, as well as guarantee sustainability of these services beyond the scope of the individual projects.

Channels. To make grant organizations aware of our services we require to get their attention in relevant conferences and events, and through our Website, emails, and personal contacts. We expect that word of mouth also plays an important role to raise awareness. After the awareness, building interest and delivering these services are done via meetings and emails. Note that in this pivot the grant organizations become a channel to reach new customers among the research applicants.

Revenue Streams. Continuous funding from the grant organization is provided to keep the systems operational, in addition to small funds for project-specific activities, for example, to develop or customize SG for a new project.

Key Resources, Activities, and Partners. Except for the following changes, the content of these blocks mostly remain the same as the original BMC: *a*) In key resources, additional data servers and computing resources are used by the different projects. Also, various other (legacy) applications are deployed in the SG. *b*) In key activities, additional software design and development is needed to address the requirements of specific projects; Moreover, fund raising activities are replaced by business development activities towards the grant organizations. *c*) In key partners, the project-based resource providers also become key partners.

6.4 Discussion and Conclusions

Since our training, we already applied the method various times, exploring alternative business models for our gateway. Because BMC allows illustrating various aspects of a business model on one page, it facilitates comparing various business models and inventing new ones. We found the application of the BMC challenging at first, even after our training. For example depending on the focus of the business, training can be seen both as a customer relationship, value proposition, channel, or key activity. Based on our experience, it requires a couple of iterations before one feels comfortable

with the BMC layout and vocabularies and to start benefiting from it for the business model analysis. Moreover, there are alternative methods to the BMC, for example, lean canvas [170], and BMC for User eXperience (UX) [24], which we have not considered yet. Lastly, this method is extensively applied in business modeling today, therefore we are aware that what we are doing is not new.

In this chapter we detailed the current business model of the AMC-NSG operation, and we speculated about a possible iteration and a pivot. It is our intent to continue to explore several business model alternatives using BMC in the near future, to find means to sustain the operation of our science gateways in the long term. Similarly to many other e-Science research groups, we have put a lot of effort into designing and implementing various science gateways, and logically we would like to maximize their utilization. This requires going to the field to test the business hypothesis. We acknowledge to have been too focused so far on the technical aspects of science gateways, whereas we should be looking at the two most important parts of the BMC, namely customer segments and value proposition. In fact without correct identification of customer segments and clear value propositions towards those customers, the business itself and the rest of the BMC blocks would be irrelevant. The clear organization and spacial distribution of the blocks on the canvas help to visualize the whole, whereas showing details of all relevant relationships. This structure helps deriving the complete picture of such a complex case, identifying hypotheses, and focusing on the parts that matter most.

Acknowledgments

We thank Prof. Dr. Antoine H.C. van Kampen and the anonymous reviewers for their critical review of this paper, as well as the inspiration and support provided by the current and past members of the AMC e-Science group, the AMC Bioinformatics laboratory, and the Amsterdam Center for Entrepreneurship. This publication was supported by the Dutch national program COMMIT/.

CHAPTER 7

Science Gateway Canvas
A Business Reference Model for Science Gateways

In Proceedings of the 1st Workshop on The Science of Cyberinfrastructure: Research, Experience, Applications and Models, SCREAM'15, pages 45–52, Portland, OR, USA, 2015.

Shayan Shahand, Antoine H. C. van Kampen,
Sílvia D. Olabarriaga

Abstract

Science Gateways (SGs) have emerged as systems that facilitate access to cyberinfrastructures. There is a growing interest in the exploitation and development of SGs. However, it remains challenging to understand and design SG with the required properties because of the complex nature of SGs. Additionally, it is difficult to decide upon frameworks that can be used to build them. In this chapter we propose the Science Gateway Canvas, a business reference model for SGs that embodies the common SG functions and their organization into groups and categories. We used the Science Gateway Canvas for a systematic analysis and comparison of the functions offered by a selection of available frameworks that are used to build SGs. This illustrated the applicability of the Science Gateway Canvas as a comprehensive and generic reference model for understanding SGs. The canvas can also be used for systematic analysis by scientists who are searching for an existing SG that supports their research goal or developers who want to determine the functional requirements for a new SG.

Copyright Information
S. Shahand, A. H. C. van Kampen, and S. D. Olabarriaga. "Science Gateway Canvas: A business reference model for Science Gateways". In Proceedings of the 1st Workshop on the Science of Cyberinfrastructure: Research, Experience, Applications and Models, SCREAM'15, pages 45–52, Portland, OR, USA, 2015.
SCREAM'15, June 16, 2015, Portland, Oregon, USA.

DOI: 10.1145/2753524.2753527

7.1 Introduction

Advanced distributed computing and data infrastructures, a.k.a. cyberinfrastructures or e-Infrastructures, are necessary to enable the new paradigm of data-driven scientific research which, regardless of domain, increasingly require management and processing of large volumes of complex data [66]. Moreover, scientific research is not a solo person's activity anymore; rather, a group of researchers with various expertises and profiles collaborate with each other towards a common goal. Because it is not possible to collect and process such data on a single computer, and researchers are dispersed between organizations, cyberinfrastructures are therefore essential. However, researchers often do not have the advanced technical knowledge that is required to fully exploit these cyberinfrastructures. *Science Gateways (SGs)* have emerged to address this challenge. SGs are Web-based systems that provide researchers with customized and easy access to community-specific data and tools on distributed computing and data infrastructures, as well as to collaboration tools for bringing in their expertise towards a common scientific goal. SGs hide the complexities of underlying infrastructures from the users and enable them to manage their data, tools, computations, and to collaborate with each other more easily.

As more researchers find out about SGs and become interested in developing their own, they need to face the complexity of SG construction. SGs are complex to design, build, and operate because they need to address a large number of often contradicting user requirements, as well as to integrate distributed computing and data infrastructures that are utilized through various middleware and operated by different organizations. Fortunately there are many technologies and concepts that can be (re)used to build SGs more effectively. For a newcomer, however, it is difficult to locate, understand and put all these alternatives into perspective in the context of SG functions.

In this chapter we propose the *Science Gateway Canvas (SGC)*, a business reference model for SGs that embodies the common SG functions and their organization into groups and categories. A *business reference model* is a reference model concentrating on the functional and organizational aspects of the core business of an enterprise, service organization, or government agency. A *reference model* can be constructed in layers and serve as a reference for various purposes, offering a foundation for the analysis of service components, technology, data, and performance [46]. In this chapter we take the analogy between SGs and enterprises and propose the SGC to express function groups that are offered by a SG. We then provide an overview of a selection of available frameworks (i.e., technologies, tools, approaches) for building SGs, discussing how they fit into the SGC. This exercise helps the reader better understand both the SGC and the functionality offered by available solutions in a comparative fashion. The SGC provides a comprehensive and generic reference model that can be applied to any SG regardless of its domain or implementation details. It facilitates understanding and comparison of both the capabilities of SGs and the tools that can be used to build them.

7.2 Science Gateway Canvas

The Science Gateway Canvas is a business reference model that embodies the common SG functions (see Figure 7.1). It groups SG functions based on their realm (domain of activity) into nine function groups: data, computing, and community management, coordination, security, monitoring, and provenance functions, and delivery functions for humans and other programs. These groups are further classified into three categories based on their theme, namely, resource management, universal, and delivery functions. Note that these function groups should not be confused with an architectural view of software components and services. Whereas here the focus is on the functions that are provided to the users and other programs by the (collection of) SG software components and services.

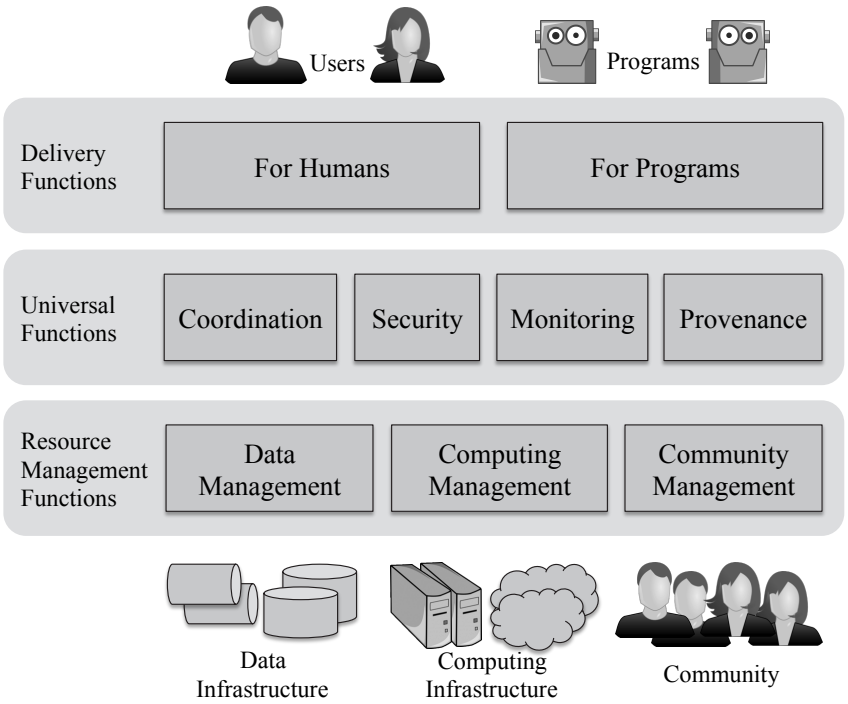


Figure 7.1: Science Gateway Canvas. SG functions are grouped and illustrated with boxes based on their realm, which are further classified into three categories: namely, resource management, universal, and delivery functions. These functions collectively provide users and other programs (illustrated at the top) with high-level and easy access to data and computing infrastructures, and community expertise (illustrated at the bottom).

7.2.1 Resource Management Functions

Data, computers, and communities are the entities that come together at SGs. It is common that they exist independently from SGs, and they can also be accessed through other means than SGs. SGs wrap and integrating these individual resources and services into a higher-level set of functions as described below.

Data Management Functions that support data-related tasks such as acquisition, storage, retrieval, transport, organization, replication, curation, integration, and aggregation of data and metadata. A SG typically provides functions to manage the data, hiding the details of the physical storage resources from the user. It provides transparent access to data and metadata, which can be distributed in various physical locations, stored with different protocols and formats, or integrated from various sources. It also automatically handles data transport and replication when it is necessary.

Computing Management Functions that schedule, execute, and manage computer programs executed on a pool of computing resources to perform tasks such as data analysis, simulation, or visualization. These resources can be homogeneous or heterogeneous, or they can be centralized in one location or geographically distributed. The computing management functions may implement standards or use abstraction layers to achieve transparent, scalable and interoperable computing management to run the applications that are relevant for the user.

Community Management Functions that manage the groups, relations, and communications between the people who collaborate through the SG. Note this not only includes the researchers who perform some scientific activity through the SG, but also those who provide technical support, for example to operate and maintain a SG.

7.2.2 Universal Functions

In addition to resource management, SGs also provide functions to manage the complex relations and interplay among data, computing, and communities. Universal functions utilize resource management to provide higher-level functions as described below. Function groups in this category typically can make use of other groups.

Coordination Functions that integrate and coordinate data, computing, and community management to deliver higher-level and more complex functions such as instantiating and setting up computer programs from application repositories, transferring data between distinct data resources or to the computing resources, starting and managing the execution order of computer programs, transferring outputs between computer programs and storing the results, and involving users whenever it is required (e.g., human-in-the-loop). The coordination steps need to be carried out such that they can be correctly executed, saved, repeated, reused and revisited. Such functions are typically provided by scientific Workflow Management System (WfMS).

Security Functions that provide authentication, authorization, and accounting. The information provided by the community management functions about persons, groups, roles, etc., is used here together with their credentials to enforce authenticated and authorized access to data, applications, computers, communities, information, and SG functions. Typically a large number and variety of credentials need to be managed by the SG. Security functions usually include Single Sign-On (SSO) or federated identity management, as well as transparent translation of credentials throughout the resources and systems integrated into the SG.

Monitoring Functions that collect and store the status of resources and different parts of the system, keeping track of the events and actions performed via the SG by users and automated processes. The monitoring functions need means to communicate with other SG functions to exchange information about the ongoing activities and their latest status throughout the SG.

Provenance Functions that collect and provide lineage information about the actions performed by the SG and its users. There are various methods to collect provenance information, such as mining structured and annotated databases, parsing log files, and utilizing the monitoring functions at runtime. This information is usually stored as metadata that is attached to the data items handled by the SG for later reference. Provenance functions enable to establish trust in the scientific experiments conducted via the SG, to troubleshoot applications, and to reproduce, interpret, and audit results.

7.2.3 Delivery Functions

To deliver the SG functions to the users, an additional set of functions are required to capture user inputs, and present the system status and information to the users. Additionally, SGs deliver their functions in a way that can be utilized by other programs too, for example, to interact with other SG and enterprise systems.

For Humans Users typically interact with the SGs through Web-based Graphical User Interfaces (GUIs) to perform scientific tasks and to interpret information in support to these tasks. These interfaces are customized to the scientific case of a particular community, for example, using more friendly terminology or streamlining workflows. Users search and filter their data collections; start, control and monitor activities on the SG; manage data and applications collaboratively; provide inputs to the SG; and observe the effects of these manipulations through GUIs. Visualizations such as rendered images, diagrams, and animations can be used to help the users interpret SG-generated information, such as the provenance or monitoring data, or domain-specific information generated by the applications. The style of interaction, however, varies significantly according to scientific area and user profile addressed by the SG. Whereas some users might prefer a simplified interface to execute predefined functions, others prefer Command-Line Interfaces (CLIs) that offer more control and from which it is easier to automate repetitive user tasks.

For Programs External programs can utilize the functions provided by the SG through Application Programming Interfaces (APIs). Well-defined APIs, with robust specifications and clear documentations, provide an abstraction layer to the API consumer. Moreover, APIs pave the road towards Service Oriented Architectures (SOAs). Abstraction layers and SOAs enable outsourcing parts of the system to experts who often have more control over the respective infrastructure, which will result in a more scalable and extensible system. Additionally, CLIs can also be used in scripts and programs to automate tasks, integrate with other systems, and provide higher-level functions.

7.3 Existing SG Frameworks

In the last ten years, the design, development, deployment, and operation of SGs was the topic of several special issues in international journals, namely the Springer "Journal of Grid Computing" and the Wiley "Concurrency and Computation: Practice and Experience" journal. The theme also gathered researchers in several international workshops, namely the "International Workshop on Science Gateways (IWSG)" and the "Gateway Computing Environments (GCE)" series. We have collected titles and abstract of publications from these special issues and events and also performed a search on the Inspec database through the Ovid search engine with ``science gateway*''.mp. as the search term. The joint collections included 319 papers excluding duplicates. Then we read those titles and abstracts and selected papers that explicitly referred to technologies that can be used to build SGs. This selection included 51 papers. Because SG frameworks usually evolve rapidly to accommodate changes in the infrastructure, and also to address the new or changing user requirements, we have limited our study to the papers published after 2011. We also referred to online material to update the information especially for older papers. For example, we have excluded Vine toolkit [41] because at the time of writing its website was expired, which means it is not likely to be an active project. In the end we selected 11 frameworks that can be used to build SGs for consideration in our study.

Below we present the selected SG frameworks in alphabetical order, using the SGC as a means to characterize the functions offered by each one of them. We start with a brief description of the system and then elaborate on the SGC function groups. Details about the technologies used to provide the functions can be found in references contained in the papers cited here. Whenever a SG function group is not mentioned, it means that we were unable to find anything about that particular function group in the respective publications.

Apache Airavata Apache Airavata, the reincarnation of the popular Open Gateway Computing Environment (OGCE), provides APIs to manage application and workflow metadata, and to distribute execution on a range of computing infrastructures [119]. Data management functions are provided by the Airavata Registry, which is the repository of all gateway metadata, including program and host descriptions, workflow templates and instances, etc. Computing management and monitoring functions are provided by the Application Factory (GFAC), which manages the submission and monitoring of programs to the computing resources. GFAC supports Basic Execution Service

(BES) and Job Submission Description Language (JSDL) standards and provides plugins for systems such as Hadoop, Amazon Elastic Compute Cloud (EC2), local, and Secure Shell (SSH). Applications that are submitted to GFAC are either individual programs or workflows that are enacted by the Workflow Interpreter. The Workflow Interpreter also supports iterative loops, conditionals, and human-in-the-loop executions, and keeps track of input data and configurations, generated outputs, and execution logs (provenance functions). The Orchestrator provides an abstract scheduling layer for individual programs or workflows. Therefore coordination functions are provided by both the Workflow Interpreter and the Orchestrator. The Credential Store manages different credentials associated with different cyberinfrastructures, providing security and community management functions. Apache Airavata functions are delivered only through an API.

Catania SG framework The Catania SG framework is built within the Liferay portal framework and enables its users to manage their data and computing on various distributed infrastructures [9]. Data management functions are provided by the gLibrary (using Grid File Transfer Protocol (GridFTP), LCG File Catalog (LFC), etc.) and ARDA Metadata Catalogue (AMGA), collectively known as the SAGA-based Data Engine. Computing management functions are provided by the SAGA-based Job Engine. The framework Grid Engine, which includes the Data Engine and the Job Engine, coordinates and monitors the program executions on a variety of cyberinfrastructures such as Lightweight Middleware for Grid Computing (gLite) and Uniform Interface to Computing Resources (UNICORE) (coordination functions). It also tracks and monitors user activities for audit and accounting purposes (monitoring and security functions). Liferay provides advanced community management functions such as user groups and roles, wikis and blogs. These features together with the role-based authorization features of the integrated Lightweight Directory Access Protocol (LDAP), creates a rich set of functions for community management. The Catania SG is compliant with the Security Assertion Markup Language (SAML) standard and uses Shibboleth to provide SSO and federated identity functions. Additionally, the integration with the MyProxy server, provision of community grid certificates (a.k.a. robot certificates), and support for Virtual Organization Membership Service (VOMS) (all provided by the eTokenServer), create a feature-rich set of security functions. In addition to the GUI that can be used by scientists, most of the Catania SG functions are also delivered via a RESTful API.

DARE The Distributed Application Runtime Environment (DARE) provides a framework for computing and data management using a pilot-based abstraction layer based on the Simple API for Grid Applications (SAGA) API [92]. Data and computing management, monitoring and security functions are provided by the SAGA-based components that can utilize various data and computing infrastructures. Coordination functions are realized by the pilot-based abstraction layer, in particular, the BigJob and BigData managers and agents. DARE functions are delivered to scientists via a GUI based on Pylons web application framework, and CLIs for advanced users or scripting.

Globus [Online] Globus, formerly known as Globus Online, provides software-as-a-service (GUIs) and APIs to platform-as-a-service for data, identity, and group management on several cyberinfrastructures [7]. Data management functions for high-performance, reliable, and third-party transfers are provided by Globus Transfer. Community management functions for user profile and group management are provided by Globus Nexus. Third-party file transfers fall into the coordination function group. Security functions for identity provisioning and federation, SSO, and data sharing are provided by both Globus Nexus and Globus Transfer. Monitoring functions are also available for checking the progress of data transfers. Globus functions are delivered through Web-based GUIs, desktop tools, CLIs, and APIs.

HUBzero + Pegasus HUBzero provides a platform to scientific communities to collaborate and share information, and to develop and run simulation and modeling tools on cyberinfrastructures for education and research purposes [100]. Pegasus is a WfMS that maps abstract workflows descriptions onto cyberinfrastructures [33]. As described in [101], the integration of these two platforms provides a Web-based venue for building, sharing, and delivering tools for education and research, and for executing them on a number of computing infrastructures. Data management functions are provided by HTCondor I/O using protocols and systems such as GridFTP, Amazon Simple Storage Service (S3), and Storage Resource Management (SRM). Computing management functions are provided by HTCondor, which submits computing jobs to systems such as HTCondor-G, Load Leveler, Load Sharing Facility (LSF), Oracle / Sun Grid Engine (OGE), Portable Batch System (PBS), and Simple Linux Utility for Resource Management (SLURM). Community management such as group management, blogs, and wikis are provided by the HUBzero platform. Coordination is provided by the Pegasus WfMS, in particular Pegasus Planner and Condor DAGMan. Monitoring and security functions, such as shared credentials and applications, are also provided. Provenance functions to keep track of what has been performed including the location of used and produced data, and software and its parameters are provided by Pegasus. HUBzero functions are provided via GUIs and CLIs, and Pegasus functions are delivered through CLIs and APIs.

ICAT Job Portal ICAT Job Protal provides a generic and configurable computing and data management portal that also offers CLIs and APIs as delivery functions [49]. Data management functions are provided by the data catalog that enables the users to store and query information about data files. For computing management, the users can submit batch or interactive computing jobs to Terascale Open-Source Resource and Queue Manager (TORQUE). Flexible rule-based system and community accounts are provided for community management and security functions. System and job monitoring functions are also provided. Provenance information is captured to be able to trace back the results through the chain of applications and intermediate data to the original input data.

InSilicoLab InSilicoLab provides a framework for building SGs that enable the users to utilize advanced computing infrastructures in a way that resembles using a personal

computer [83]. Functions to manage data on LFC resources together with storage and data structures, metadata models, and annotation and tagging, constitute the data management functions. Computing management functions are provided by an Execution Engine that utilizes Distributed Infrastructure with Remote Agent Control (DIRAC) and gLite frameworks. Community management functions such as user and group management, wiki, and blogs are provided by the Liferay portal framework. The Worker Management provides the coordination functions it utilizes a master-worker approach (a.k.a. pilot job) to schedule and coordinate computing jobs, together with the Parallelization and Experiment management components. Security functions for management of grid certificate and generation of grid proxies, and monitoring and provenance functions are also provided. InSilicoLab offers its functions through both GUIs and APIs.

iPlant iPlant is a SG for plant sciences community, however it is designed in a way that could be easily used for another scientific domains [90]. Data management functions such as data transfer and metadata management are provided by integrating integrated Rule-Oriented Data System (iRODS) and supporting various protocols such as Web Distributed Authoring and Versioning (WebDAV), GridFTP, and HyperText Transfer Protocol (HTTP). Computing management functions are realized by integrating HTCondor and Foundation API. Group management is provided as community management functions. Coordination functions are provided by a Job Execution Framework (JEX) that converts abstract execution descriptions into DAG files, and the HTCondor system that executes the DAG files. Shibboleth based SSO functions and support for SAML makes the security functions of the iPlant framework quite rich. An Object State Management system (OSM) provides the monitoring functions with a publish-subscribe approach. The iPlant SG functions can be utilized through a Web-based window-oriented environment (eyeOS) and RESTful APIs.

NEWT The NEWT platform provides a framework for creating RESTful APIs for High Performance Computing (HPC) [31]. Data management functions such as uploading and downloading files to the framework, directory listing, and database manipulation are provided by a service that integrates local file system and databases, and remote GridFTP. Computing management functions are provided by integrating several job managers such as TORQUE/Moab and Univa Grid Engine (UGE). More infrastructures could be supported by implementing adapters. Group management functions are also provided for community management. Security functions are built upon the Django authentication module, and integrate site-specific authentication schemes such as OAuth, Shibboleth, LDAP, and basic HTTP authentication. Moreover, MyProxy is used internally to generate short-lived grid certificates on behalf of the user. Monitoring functions provided are computing job status, accounting information, and resource availability check.

SINAPAD SG The SINAPAD SG provides a gateway engine that can be configured by a set of eXtensible Markup Language (XML) files to offers data and computing management on the Brazilian national high-performance computing network (SINAPAD) [61].

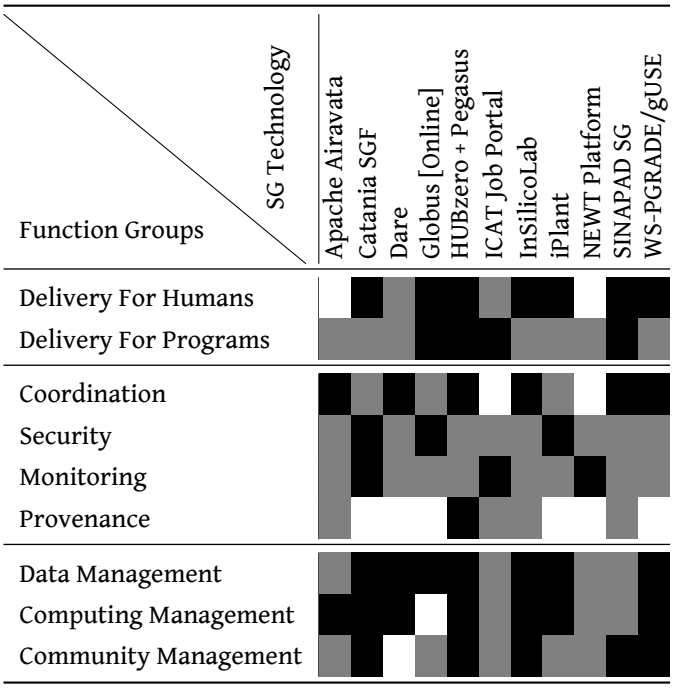
Data management functions are provided by integrating the CSGrid framework, which is based on CSBase middleware and its file system called CSFS, with the ProjectService component, which offers functions to upload, access, and manipulate files. Computing management and monitoring functions such as submission, monitoring, and control of computing jobs are also provided by integrating the CSGrid framework, which uses SLURM, PBS, and OGE, into the SG by a service called OpenDreams. Community management functions are provided by CSBase via the Project Area and Open Access Area, in addition to the Liferay portal framework. Coordination functions are provided by the Open Scientific Connectors (OSC) workflow description language and a service called GWO that maps the workflows to the computing jobs. Security functions are provided by a LDAP-based system that also allows restricted anonymous access to computing resources. Basic provenance information such as computing job and file history, and the versions of computer programs is also collected by the system. The SINAPAD SG functions are provided through GUIs, which can be portlets that are automatically generated by a system called PortEngin, as well as CLIs and APIs.

WS-PGRADE/gUSE Web Service – Parallel Grid Run-time and Application Development Environment / grid and cloud User Support Environment (WS-PGRADE/gUSE) provides a workflow-oriented GUI and APIs to create and execute workflows and manage data on various computing infrastructures [77]. Data management functions are provided to manage databases, local files, and remote files on gLite resources (LFC). Computing management functions are provided by the DCI-BRIDGE, a job submission service based on the BES standard, which is able to utilize various resources such as gLite, Globus Toolkit (GT) two and four, Advanced Resource Connector (ARC), UNICORE, local, PBS, LSF, Web services, Berkeley Open Infrastructure for Network Computing (BOINC) and Google App Engine. Community management functions are provided by group and communication management functions (e.g., wiki, blog) of Liferay. Coordination functions are provided by the gUSE workflow interpreter that uses its own XML-based workflow language and enables multi-DCI workflow execution with parameter sweep and embedded workflow support. Users can develop and store workflows in an application repository via the gateway graphical interfaces. These functions can also be invoked via the Application Specific Module and the Remote APIs to build customized SGs [76]. Support for Public Key Infrastructure (PKI) and community-specific grid certificates (robot certificates), and MyProxy integration are provided as security functions. Monitoring functions are also provided by the DCI-BRIDGE to check job status.

7.3.1 Summary

Table 7.1 presents a qualitative overview of the functions provided by the SG frameworks selected for this study. In order to derive this qualitative overview, we have compared the available functions for each function group and labeled each framework roughly based on its richness. The following subjective labels were used: advanced set of functions are present, some functions are present and could not determine presence of functions from the studied publications.

Table 7.1: Qualitative overview of the functions provided by selected SG technologies: Black = advanced set of functions is present; Gray = some functions are present; and White = we were unable to determine presence from the publications.



7.4 Related Work

Various works have recognized and addressed the need for guidance through the understanding and comparison of both the capabilities of SGs and the tools that can be used to build them. However, all of these efforts approach the question from a technical perspective whereas in this chapter we tried to look at it from a functional point of view.

Soddemann [133] presents requirements and technologies for SGs for Material Sciences research. It discusses three- versus four-tier patterns for SG system architecture, and reviews a few tools and frameworks that can be used to combine Web and grids. Gesing et al. [56] present and compare SG functionalities for Life Sciences applications. The categories used for comparison cover a subset of the function groups of our reference model, mostly focusing on management of grid computation, data and security. These two early works differ from ours because they do not structure the comparison categories into groups; moreover, they do not mention community management and provenance functions. Balasko et al. [11] considered a few SG frameworks and compared them based on a taxonomy that they defined. Their taxonomy consists of three aspects: sharing (hardware or software), execution management (simple jobs or workflows), and interfaces (existing, automatic, or fine-

tuned), which also covers a subset of the function groups of our reference model, mostly focusing on computation management and interfaces.

The European Grid Infrastructure (EGI) SG Primer [43] resulted from a workforce organized by EGI to discuss technical aspects and best practices to be considered during the SG design and implementation of SGs[1]. Typical functionalities of the SGs are categorized into processing management, data management, security, community support, monitoring & reporting, and visualization. A list of SG qualities is provided to help assessment and selection from existing SGs and SGs frameworks. Four different perspectives are contemplated in the EGI Primer, namely SG developers, SG operators, application developers, and end-users. Additionally, some of the most prominent European SG frameworks are compared based on their functionality.

Similarly to the EGI Primer, the eXtreme Science and Engineering Discovery Environment (XSEDE) SGs [125] website also provides a set of technical and organizational guidelines and best-practices for the principal investigators and developers who wish to build and operate science gateways on the XSEDE resources. It contains practical information, for example a list of technologies that can be used to build SGs [35]. Recently an extensive survey carried out by the Science Gateway Institute collected a large amount of information that helps understand SGs more deeply. For example, this survey revealed the background of developers and the profile of tools used to build SGs [89].

Marru et al. [97] present the approach used to develop the XSEDE SGs cookbook [179]. In that work there was no ambition to provide a generic framework, as the authors recognize that "digesting the myriad of technical details into a common cohesive form is as seemingly insurmountable challenge". Instead, the authors distilled a set of "recipes" about how to address specific topics concerning SG construction. The themes of recipes are closely related to the functional groups of the SGC, however the themes are not structured into groups and in some cases they overlap in function.

To our best knowledge, none of these works (EGI Primer and XSEDE Cookbook) has attempted to provide a comprehensive reference model as the SGC proposed here, which is generic and can be applied to any SG framework regardless of its domain or implementation details.

7.5 Discussion

We have proposed the SGC, a business reference model that embodies the common functions of SGs and their organization. The SGC was used to review and analyze the functions that are provided by existing frameworks that can be used to build SGs. During this exercise, all the SG features that we came across in the publications could be categorized into one of the SG function groups in the SGC, which indicates that the proposed reference model is sufficiently complete. This reference model made it feasible to compare and summarize the functions of a large and varied set of frameworks, as illustrated in Table 7.1. From this study we also have identified that some functions are better represented than others. For example, computing, data management and

[1] At the time of writing, the EGI SG Primer has not been published officially yet. However, its draft is available on the EGI website.

security functions are very well represented, whereas provenance functions are barely present. Depending on the requirements of the SG it might be that not all of the functions are essential. However, provenance will play a more prominent role with the growing demand for reproducibility and integrity in scientific research.

In spite of the usefulness of understanding the SGs and frameworks from a functional perspective, in this study we also noticed that this understanding is still disconnected from their implementations. Note that the SG function groups of the SGC should not be confused with software components or services. In reality, when a SG is implemented, a software component can deliver functions from two or more function groups, or similarly, a number of software components can deliver functions of a single SG function group. For example, in Apache Airavata [119] coordination functions are provided by two components, the Workflow Interpreter and the Orchestrator. Similarly, the computing management and monitoring functions are provided by a single component (OpenDreams) in the SINAPAD SG [61]. And in [101] coordination and provenance functions are provided by the Pegasus WfMS. Putting all these SG implementation variations into perspective, in addition to their function, would be also very useful for newcomers SG development.

Note also that, by design, the abstraction level chosen for the SGC hides the technical details of the SG implementations. On the one hand, it gives the advantage of making it feasible to analyze and compare such varied set of SGs and frameworks. On the other hand, certain technical details, especially those that have no effect on the SG functions, are not reflected in the SGC, and therefore have not been revealed by our study. For example, interesting technical solutions for the communication between the system components found in the Apache Airavata [119] and the SINAPAD SG [61] were not highlighted in our study.

Finally, the frameworks analyzed in this chapter have been selected from a collection of publications that are related to the SG research community, or that have explicitly identified themselves as related to SGs. In fact there are many more technologies that exist on a broader scope than SGs, or that do not identify themselves in the scope of SGs, or that have not been described in indexed scientific papers. These have been missed in our study altogether, but which are in fact also used for SG construction and could be worth examining. Examples are WfMS, Pilot job frameworks, social networking, data management systems, and various security solutions. Also, we noticed during this study that the description of the frameworks varies a lot according to the focus of the authors in the selected papers. This raises the risk of inconsistency in our results, because the absence of reference to a particular function group in the publication does not necessarily imply its absence in the actual SG framework. A broader study addressing more technologies and other sources of information (e.g., interviews with the SG developers) would be necessary to provide a more complete and refined overview of existing frameworks for SG construction.

7.6 Conclusions

In this chapter we have presented the Science Gateway Canvas (SGC), a business reference model that embodies the common functions of SGs and their organization.

It categorizes three classes (resource management, universal and delivery) of nine function groups (data, computing and community management; coordination, security, monitoring and provenance; and delivery functions for humans or other programs). This reference model is novel because it approaches the landscape of science gateways from a higher level of abstraction (functional capabilities), in contrast to architectural views that are more tight to technologies. This reference model is particularly useful to understand and compare both the capabilities of SGs, and the tools that can be used to build them, in a systematic manner.

To illustrate its concepts and give examples of its application, the SGC has been used in this chapter to review, analyze, compare and summarize the functions that are provided by existing frameworks that can be used to build SGs. Eleven SG construction frameworks have been reviewed, and a simplified overview has been generated (Table 7.1). From this overview, it is easy to detect that some functions groups (data and computation management; monitoring, security) are more developed than others (e.g., provenance). This might indicate the need for further development of SGs and the supporting technologies to address under-represented function groups.

To the best of our knowledge, the SGC is the first attempt to provide a comprehensive and generic reference model of SGs that can be applied to any SG framework regardless of its domain or implementation details. The abstraction level chosen for the SGC hides the technical details of the SG implementations, which makes it feasible to analyze and compare such varied set of SGs. This has been also illustrated by the systematic study conducted here for the frameworks that can be used to build SGs. This study can be further extended to cover a broader set of frameworks in the future.

Finally, note that the SGC can also be used for other purposes. Scientists who are searching for an existing SG for a certain domain or application can use the SGC to analyze and compare the available options. Moreover the SGC can also be used by SG developers to analyze and define the functional requirements for a new SG.

Acknowledgments

We thank the current and past members of the Academic Medical Center (AMC) e-Science group, and the AMC Bioinformatics laboratory for the inspirations that were sparked in various conversations. This publication was supported by the Dutch national program COMMIT/.

CHAPTER 8

Discussion and Future Research

The research presented in this thesis advanced our understanding of the fundamentals of Science Gateways (SGs) for biomedical research. This understanding is important because it promotes cross-fertilization, facilitates design, development, and operation of new SGs, and, most importantly, it guides future research on SGs. These fundamentals were derived from an iterative study of concrete cases of specific biomedical research communities following a user-driven and bottom-up approach. This study resulted in the design and development of four SG generations. The first and second generations were prototypes and exploratory, which were evaluated with a small number of users in scientific projects and courses [25]. The third and the fourth generations, which are detailed in this thesis (Chapters 4 and 5), were deployed and used in biomedical research projects. These SG generations were studied in partnership with the computational neuroscience, omics, and medical chemistry research communities from the Academic Medical Center (AMC) of the University of Amsterdam as shown in Table 8.1. This chapter reflects on the main findings of this study with respect to the research questions posed in this thesis (Chapter 1) and delineates possible future research.

Table 8.1: Overview of the biomedical research communities involved in the four SG generations.

	Biomedical Research Communities
1st	Computational Neuroscience
2nd	Computational Neuroscience
3rd	Computational Neuroscience • Omics
4th	Computational Neuroscience • Medical Chemistry

8.1 Requirements of Science Gateways for Biomedical Research

In this section we discuss the answers to the first research question: "*What are the requirements of biomedical researchers to efficiently use e-Infrastructures?*"

During the life cycle of a biomedical research project, several researchers collaborate with a wide spectrum of complementary background and expertise, such as medicine, biology, statistics, data processing, and information technology. These collaborators take various roles along the project life cycle, such as data collector, data analyst, and principal investigator. Moreover, researchers perform several tasks in different phases of the research life cycle based on their roles. Finally, on top of all that, some characteristics may vary in each discipline, or depending on local culture and research project setup. All these characteristics translate to an overwhelming and complex set of requirements, with two consequences. Firstly, strong partnership with the specific research communities is needed to fully understand and describe all the requirements for an effective SG. Secondly, it is unlikely that a single SG will be able to address all these requirements, leading to the need to build various SGs for specific research communities.

In spite of all this diversity, in our research we identified three main groups of functional requirements at the core of any SG for biomedical research, namely related to data, computing, and collaboration:

- *Data*-related requirements concern management of complex, distributed, and heterogeneous datasets on e-Infrastructures. Examples of data-related requirements are storing, annotating, searching, retrieving, replicating, and archiving datasets, as well as capturing metadata and provenance.

- *Computing*-related requirements concern management of complex, computationally demanding, distributed, and coordinated data processing on e-Infrastructures. Examples of such requirements include negotiating resources, porting data processing methods to e-Infrastructures, as well as running computations and handling errors.

- *Collaboration*-related requirements concern management of communications and interactions among scientists involved in a research project. Such collaborations entail, among others, exchanging information with reference to data and computing, sharing and reusing data and processing methods, and defining collaboration teams and their members.

Note that these requirements may change rapidly in such a fast evolving scientific field as biomedical research. With the advent of new measurement technologies, the volume, complexity, or types of datasets may grow and introduce new data- and computing-related requirements. Or the number of participants in a project can grow as new (inter)national consortia are formed. Therefore, it is also crucial to study and understand the dimensions along which data-, computing-, and collaboration-related requirements can evolve along time.

We have observed that it is easier to discover and see through the complex set of requirements of the biomedical research communities in the context of SGs when

organizing them explicitly around data, computing, and collaboration requirements. It was only when we deployed the third SG generation, which addressed the most prominent computing-related requirements, that the scientists and ourselves understood that there was a second dimension to consider, namely the data-related requirements. Likewise, when the data-related requirements were better addressed in the fourth generation, the need for more sophisticated collaboration mechanisms became evident, motivating for a future fifth SG generation. Our experience illustrates that discovering the requirements and potentials of SGs is a learning process for both the SG researchers and biomedical scientists. Similar learning process is also observed in other communities. For example, the Molecular Simulation Grid (MoSGrid), for molecular simulation in quantum chemistry, departed from offering computing capacity [169] and later also addressed data management [86]. Discovering requirements along these three functional requirement groups (data, computing, and collaboration) from the beginning may help to shorten this progressive refinement process. Note that these three groups seem sufficient for the biomedical research requirements, and in fact we could observe the same groups in other disciplines. However, one could argue whether these are the only ones. For example, do instrumentation-related requirements, such as management and control of data acquisition devices, fit under the data-related requirements or should they be categorized into a different group? Or can the organizational aspect of research life cycle, such as recruiting participants, be captured by the collaboration-related requirements, or should it also be considered as a separate group? Further research is necessary to answer these questions.

On top of the three functional requirement groups, our research also revealed that there are additional non-functional requirements, such as usability, scalability, flexibility, and efficiency. Addressing all these requirements simultaneously is challenging because they may contradict with each other. For example, a SG can streamline the actions and concepts for a scientific community to increase usability, which implies sacrificing flexibility because it will likely constrain the usage to these very actions and concepts. We observed that in many cases the scientists work with well-established research methods, which are possible to streamline and encapsulate into automatic processes offered by SGs. This was the case for the computational neuroscience and medical chemistry communities that were targeted by the fourth SG generation, which explains why they were successfully adopted by so many users. However, we also observed that there are scientists that require more flexibility to fine-tune or define their own scientific processes. For example bioinformaticians dealing with omics data typically define their own data analysis pipelines, which explains their lack of interest in using the third SG generation. These scientists need programmable and flexible systems that normally require more training and technical expertise. Scientific Workflow Management Systems (WfMSs) potentially offer such functionality, however, the design and development of such systems is a topic of research in itself. In this thesis we therefore concentrated on SGs that enable research for the first type of research methods.

8.2 Design, Development, and Operation of Science Gateways for Biomedical Research

In this section we discuss the answers to the second research question: "*How to build SGs that address these requirements? What are the design, development, and operation considerations?*"

The main finding regarding SG design is that all of the three functional requirement groups, namely, related to data, computing, and collaboration, should be considered in the design. The SG should integrate data, computing, and collaboration resources seamlessly. Failing to integrate any of these resources will result in a SG that is not effective and will find limited use. For example, we observed that the data resources were not fully integrated into the third SG generation, which led us to redesign the SG. The other finding is that the SG design should also be flexible enough to accommodate potential changes in the e-Infrastructure resources and in the three functional requirement groups due to expected evolution of research practices. Take for example of WS-PGRADE/gUSE SG [77], which was redesigned to allow flexibility to access multiple Distributed Computing Infrastructures (DCIs). One could argue that redesign of SGs is inevitable because it is challenging to foresee the potential future changes in requirements and e-Infrastructures. Nevertheless, it is crucial to encapsulate the SG functions into well-defined and generic services that can be reused and evolved independently as the SG is refined. A way to achieve this is to adopt Service Oriented Architecture (SOA), standards, and abstraction layers. Designing SGs based on these principles also enables reuse of software components and services across SGs and scientific disciplines. Encapsulating the computing management functions into an independent service in the design of the fourth SG generation helped us to reuse this service in the design of both the computational neuroscience and the medical chemistry SGs. On the other hand, designing generic services is more complex, which typically takes more time to realize and might deviate efforts from the domain-specific community requirements. It remains challenging to find the right balance between generic and specific design.

SG development is challenging because there is a plethora of alternative technologies to realize them. Moreover, the technologies for distributed computing, data management, and web development are evolving quickly. Evaluating all combinations of alternatives is nearly impossible, with the consequence that SG developers make choices based on their existing experience and collaboration. Therefore, it is particularly beneficial to share best practices for SG development in order to help developers to make better informed decisions. Discussing the design and development of SGs and sharing best practices has been the main topic of publications in the SG research community, for example, the EGI SG Primer [43] and the XSEDE SG cookbook [97]. There is also a trend to join efforts to build SGs, by means of which existing SGs are customized to develop new ones. The existing SGs already provide some SG functions such as integration and management of computing resources, reducing development time. However, this approach also confines the developers to a certain software stack and SG functions, which implies they need to understand the architecture and details of the existing gateway before being able to customize it or add new functions. In contrast

to this approach, it is also possible to develop SGs from scratch. Although this approach gives the developers full control over the software stack and SG functions, it increases the development time. It is therefore important to choose the approach that suits best the requirements based on available resources and expertise.

Our research also revealed three findings concerning the operation of SGs. The first and the most important finding is that the SG operation is a team work between scientific domain, SG, and e-Infrastructure experts. Unlike other scientific software that users can download and run on their computers, SGs require deployment and operation to offer services for biomedical researchers. This requires funding for operation, which hampers sustainability as discussed in Section 8.3. The second finding is that the design and development decisions have consequences for the operation of SGs. For example, the third SG generation provided lower level information about computations to the scientists, which have been shielded from them in the fourth generation. Although this increased the SG usability for scientists and streamlined operation, it also increased the effort required to operate the SG behind the scenes. The third finding is that the operation of SGs is challenging particularly when something goes wrong, which is likely to happen in such large and complex distributed systems. In such cases, the operation team needs to investigate and solve the causes of the problem, which might be domain-specific, related to e-Infrastructures, or to the various software layers in between. e-Infrastructure-related problems are especially challenging to troubleshoot because of complex interaction among different parts of the gateway. Therefore it is important to design and develop SGs considering also how to facilitate operation and troubleshooting.

A final remark about the design, development, and operation of SGs is that all of these efforts should be performed in partnership with the specific research communities they address. In other words, as also recognized in [29], these efforts should be approached as community building processes, which means they are performed *with* the specific research communities, and not *for* them. This approach aims to ensure the usability and effectiveness of SGs by involving research communities from the beginning. Moreover, based on our experience, it helps to build a vibrant ecosystem that is necessary to sustain SGs in the long run.

8.3 Offering Science Gateways as Sustainable Services

In this section we discuss the answers to the third research question: "*How to offer these SGs as sustainable services for biomedical researchers?*"

In our study we have analyzed sustainability of SGs in a methodological way using the Business Model Canvas (BMC). We found that, as in any business, there are two sides to sustainability: costs and revenues. However, in the case of SGs the costs are typically high and revenues are often low, which make the business case very challenging.

About the costs, as stressed also in Section 8.2, SG design and development is costly because it is a complex process due to a large set of community-specific requirements that need to be addressed. In addition to that, the development of SGs is a non-ending process because they need to be maintained and adapted according to the evolving community requirements, e-Infrastructures, and technologies. Moreover, their oper-

ation is costly because it requires a team including domain, SG, and e-Infrastructure experts. Note also that the design and implementation choices may influence the costs greatly, for example, by affecting the amount of effort required to operate or maintain the gateway. Therefore, it is important to reduce the costs in order to increase SG sustainability. One example is to reduce development costs by reusing technologies and joining efforts with developers from other communities, as it has been done in the SCI-BUS project [157]. Another way is to invest on autonomous error recovery, which can reduce the operation efforts, as it is the case in the Virtual Imaging Platform (VIP) SG [58].

The revenues often come from public funding of SG projects that are focused on innovation, which typically leave the operation phase uncovered [123]. We found that researchers are not usually in a position to pay themselves for the usage of SGs, which makes the pay-per-use model difficult to implement. Another type of revenue could be from partnership between SG and scientific research communities in which they form joint long-term projects to design, develop, and operate SGs for a particular community, as it is the case of MoSGrid [55] and WS-PGRADE [77] SGs. In such a complex scenario it is still unclear from where and in which form the revenues will come from.

Note also that our observations are still limited to our own environment and experiences. It is unclear to what extent they have been influenced by the culture and requirements of scientific communities that we collaborated with, or the technologies that were used to implement our SGs. However, the attention given to sustainability worldwide is an indication that this is a challenging and important aspect of SGs.

8.4 Essential Science Gateway Functions

In this section we discuss the answers to the fourth research question: "*What are the essential functions of SGs? How can these guide future research and development on SG?*"

The need to organize concepts and technologies related to SGs has also been recognized by other SG researchers, however they mostly focus at the technical level [43, 97]. One of the main findings of this research is the identification of the three functional requirement groups related to data, computing, and collaboration. In order to address these requirements, any SG needs to provide a set of essential functions. We have identified these essential functions and organized them into a reference model coined Science Gateway Canvas (SGC). This reference model classifies the functions into groups and categories. The most prominent SG function groups concern resource management, namely, data, computing, and collaboration management. On top of these, there is another category of universal management functions that cross-cut individual resource management, for example for security and provenance management. Finally, there is another category for delivery functions that are used by users and external programs.

The reference model provided by the SGC helps newcomers to locate, understand and put various concepts and technologies related to SGs into perspective. It also serves as a guide to identify the requirements and functions of SGs. The ultimate goal of SGC is to facilitate construction and delivery of SG for biomedical researchers, enhancing their research practices.

The application of SGC to compare a set of popular SG frameworks qualitatively confirmed its practicality. SGC owes its practicality to a large extent to its abstraction level that is focused on the functional aspects of SGs rather on technical details. This can be seen both as its advantage and its disadvantage. On the one hand, it makes it feasible to analyze and compare such varied set of SGs and SG technologies. On the other hand it is detached from implementation and technical details of SGs and SG technologies, which would be extremely useful. The initial feedback from the SG research community about the SGC is that it is useful, but there are also other function groups to consider, for example, related to learning and operation activities. This means our analysis needs to be extended to build a more comprehensive reference model that includes other functions that we overlooked.

8.5 Future Research

The lessons learned in this research motivated the development of yet another generation of SGs. At the time of writing this thesis the fifth SG generation, coined Rosemary, is being designed in partnership with computational neuroscience, genomics, and assisted human reproduction research communities. The design of Rosemary considers all the three functional requirement groups related to data, computing, and collaboration. Moreover, its implementation aims to ensure flexibility to accommodate changing requirements and e-Infrastructures. We also need to investigate the interaction with other existing SGs to reduce the efforts and increase its sustainability.

Another possible line of further research is to study the business models of SGs that attracted hundreds of users and sustained their operation successfully. We hope that this study will lead to the discovery of a few common patterns that could be later formulated into new business models. We also hope that this effort will reveal best practices to increase the value of SGs and decrease their design, development, and operation costs. Such a study would contribute to the sustainability of future SGs as useful tools for biomedical research and other disciplines.

Finally to better understand the fundamentals of SGs for biomedical research and other disciplines, we would like to extend the SGC in order to build a comprehensive reference model. Further analysis of SGs from different disciplines, and in partnership with their designers, is required to achieve this goal. Such a comprehensive reference model facilitates SG developers to analyze alternative SG technologies in a structured way. The SGC can also be extended with a detailed list of SG functions, to help as a roadmap for SG designers. We hope that the adoption of SGC by the SG research community could in the future contribute to research that will reduce the efforts to design and develop SGs, paving the way for more widely adoption of this useful tool in biomedical research and other disciplines.

8.6 Closing Remarks

In this thesis we advanced the understanding of the fundamentals of SGs for biomedical research by organizing the findings about the requirements of biomedical researchers. We also organized the considerations about the design, development, operation, and

sustainability of effective SGs. Moreover, we constructed a few successful SGs that were adopted by a large number of scientists and facilitated their biomedical big data analysis on e-Infrastructures. Finally we proposed a reference model that organizes the essential functions of SGs. We think that these efforts will facilitate design, development, operation, sustainability, and most importantly, adoption of SGs for biomedical research.

Appendix

List of Acronyms

ACE	Amsterdam Center for Entrepreneurship
ACM	Association for Computing Machinery
ADNI	Alzheimer's Disease Neuroimaging Initiative
ALS	Amyotrophic Lateral Sclerosis
AMC	Academic Medical Center
AMC-NSG	AMC computational NeuroScience Gateway
AMGA	ARDA Metadata Catalogue
API	Application Programming Interface
ARC	Advanced Resource Connector
ARDA	A Realisation of Distributed Analysis for LHC
ASM	gUSE Application Specific Module
BDII	Berkeley Database Information Index
BEDPOSTX	Bayesian Estimation of Diffusion Parameters Obtained using Sampling Techniques for modeling crossing fibers
BES	Basic Execution Service
BIC	Brain Imaging Center
BIRN	Biomedical Informatics Research Network
BLAST	Basic Local Alignment Search Tool
BMC	Business Model Canvas
BOINC	Berkeley Open Infrastructure for Network Computing
CCGrid	IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing
CCPE	Concurrency and Computation: Practice and Experience
CE	Compute Element
CIPRES	CyberInfrastructure for Phylogenetic RESearch
CLI	Command-Line Interface

CSF	CerebroSpinal Fluid
DAG	Directed Acyclic Graph
DARE	Distributed Application Runtime Environment
DCI	Distributed Computing Infrastructure
DECIDE	Diagnostic Enhancement of Confidence by an International Distributed Environment
DICOM	Digital Imaging and COmmunication in Medicine
DIRAC	Distributed Infrastructure with Remote Agent Control
DMZ	DeMilitarized Zone
DNA	DeoxyriboNucleic Acid
DOI	Document Object Identifier
DTI	Diffusion Tensor Imaging
DTS	Data Transport Service
eBrowser	e-BioInfra Browser Portlet
EBI	European Bioinformatics Institute
EC2	Amazon Elastic Compute Cloud
eCAT	e-BioInfra Catalog
EGEE	Enabling Grids for E-sciencE
EGI	European Grid Infrastructure
EPFL	École Polytechnique Fédérale de Lausanne
FGCS	Future Generation Computer Systems
fMRI	functional Magnetic Resonance Imaging
FMRIB	Functional MRI of the Brain
FSL	FMRIB Software Library
FTP	File Transfer Protocol
GCE	Gateway Computing Environments
GENIUS	Grid Enabled web eNvironment for site Independent User job Submission
gLite	Lightweight Middleware for Grid Computing
GridFTP	Grid File Transfer Protocol
GSI	Grid Security Infrastructure
GT	Globus Toolkit
GUI	Graphical User Interface
gUSE	grid and cloud User Support Environment

GWT	Google Web Toolkit
HPC	High Performance Computing
HPCN	High Performance Computing and Networking
HPDC	ACM Symposium on High-Performance Parallel and Distributed Computing
HTC	Hight Throughput Computing
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ICT	Information and Communications Technology
IEEE	Institute of Electrical and Electronics Engineers
IMS	Information Management System
iRODS	integrated Rule-Oriented Data System
IT	Information Technology
IVF	In Vitro Fertilization
IWSG	International Workshop on Science Gateways
IWSG-Life	International Workshop on Science Gateways for Life Sciences
JGC	Journal of Grid Computing
JISC	Joint Information Systems Committee
JSDL	Job Submission Description Language
JSP	JavaServer Pages
KEBB	Clinical Epidemiology, Biostatistics, and Bioinformatics (Dutch: Klinische Epidemiologie, Biostatistiek en Bio-informatica)
LATAM	Latin American e-Science workshop
LCG	LHC Computing Grid
LDAP	Lightweight Directory Access Protocol
LFC	LCG File Catalog
LHC	Large Hadron Collider
LONI	Laboratory Of Neuro Imaging
LSF	Load Sharing Facility
MIK	Medical Informatics (Dutch: Medische InformatieKunde)
MoSGrid	Molecular Simulation Grid
MRI	Magnetic Resonance Imaging
MVC	Model–View–Controller
N4U	neuGRID for you

NBIC	Netherlands BioInformatics Centre
NMR	Nuclear Magnetic Resonance
NWO	Netherlands Organisation for Scientific Research (Dutch: Nederlandse Organisatie voor Wetenschappelijk Onderzoek)
OGCE	Open Gateway Computing Environment
OGE	Oracle / Sun Grid Engine
ORM	Object-Relational Mapping
OSG	Open Science Grid
PACS	Picture Archiving and Communications System
PBS	Portable Batch System
PET	Positron Emission Tomography
PI	Principal Investigator
PKI	Public Key Infrastructure
PM	Processing Manager
P&S	Product and Services
PUCOWO	P-GRADE Portal User COmmunity WOrkshop
REST	Representational State Transfer
S3	Amazon Simple Storage Service
SAGA	Simple API for Grid Applications
SAML	Security Assertion Markup Language
SAXS	Small Angle X-ray Scattering
SCI-BUS	SCIentific gateway Based User Support
SCREAM	Science of Cyberinfrastructure: Research, Experience, Applications and Models
SE	Storage Element
SG	Science Gateway
SGC	Science Gateway Canvas
SGI	Science Gateways Institute
SHIWA	SHaring Interoperable Workflows for large-scale scientific simulations on Available DCIs
SILS	Swammerdam Institute for Life Sciences
SINAPAD	Brazilian national high-performance computing network (Portuguese: SIstema NAcional de Processamento de Alto Desempenho)
SLF4J	Simple Logging Facade for Java

SLURM	Simple Linux Utility for Resource Management
SOA	Service Oriented Architecture
SPECT	Single-Photon Emission Computed Tomography
SPSAS	São Paulo School of Advanced Science on e-Science for BioEnergy Research
SRM	Storage Resource Management
SSH	Secure Shell
SSO	Single Sign-On
TCGA	The Cancer Genome Atlas
TORQUE	Terascale Open-Source Resource and Queue Manager
UGE	Univa Grid Engine
UI	User Interface
UNICORE	Uniform Interface to Computing Resources
URL	Uniform Resource Locator
UvA	University of Amsterdam
UX	User eXperience
VBrowser	Virtual Resource Browser
VIP	Virtual Imaging Platform
VL-e	Virtual Laboratory for e-Science
VO	Virtual Organization
VOMS	Virtual Organization Membership Service
VRE	Virtual Research Environment
WebDAV	Web Distributed Authoring and Versioning
WfMS	Workflow Management System
WMS	Workload Management System
WN	Worker Node
WS-PGRADE	Web Service – Parallel Grid Run-time and Application Development Environment
WS-PGRADE/gUSE	Web Service – Parallel Grid Run-time and Application Development Environment / grid and cloud User Support Environment
XML	eXtensible Markup Language
XNAT	eXtensible Neuroimaging Archive Toolkit
XSEDE	eXtreme Science and Engineering Discovery Environment

Bibliography

[1] H. Akil, M. E. Martone, and D. C. Van Essen. "Challenges and Opportunities in Mining Neuroscience Data". In: *Science* 331.6018 (2011), pp. 708–712.

[2] R. Alfieri et al. "VOMS, an Authorization System for Virtual Organizations". In: *Grid Computing*. Ed. by F. Fernández Rivera et al. Vol. 2970. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, pp. 33–40.

[3] R. Allan. *Virtual Research Environments: From Portals to Science Gateways*. Chandos Information Professional Series. Elsevier Science, 2009.

[4] R. Allan et al. *Roadmap for a UK Virtual Research Environment*. Tech. rep. JISC VRE Working Group, 2004.

[5] M. Altunay et al. "A Science Driven Production Cyberinfrastructure—the Open Science Grid". In: *Journal of Grid Computing* 9 (2011), pp. 201–218.

[6] *AMC Neuroscience Gateway Portlet on the SCI-BUS portlet repository*. Online: https://scibus-repo.cpc.wmin.ac.uk/scibus-repo/public/details-view.xhtml?portletID=4353. Accessed: 2013-12-01.

[7] R. Ananthakrishnan et al. "Globus platform-as-a-service for collaborative science applications". In: *Concurrency and Computation: Practice and Experience* 27.2 (2015), pp. 290–305.

[8] G. Andronico et al. "e-Infrastructures for e-Science: A Global View". In: *Journal of Grid Computing* 9 (2011), pp. 155–184.

[9] V. Ardizzone et al. "The DECIDE Science Gateway". In: *Journal of Grid Computing* 10.4 (2012), pp. 689–707.

[10] Australian Research Council. *ARC E-research Support: Invitation for Funding Proposals under ARC Special Research Initiatives for Funding to Commence in 2005*. Online: http://www.arc.gov.au/pdf/Invitation_for_Funding_Proposals_ER05_060105.pdf. Accessed: 2015-06-26. 2005.

[11] A. Balasko, Z. Farkas, and P. Kacsuk. "Building Science Gateways By Utilizing The Generic WS-PGRADE/gUSE Workflow System". In: *Computer Science* 14.2 (2013).

[12] R. Barbera et al. "A grid portal with robot certificates for bioinformatics phylogenetic analyses". In: *Concurrency and Computation: Practice and Experience* 23.3 (2011), pp. 246–255.

[13] R. Barbera et al. “The GENIUS Grid Portal: Its Architecture, Improvements of Features, and New Implementations about Authentication and Authorization”. In: *16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. 2007, pp. 279–283.

[14] J. Basney, M. Humphrey, and V. Welch. “The MyProxy online credential repository”. In: *Software: Practice and Experience* 35.9 (2005), pp. 801–816.

[15] *Berkeley Database Information Index (BDII)*. Online: https://twiki.cern.ch/twiki/bin/view/EGEE/BDII. Accessed: 2013-12-01.

[16] I. Bertini et al. “A Grid-enabled web portal for NMR structure refinement with AMBER”. In: *Bioinformatics* (2011).

[17] G. Birkenheuer et al. “MoSGrid: Progress of Workflow driven Chemical Simulations”. In: *Proceedings of Grid Workflow Workshop (GWW)*. 2011.

[18] S. Blank. “Why the lean start-up changes everything”. In: *Harvard Business Review* 91.5 (2013), pp. 63–72.

[19] A. M. J. J. Bonvin, A. Rosato, and T. A. Wassenaar. “The eNMR platform for structural biology”. In: *Journal of structural and functional genomics* 11.1 (Mar. 2010), pp. 1–8.

[20] V. Breton et al. “The Healthgrid White Paper”. In: *Stud Health Technol Inform* 112 (2005), pp. 249–321.

[21] M. Bubak et al. “Virtual Laboratory for Development and Execution of Biomedical Collaborative Applications”. In: *21st International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, 2008, pp. 373–378.

[22] K. H. Buetow. “Cyberinfrastructure: empowering a “third way” in biomedical research”. In: *Science* 308.5723 (May 2005), pp. 821–824.

[23] *Business Model Canvas - From Wikipedia, the free encyclopedia*. Online: http://en.wikipedia.org/wiki/Business_Model_Canvas. Accessed: 2014-08-24.

[24] *Business Model Canvas for User Experience*. Online: http://grasshopperherder.com/business-model-canvas-for-user-experience. Accessed: 2014-08-24.

[25] M. W. A. Caan et al. “Evolution of grid-based services for Diffusion Tensor Image analysis”. In: *Future Generation Computer Systems* 28.8 (2012), pp. 1194 –1204.

[26] M. W. A. Caan et al. “Gridifying a Diffusion Tensor Imaging Analysis Pipeline”. In: *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*. 2010, pp. 733–738.

[27] California Biomedical Research Association. *Fact Sheet: What is Biomedical Research?* Online: http://ca-biomed.org/pdf/media-kit/fact-sheets/FS-WhatBiomedical.pdf. Accessed: 2015-06-24.

[28] S. Camarasu-Pop et al. “Dynamic Partitioning of GATE Monte-Carlo Simulations on EGEE”. In: *Journal of Grid Computing* 8.2 (2010), pp. 241–259.

[29] A. Carusi and T. Reimer. *Virtual research environment collaborative landscape study-A JISC funded project*. Tech. rep. Joint Infrastructure Systems Committee, 2010.

[30] A. Casajus et al. “DIRAC pilot framework and the DIRAC Workload Management System”. In: *Journal of Physics: Conference Series* 219.6 (2010), p. 062049.

[31] S. Cholia and T. Sun. “The NEWT Platform: An Extensible Plugin Framework for Creating ReSTful HPC APIs”. In: *Proceedings of the 9th Gateway Computing Environments Workshop*. GCE ’14. Piscataway, NJ, USA: IEEE Press, 2014, pp. 17–20.

[32] D. De Roure, N. Jennings, and N. Shadbolt. “The Semantic Grid: Past, Present, and Future”. In: *Proceedings of the IEEE* 93.3 (Mar. 2005), pp. 669 –681.

[33] E. Deelman et al. “Pegasus, a workflow management system for science automation”. In: *Future Generation Computer Systems* (2014).

[34] E. Deelman et al. “Workflows and e-Science: An overview of workflow system features and capabilities”. In: *Future Generation Computer Systems* 25.5 (2009), pp. 528 –540.

[35] *Developer-recommended Software for TeraGrid Science Gateways*. Online: http://gw55.quarry.iu.teragrid.org/mediawiki/index.php?title=Developer-recommended_Software_for_TeraGrid_Science_Gateways. Accessed: 2015-02-20.

[36] *DIANE: Distributed Analysis Environment*. Online: http://cern.ch/diane. Accessed: 2013-12-01.

[37] *Digital Imaging and COmmunications in Medicine*. Online: http://medical.nema.org. Accessed: 2013-12-01.

[38] I. Dinov et al. “Neuroimaging Study Designs, Computational Analyses and Data Provenance Using the LONI Pipeline”. In: *PLoS ONE* 5.9 (Sept. 2010), e13070.

[39] *DTI Preprocessing on the AMC-NSG*. Online: https://neuro.ebioscience.amc.nl/portal/web/nsg/dtipreprocessing. Accessed: 2015-01-15.

[40] *DTI Preprocessing on the e-BioinfraGateway*. Online: http://www.bioinformaticslaboratory.nl/twiki/bin/view/EBioScience/PredtiUserDoc. Accessed: 2013-12-01.

[41] P. Dziubecki et al. “Easy Development and Integration of Science Gateways with Vine Toolkit”. In: *Journal of Grid Computing* (2012), pp. 1–15.

[42] *e-Bioinfra Gateway*. Online: http://orange.ebioscience.amc.nl/ebioinfragateway. Accessed: 2013-12-01.

[43] European Grid Infrastructure (EGI) Science Gateway Virtual Team. *Science Gateway Primer*. https://documents.egi.eu/document/1463. Accessed: 2013-12-01. 2012.

[44] *European Grid Infrastructure (EGI) Science Gateways*. Online: http://go.egi.eu/sciencegateways. Accessed: 2013-12-01.

[45] Z. Farkas and P. Kacsuk. “P-GRADE Portal: A generic workflow system to support user communities”. In: *Future Generation Computer Systems* 27.5 (May 2011), pp. 454–465.

[46] *Federal Enterprise Architecture Framework version 2*. Online: https://www.whitehouse.gov/sites/default/files/omb/assets/egov_docs/fea_v2.pdf. Accessed: 2015-05-04.

[47] T. Ferrari and L. Gaido. “Resources and Services of the EGEE Production Infrastructure”. In: *Journal of Grid Computing* 9 (2011), pp. 119–133.

[48] B. Fischl et al. “Automatically Parcellating the Human Cerebral Cortex”. In: *Cerebral Cortex* 14.1 (2004), pp. 11–22.

[49] S. M. Fisher, K. Phipps, and D. J. Rolfe. “ICAT Job Portal: a generic job submission system built on a scientific data catalog”. In: *Proceedings of 5th International Workshop on Science Gateways for Life Sciences*. IWSG 2013. 2013.

[50] *FMRIB’s Diffusion Toolbox - BEDPOSTX*. Online: http://fsl.fmrib.ox.ac.uk/fsl/fsl4.0/fdt/fdt_bedpostx.html. Accessed: 2013-12-01.

[51] M. Fraser. “Virtual research environments: overview and activity”. In: *Ariadne* 44.7 (2005).

[52] G. B. Frisoni et al. “Virtual imaging laboratories for marker discovery in neurodegenerative diseases”. In: *Nat Rev Neurol* 7.8 (Aug. 2011), pp. 429–438.

[53] T. Gee et al. “Data warehousing methods and processing infrastructure for brain recovery research”. In: *Arch Ital Biol* 148.3 (Sept. 2010), pp. 207–217.

[54] *Genome Compare on the e-BioinfraGateway*. Online: http://www.bioinformaticslaboratory.nl/twiki/bin/view/EBioScience/GenomeCompareUserDoc. Accessed: 2013-12-01.

[55] S. Gesing et al. “Molecular Simulation Grid (MosGrid): A Science Gateway Tailored to the Molecular Simulation Community”. In: *Science Gateways for Distributed Computing Infrastructures*. Ed. by P. Kacsuk. Cham, Switzerland: Springer International Publishing, 2014, pp. 151–165.

[56] S. Gesing et al. “Special Issue: Portals for life sciences—Providing intuitive access to bioinformatic tools”. In: *Concurrency and Computation: Practice and Experience* 23.3 (2011), pp. 223–234.

[57] S. Gesing et al. “Workflow Interoperability in a Grid Portal for Molecular Simulations”. In: *Proceedings of the International Worshop on Science Gateways (IWSG)*. Consorzio COMETA. Consorzio COMETA, 2010, pp. 44–48.

[58] T. Glatard et al. “A Virtual Imaging Platform for Multi-Modality Medical Image Simulation”. In: *Medical Imaging, IEEE Transactions on* 32.1 (Jan. 2013), pp. 110 –118.

[59] T. Glatard et al. “Flexible and Efficient Workflow Deployment of Data-Intensive Applications On Grids With MOTEUR”. In: *International Journal of High Performance Computing Applications* 22.3 (Aug. 2008), pp. 347–360.

[60] C. Goble and D. Roure. “The Fourth Paradigm: Data-Intensive Scientific Discovery”. In: ed. by T. Hey, S. Tansley, and K. Tolle. Microsoft Research, 2009. Chap. The Impact of Workflow Tools on Data-centric Research, pp. 137–145.

[61] A. T. A. Gomes et al. “Experiences of the Brazilian national high-performance computing network on the rapid prototyping of science gateways”. In: *Concurrency and Computation: Practice and Experience* 27.2 (2015), pp. 271–289.

[62] T. Goodale et al. “SAGA: A Simple API for Grid Applications. High-level application programming on the Grid”. In: *Computational Methods in Science and Technology* 12.1 (2006), pp. 7–20.

[63] M. M. van der Graaff et al. “Upper and extra-motoneuron involvement in early motoneuron disease: a diffusion tensor imaging study”. In: *Brain* 134 (Apr. 2011), pp. 1211–1228.

[64] K. G. Helmer et al. “Enabling collaborative research using the Biomedical Informatics Research Network (BIRN)”. In: *Journal of the American Medical Informatics Association* 18.4 (2011), pp. 416–422.

[65] T. Hey, S. Tansley, and K. M. Tolle. “Jim Gray on eScience: a transformed scientific method”. In: *The Fourth Paradigm*. Ed. by T. Hey, S. Tansley, and K. M. Tolle. Microsoft Research, 2009.

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

[67] T. Hey and A. Trefethen. “The Data Deluge: An e-Science Perspective”. In: *Grid Computing*. John Wiley & Sons, Ltd, 2003, pp. 809–824.

[68] T. Hey and A. E. Trefethen. “Cyberinfrastructure for e-Science”. In: *Science* 308.5723 (2005), pp. 817–821.

[69] *International Workshop of Science Gateways (IWSG) events*. Online: http://iwsg-life.org/site/iwsglife/events. Accessed: 2014-02-28.

[70] iSGTW. *Grid computing aids study into post-traumatic stress among Afghanistan war veterans*. Online: http://www.isgtw.org/feature/grid-computing-aids-study-post-traumatic-stress-among-afghanistan-war-veterans. Accessed: 2015-01-15. Sept. 2014.

[71] iSGTW. *Identifying the first signs of Alzheimer’s and dementia*. online: http://goo.gl/mXvS7. May 2011.

[72] iSGTW. *neuGRID, the European online diagnosis environment for Alzheimer’s, goes global with outGRID and the United Nations*. online: http://goo.gl/lhKvF. Feb. 2012.

[73] M. M. Jaghoori et al. “A Grid-Enabled Virtual Screening Gateway”. In: *Science Gateways (IWSG), 2014 6th International Workshop on*. IEEE. Dublin, Ireland, 2014, pp. 24–29.

[74] S. Jan et al. “GATE : a simulation toolkit for PET and SPECT”. In: *arXiv.org* physics.med-ph.19 (Aug. 2004), pp. 4543–4561.

[75] P. Kacsuk. “P-GRADE portal family for grid infrastructures”. In: *Concurrency and Computation: Practice and Experience* 23.3 (2011), pp. 235–245.

[76] P. Kacsuk. *Science Gateways for Distributed Computing Infrastructures: Development Framework and Exploitation by Scientific User Communities*. Springer, 2014.

[77] P. Kacsuk et al. “WS-PGRADE/gUSE Generic DCI Gateway Framework for a Large Variety of User Communities”. In: *Journal of Grid Computing* 10.4 (2012), pp. 601–630.

[78] S. D. Kahn. “On the Future of Genomic Data”. In: *Science* 331.6018 (2011), pp. 728–729.

[79] J. Kim, S. Maddineni, and S. Jha. "Building gateways for life-science applications using the dynamic application runtime environment (DARE) framework". In: *Proceedings of the 2011 TeraGrid Conference: Extreme Digital Discovery*. TG '11. New York, NY, USA: ACM, 2011, 38:1–38:8.

[80] T. Kiss. "Science Gateways for the Broader Take-up of Distributed Computing Infrastructures". In: *Journal of Grid Computing* 10 (2012), pp. 599–600.

[81] T. Kiss et al. "Parameter Sweep Workflows for Modelling Carbohydrate Recognition". In: *Journal of Grid Computing* 8 (2010), pp. 587–601.

[82] P. L. Klarenbeek et al. "Human T-cell memory consists mainly of unexpanded clones". In: *Immunology Letters* 133.1 (Sept. 2010), pp. 42–48.

[83] J. Kocot et al. "A Framework for Domain-Specific Science Gateways". In: *eScience on Distributed Computing Infrastructure. Achievements of PLGrid Plus Domain-Specific Services and Tools: LNCS 8500.* Switzerland: Springer, 2014, pp. 130–46.

[84] V. Korkhov et al. "Exploring workflow interoperability tools for neuroimaging data analysis". In: *Proceedings of the 6th workshop on Workflows in support of large-scale science*. WORKS '11. New York, NY, USA: ACM, 2011, pp. 87–96.

[85] D. Krefting et al. "MediGRID: Towards a user friendly secured grid infrastructure". In: *Future Generation Computer Systems* 25.3 (2009), pp. 326–336.

[86] J. Krüger et al. "The MoSGrid Science Gateway – A Complete Solution for Molecular Simulations". In: *Journal of Chemical Theory and Computation* 10.6 (2014), pp. 2232–2245.

[87] B. de Kwaasteniet et al. "Relation Between Structural and Functional Connectivity in Major Depressive Disorder". In: *Biological Psychiatry* 74.1 (2013), pp. 40 –47.

[88] E. Laure et al. "Middleware for the next generation Grid infrastructure". In: *Computing in High Energy Physics and Nuclear Physics*. EGEE-PUB-2004-002. 2004, 4 p.

[89] K. A. Lawrence et al. "Who Cares About Science Gateways?: A Large-scale Survey of Community Use and Needs". In: *Proceedings of the 9th Gateway Computing Environments Workshop*. GCE '14. Piscataway, NJ, USA: IEEE Press, 2014, pp. 1–4.

[90] A. Lenards, N. Merchant, and D. Stanzione. "Building an Environment to Facilitate Discoveries for Plant Sciences". In: *Proceedings of the 2011 ACM Workshop on Gateway Computing Environments*. GCE '11. New York, NY, USA: ACM, 2011, pp. 51–58.

[91] A. Luyf et al. "Initial steps towards a production platform for DNA sequence analysis on the grid". In: *BMC Bioinformatics* 11.1 (2010), p. 598.

[92] S. Maddineni et al. "Distributed Application Runtime Environment (DARE): A Standards-based Middleware Framework for Science-Gateways". In: *Journal of Grid Computing* 10.4 (2012), pp. 647–664.

[93] S. Madougou et al. "Provenance for distributed biomedical workflow execution". In: *Studies in Health Technology and Informatics*. Vol. 175. 2012, pp. 91–100.

[94] S. Madougou et al. “Characterizing workflow-based activity on a production e-infrastructure using provenance data”. In: *Future Generation Computer Systems* 29.8 (2013), pp. 1931 –1942.

[95] C. Marco et al. “The gLite Workload Management System”. In: *Advances in Grid and Pervasive Computing*. Ed. by N. Abdennadher and D. Petcu. Vol. 5529. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, pp. 256–268.

[96] D. S. Marcus et al. “The Extensible Neuroimaging Archive Toolkit: an informatics platform for managing, exploring, and sharing neuroimaging data”. In: *Neuroinformatics* 5.1 (2007), pp. 11–34.

[97] S. Marru et al. “Authoring a Science Gateway Cookbook”. In: *Cluster Computing (CLUSTER), 2013 IEEE International Conference on*. Sept. 2013, pp. 1–3.

[98] V. Marx. “Biology: The big challenges of big data”. In: *Nature* 498.7453 (June 2013), pp. 255–260.

[99] D. J. H. Mathews et al. “Access to Stem Cells and Data: Persons, Property Rights, and Scientific Progress”. In: *Science* 331.6018 (2011), pp. 725–727.

[100] M. McLennan and R. Kennell. “HUBzero: A Platform for Dissemination and Collaboration in Computational Science and Engineering”. In: *Computing in Science Engineering* 12.2 (2010), pp. 48–53.

[101] M. McLennan et al. “HUBzero and Pegasus: integrating scientific workflows into science gateways”. In: *Concurrency and Computation: Practice and Experience* 27.2 (2015), pp. 328–343.

[102] M. A. Miller, W. Pfeiffer, and T. Schwartz. “The CIPRES science gateway: a community resource for phylogenetic analyses”. In: *Proceedings of the 2011 TeraGrid Conference: Extreme Digital Discovery*. TG ’11. New York, NY, USA: ACM, 2011, 41:1–41:8.

[103] *Model–view–controller – From Wikipedia, the free encyclopedia*. Online: http://en.wikipedia.org/wiki/Model-view-controller. Accessed: 2013-12-01.

[104] J. Montagnat et al. “NeuroLOG: a community-driven middleware design”. In: *Stud Health Technol Inform* 138 (2008), pp. 49–58.

[105] J. Montagnat et al. “A data-driven workflow language for grids based on array programming principles”. In: *Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science (WORKS)*. Nov. 2009.

[106] J. T. Moscicki et al. “Processing moldable tasks on the grid: Late job binding with lightweight user-level overlay”. In: *Future Generation Computer Systems* 27.6 (June 2011), pp. 725–736.

[107] Nature Staff. “Community cleverness required”. In: *Nature* 455.7209 (Sept. 2008), pp. 1–1.

[108] Nature Staff. “Focus on big data”. In: *Nat Neurosci* 17.11 (Nov. 2014), pp. 1429–1429.

[109] J. Novotny, M. Russell, and O. Wehrens. “GridSphere: An Advanced Portal Framework”. In: *Proceedings of the 30th Euromicro Conference*. IEEE, 2004, pp. 412–419.

[110] J. Novotny, M. Russell, and O. Wehrens. "GridSphere: a portal framework for building collaborations". In: *Concurrency and Computation: Practice and Experience* 16.5 (2004), pp. 503–513.

[111] S. D. Olabarriaga et al. "Virtual Lab for fMRI: Bridging the Usability Gap". In: *2nd IEEE International Conference on e-Science and Grid Computing*. Dec. 2006, p. 53.

[112] S. D. Olabarriaga, T. Glatard, and P. T. de Boer. "A Virtual Laboratory for Medical Image Analysis". In: *IEEE Transactions on Information Technology in Biomedicine* 14.4 (2010), pp. 979–985.

[113] S. D. Olabarriaga et al. "From "low hanging" to "user ready": initial steps into a HealthGrid". In: *Global Healthgrid: e-Science Meets Biomedical Informatics - Proceedings of HealthGrid 2008*. Vol. 138. 2008, pp. 70–79.

[114] S. D. Olabarriaga et al. "Integrated Support for Medical Image Analysis Methods: From Development to Clinical Application". In: *IEEE Transactions on Information Technology in Biomedicine* 11.1 (2007), pp. 47–57.

[115] A. Osterwalder and Y. Pigneur. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Wiley Desktop Editions. Wiley, 2010.

[116] S. Pandey et al. "A grid workflow environment for brain imaging analysis on distributed systems". In: *Concurrency And Computation-Practice & Experience* 21.16 (2009), pp. 2118–2139.

[117] K. Peffers et al. "A Design Science Research Methodology for Information Systems Research". In: *Journal of Management Information Systems* 24.3 (2007), pp. 45–78.

[118] B. D. Peters et al. "Polyunsaturated Fatty Acid Concentration Predicts Myelin Integrity in Early-Phase Psychosis". In: *Schizophrenia Bulletin* 39.4 (2013), pp. 830–838.

[119] M. Pierce et al. "Apache Airavata: Design and Directions of a Science Gateway Framework". In: *6th International Workshop on Science Gateways*. IWSG 2014. June 2014, pp. 48–54.

[120] A. Redolfi et al. "Grid infrastructures for computational neuroscience: the neuGRID example". In: *Future Neurology* 4.6 (Nov. 2009), pp. 703–722.

[121] A. Rienstra et al. "Symptom validity testing in memory clinics: Hippocampal-memory associations and relevance for diagnosing mild cognitive impairment". In: *Journal of Clinical and Experimental Neuropsychology* 35.1 (2013), pp. 59–70.

[122] E. E. Schadt et al. "Computational solutions to large-scale data management and analysis". In: *Nat Rev Genet* 11.9 (Sept. 2010), pp. 647–657.

[123] SCI-BUS project. *D3.3: Workshop with Key Players*. Online: http://www.sci-bus.eu/documents/94981/616004/SCI-BUS_D3.3_v1.2.doc. Accessed: 2015-01-15. Sept. 2013.

[124] E. Sciacca et al. "VisIVO Science Gateway: a Collaborative Environment for the Astrophysics Community". In: *Proceedings of the 5th International Workshop on Science Gateways*. Zurich, Switzerland, 2013.

[125] *Science Gateways on the XSEDE (Extreme Science and Engineering Digital Environment) website*. Online: https://www.xsede.org/gateways-overview. Accessed: 2013-12-01.

[126] Science Staff. "Challenges and Opportunities". In: *Science* 331.6018 (2011), pp. 692–693.

[127] S. Shahand et al. "Front-ends to Biomedical Data Analysis on Grids". In: *Proceedings of HealthGrid 2011*. Bristol, UK, 2011.

[128] S. Shahand et al. "Integrated support for neuroscience research: from study design to publication". In: *Studies in Health Technology and Informatics*. Vol. 175. 2012, pp. 195–204.

[129] S. Shahand et al. "A data-centric neuroscience gateway: design, implementation, and experiences". In: *Concurrency and Computation: Practice and Experience* 27.2 (2015), pp. 489–506.

[130] S. Shahand et al. "A Grid-Enabled Gateway for Biomedical Data Analysis". In: *Journal of Grid Computing* 10.4 (2012), pp. 725–742.

[131] R. Ferreira da Silva et al. "Multi-infrastructure workflow execution for medical simulation in the Virtual Imaging Platform". In: *Proceedings of HealthGrid 2011*. Bristol, UK, 2011.

[132] S. Sivagnanam et al. "Introducing The Neuroscience Gateway". In: *Proceedings of the 5th International Workshop on Science Gateways*. June 2013.

[133] T. Soddemann. "Science gateways to DEISA: user requirements, technologies, and the material sciences and plasma physics gateway". In: *Concurrency and Computation: Practice and Experience* 19.6 (2007), pp. 839–850.

[134] C. A. Stewart, G. T. Almes, and B. C. Wheeler. *Cyberinfrastructure Software Sustainability and Reusability: Report from an NSF-funded workshop*. Tech. rep. Bloomington, Indiana, USA: Indiana University, 2010.

[135] G. A. Stewart et al. "Storage and data management in EGEE". In: *Proceedings of the fifth Australasian symposium on ACSW frontiers - Volume 68*. ACSW '07. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2007, pp. 69–77.

[136] *The Alzheimer's Disease Neuroimaging Initiative (ADNI) Website*. Online: http://www.adni-info.org/. Accessed: 2013-12-01.

[137] *The AMC BIC (Brain Imaging Center) Website*. Online: http://www.lebic-amc.nl. Accessed: 2013-12-01.

[138] *The AMC e-Science Website*. Online: http://www.ebioscience.amc.nl. Accessed: 2014-12-01.

[139] *The BiG Grid Project Website*. Online: http://www.biggrid.nl. Accessed: 2013-12-01.

[140] *The Biomedical Informatics Research Network (BIRN) Initiative Website*. Online: http://www.birncommunity.org/. Accessed: 2015-06-29.

[141] *The Brainmap Database*. Online: http://brainmap.org. Accessed: 2013-12-01.

[142] *The Brede Database*. Online: http://neuro.imm.dtu.dk/services/jerne/brede. Accessed: 2013-12-01.

[143] *The Business Model Canvas – Your business model on one page*. Online: http://www.businessmodelgeneration.com/canvas/bmc. Accessed: 2014-08-24.

[144] *The Cancer Genome Atlas Project Website*. Online: http://cancergenome.nih.gov/. Accessed: 2015-06-26.

[145] *The Catania Science Gateway Framework*. Online: http://www.catania-science-gateways.it. Accessed: 2013-12-01.

[146] *The COMMIT Project Website*. Online: http://www.commit-nl.nl. Accessed: 2014-12-01.

[147] *The Enginframe Project Website*. Online: http://www.enginframe.com. Accessed: 2013-12-01.

[148] *The ER-FLOW Project Website*. Online: http://www.erflow.eu. Accessed: 2015-01-15.

[149] *The gLite Project Website*. Online: http://glite.cern.ch. Accessed: 2013-12-01.

[150] *The Google Web Toolkit*. Online: https://developers.google.com/web-toolkit. Accessed: 2013-12-01.

[151] *The Hibernate Project Website*. Online: http://www.hibernate.org. Accessed: 2013-12-01.

[152] *The JISC Website*. Online: http://jisc.ac.uk/. Accessed: 2015-06-28.

[153] *The Liferay Project Website*. Online: http://www.liferay.com. Accessed: 2013-12-01.

[154] *The myGrid Project Website*. Online: http://www.mygrid.org.uk/. Accessed: 2015-06-29.

[155] *The N4U (neuGRID for you) Project Website*. Online: http://neugrid4you.eu. Accessed: 2013-12-01.

[156] *The Pylons Project Website*. Online: http://www.pylonsproject.org. Accessed: 2013-12-01.

[157] *The SCI-BUS Project Website*. Online: http://www.sci-bus.eu. Accessed: 2014-12-01.

[158] *The Science Gateway Institute Website*. Online: http://sciencegateways.org. Accessed: 2014-12-01.

[159] *The Science Gateway Sustainability and Industry Utilization Workshop (organized by SCI-BUS) at IWSG2013*. Online: http://www.amiando.com/iwsg2013.html?page=902868. Accessed: 2015-01-15.

[160] *The Shiwa Project Website*. Online: http://www.shiwa-workflow.eu. Accessed: 2013-12-01.

[161] *The Spring Project Website*. Online: http://www.springsource.org. Accessed: 2013-12-01.

[162] *The SURFsara Website*. Online: http://www.surfsara.nl. Accessed: 2013-12-01.

[163] *The VisIVO Governance Model*. Online: https://indico.egi.eu/indico/contributionDisplay.py?contribId=145&confId=1019. Accessed: 2015-01-15. 2012.

[164] *The VL-e Toolkit (VBrowser) on Sourceforge*. Online: http://sourceforge.net/projects/vlet. Accessed: 2013-12-01.

[165] *The XSEDE (Extreme Science and Engineering Digital Environment) Website*. Online: http://www.xsede.org. Accessed: 2013-12-01.

[166] M. Thomas. "Special Issue: Workshop on Grid Computing Portals (GCE 2005)". In: *Concurrency and Computation: Practice and Experience* 19.12 (2007), pp. 1563–1570.

[167] J. Thornton, R. Apweiler, and E. Birney. *Annual Scientific Report 2013*. Tech. rep. EMBL-European Bioinformatics Institute, 2013.

[168] *Using an Aladdin eToken PRO to store grid certificates*. Online: http://wiki.nikhef.nl/grid/EToken. Accessed: 2013-12-01.

[169] M. Wewior et al. "The MoSGrid Gaussian Portlet – Technologies for the Implementation of Portlets for Molecular Simulations". In: *Proceedings of the International Worshop on Science Gateways (IWSG)*. 2010.

[170] *Why Lean Canvas vs Business Model Canvas?* Online: http://practicetrumpstheory.com/2012/02/why-lean-canvas. Accessed: 2014-08-24.

[171] N. Wilkins-Diehr et al. "TeraGrid Science Gateways and Their Impact on Science". In: *Computer* 41.11 (Nov. 2008), pp. 32 –41.

[172] N. Wilkins-Diehr. "Special Issue: Science Gateways—Common Community Interfaces to Grid Resources". In: *Concurrency and Computation: Practice and Experience* 19.6 (2007), pp. 743–749.

[173] N. Wilkins-Diehr, S. Gesing, and T. Kiss. "Science gateway workshops 2013 special issue conference publications". In: *Concurrency and Computation: Practice and Experience* 27.2 (2015), pp. 253–257.

[174] N. Wilkins-Diehr and K. A. Lawrence. *Opening Science Gateways to Future Success*. Online: http://sciencegateways.org/wp-content/uploads/2012/06/Final_Report_OCI-0948476.pdf. Accessed: 2015-01-15. Nov. 2012.

[175] G. A. van Wingen et al. "Persistent and reversible consequences of combat stress on the mesofrontal circuit and cognition". In: *Proceedings of the National Academy of Sciences* 109.38 (2012), pp. 15508–15513.

[176] *Working towards Sustainable Software for Science Practice and Experiences (WSSSPE) workshop series*. Online: http://wssspe.researchcomputing.org.uk. Accessed: 2015-01-15.

[177] J. Wu et al. "The Charité Grid Portal: User-friendly and Secure Access to Grid-based Resources and Services". In: *Journal of Grid Computing* 10 (2012), pp. 709–724.

[178] W. A. Wulf. "The national collaboratory—a white paper". In: *Towards a national collaboratory* (1989), pp. 17–18.

[179] *XSEDE Gateways Cookbook*. Online: https://www.xsede.org/web/gateways/gateways-cookbook. Accessed: 2015-02-20.

[180] E. Zerhouni. "The NIH roadmap". In: *Science* 302.5642 (2003), pp. 63–72.

Summary

Biomedical researchers are facing data deluge challenges such as dealing with large volume of complex heterogeneous data and complex and computationally demanding data processing methods. Such scale and complexity of biomedical research requires multi-disciplinary collaboration between scientists from different organizations.

Data-driven or e-Science methods are defined as a combination of Information Technology (IT) and science that enables scientists to tackle the data deluge challenges. The IT infrastructures that address these challenges are known as cyberinfrastructures or e-Infrastructures, which are the environments that provide collaborative sharing of distributed computing and data resources. However, e-Infrastructures fall short of high-level and customized services to support the needs of scientists genuinely, and scientists find interacting with e-Infrastructures challenging, as it requires detailed technical knowledge.

Science Gateway (SG) research addresses these drawbacks. SGs are web-based enterprise information systems that provide scientists with customized and easy access to community-specific data collections, computational tools, and collaborative services on e-Infrastructures.

The aim of the research presented in this thesis was to advance our understanding of the fundamentals of SGs for biomedical research. This understanding is important because it promotes cross-fertilization, facilitates design, development, and operation of new SGs, and, most importantly, it guides future research on SGs. These fundamentals were derived from an iterative study of concrete cases of specific biomedical research communities using a user-driven and bottom-up approach. This study resulted in the design and development of four SG generations, the last two of which are described in this thesis.

We have started our research by understanding the characteristics and requirements of biomedical researchers. We have found biomedical research requirements overwhelming and complex due to the diversity of biomedical researchers' background and expertise, and variety of their roles and tasks along the research life cycle. In spite of all this diversity, we identified three main groups of functional requirements at the core of any SG for biomedical research, which are related to data, computation, and collaboration. We have observed that it is easier to discover and see through the complex set of requirements of the biomedical research communities when organizing requirements explicitly around these three functional requirement groups. This shortens the progressive requirements discovery process that is often observed in SG projects and therefore facilitates construction of effective SGs for biomedical research.

We have continued our research by organizing the considerations about the design, development, and operation of effective SGs for biomedical research. Our most important finding is that all of the three functional requirement groups, namely, related to data, computing, and collaboration, should be considered in the design. This means that any SG should integrate all of the data, computing, and collaboration resources seamlessly. Failing to integrate any of these resources will result in a SG that is not effective and will find limited use. The other finding is that the SG design should also be flexible enough to accommodate potential changes in the e-Infrastructure resources and in the three functional requirement groups due to expected evolution of research practices. To achieve such flexibility, it is crucial to encapsulate SG functions into well-defined and generic services that can be reused and evolved independently as the SG is refined. We have also found SG development challenging because of the large number of alternative and evolving technologies to realize SGs. Therefore, it is particularly beneficial to share best practices and join efforts to build SGs. These help SG designers and developers to make better informed decisions and construct SGs quicker, which enhance effectiveness of SGs for biomedical research and reduce the costs to realize them. Regarding the operation of SGs, we have found that the SG operation is a team work between scientific domain, SG, and e-Infrastructure experts, which requires funding for operation. Moreover, we have found that the operation of SGs, which is challenging particularly when something goes wrong, is influenced largely by the decisions that have been taken during their design and development. Therefore it is important to design and develop SGs considering also how to facilitate operation and troubleshooting. Finally, we have found that all of the efforts related to design, development, and operation of SGs should be performed in partnership with the specific research communities they address. In other words, these efforts should be approached as community building processes, with the biomedical researchers involved from the beginning, to ensure effectiveness of SGs and to establish the vibrant ecosystems necessary for viable SGs.

The next topic we have approached in our research was about offering SGs as sustainable services for biomedical researchers. We have analyzed sustainability of SGs in a methodological way using the Business Model Canvas (BMC). We found that, as in any business, there are two sides to sustainability: costs and revenues. However, in the case of SGs the costs are typically high and revenues are often low, which makes the business case very challenging. Costs of design and development of SGs are high because they are complex processes due to a large set of community-specific requirements that need to be addressed. In addition to that, the development of SGs is a non-ending process because they need to be maintained and adapted according to the evolving community requirements, e-Infrastructures, and technologies. Moreover, their operation is costly because it requires a team including domain, SG, and e-Infrastructure experts. The revenues are low because they often come from public funding of SG projects that are focused on innovation, which typically leave the operation phase uncovered. We also found that researchers are not usually in a position to pay themselves for the usage of SGs, which makes the pay-per-use model difficult to implement. To enhance sustainability of SGs the costs should be decreased and the revenues should be increased. Although in this thesis a few examples are provided to illustrate possible methods to achieve sustainability, further research is required to identify proven methods and

validate them in the biomedical research community. Such a study would contribute to the sustainability of future SGs as useful tools for biomedical research and other disciplines.

Finally, we have addressed the need to organize concepts and technologies related to SGs. In order to address the three functional requirement groups, namely, related to data, computation, and collaboration, any SG needs to provide a set of essential functions. We have identified these essential functions and organized them into a reference model coined *Science Gateway Canvas (SGC)*. This reference model classifies the functions into groups and categories. The most prominent SG function groups concern resource management, namely, data, computing, and collaboration management. On top of these, there is another category of universal management functions that cross-cut individual resource management, namely, for coordination, security, provenance, and monitoring management. Finally, there is another category for delivery functions that are used by users and external programs. The reference model provided by the SGC helps newcomers to locate, understand and put various concepts and technologies related to SGs into perspective. It also serves as a guide to identify the requirements and functions of SGs. The ultimate goal of SGC is to facilitate construction and delivery of SG for biomedical researchers, enhancing their research practices. The initial feedback from the SG research community about the SGC is that it is useful, but there are also other function groups to consider, which requires further research. We hope that the adoption of SGC by the SG research community could in the future contribute to research that will reduce the efforts to design and develop SGs, paving the way for more widely adoption of this useful tool in biomedical research and other disciplines.

In this thesis we advanced the understanding of the fundamentals of SGs for biomedical research by organizing the findings about the requirements of biomedical researchers. We also organized the considerations about the design, development, operation, and sustainability of effective SGs. Moreover, we constructed a few successful SGs that were adopted by a large number of scientists and facilitated their biomedical big data analysis on e-Infrastructures. Finally we proposed a reference model that organizes the essential functions of SGs. We think that these efforts will facilitate design, development, operation, sustainability, and most importantly, adoption of SGs for biomedical research.

Samenvatting

Biomedische onderzoekers worden geconfronteerd met een stortvloed aan uitdagingen door de enorme hoeveelheid data die verwerkt dienen te worden. Deze data zijn vaak zeer heterogeen, en moeten worden geanalyseerd middels complexe data-analyse methodes die veel rekenkracht vergen. De schaal en complexiteit van biomedisch onderzoek vereist dan ook een multi-disciplinaire samenwerking tussen wetenschappers van verschillende organisaties.

Datagestuurde of e-Science methodes zijn gedefinieerd als een combinatie van Information Technology (IT) en wetenschap. Cyberinfrastructures of e-Infrastructures zijn IT infrastructuren die wetenschappers helpen de uitdagingen van het analyseren van grote hoeveelheden data aan te gaan. Deze e-infrastructuren bieden een analyse-omgeving die toegang geeft tot gedistribueerde, gedeelde rekenkracht en databronnen. Een belangrijk probleem is dat e-Infrastructures geen pasklare (high-level) services bieden die voorzien in de behoefte van biomedische wetenschappers. Ook vinden deze wetenschappers het lastig om met e-Infrastructures te werken omdat hiervoor een hoge mate van technische kennis vereist is.

Science Gateway (SG) onderzoek probeert oplossingen te vinden voor deze problemen. SGs zijn web-gebaseerde systemen die wetenschappers op maat gemaakte services bieden op e-Infrastructures.

Het doel van het onderzoek zoals beschreven in dit proefschrift was om meer inzicht te krijgen in de fundamentele functies, werking en eigenschappen van SGs gericht op biomedisch onderzoek. Dit inzicht in de basisbehoeften voor een SGs is belangrijk omdat dit kan bijdragen aan de ontwikkeling van nieuwe SGs met een verbeterd ontwerp en betere uitvoering van taken. Ook kan 'kruisbestuiving' plaatsvinden tussen verschillende SGs waarbij beiden verbeteren. Verbeterd inzicht in SGs zal echter vooral het startpunt zijn voor toekomstig onderzoek naar nieuwe SGs. De fundamenten zijn afgeleid uit iteratieve studies van concrete toepassingen van SGs binnen het biomedisch domein. Hierbij is gebruik gemaakt van een, door gebruikers gestuurde, 'bottom-up' methode. Deze studies hebben geleid tot het ontwerp en ontwikkeling van vier generaties van SGs. De laatste twee generaties worden besproken in dit proefschrift.

Allereerst is begonnen met het verkrijgen van inzicht in de specifieke eigenschappen en vereisten van biomedisch onderzoek. Het bleek dat biomedisch onderzoek een enorme hoeveelheid complexe eisen heeft. Dit heeft onder meer te maken met het feit dat er binnen een onderzoek een enorme diversiteit van wetenschappelijke achtergronden en expertises van de onderzoekers is. Bovendien vervullen onderzoekers ook nog een enorme verscheidenheid aan rollen en taken binnen een onderzoek. Ondanks deze

grote verscheidenheid hebben we drie generieke hoofdgroepen van functionele eisen voor SGs kunnen identificeren: data, berekening en samenwerking. Deze liggen aan de basis van elke biomedisch gerichte SG. Wij hebben gezien dat het verdelen van eisen in deze hoofdgroepen helpt bij de analyse van het gehele scala aan eisen. Dit verkort het iteratieve proces van het identificeren en categoriseren van vereisten zoals vaak te zien is bij de ontwikkeling van nieuwe SGs.

Wij hebben ons onderzoek voortgezet met het analyseren en organiseren van afwegingen over het ontwerp, de ontwikkeling en het functioneren van effectieve SGs. Onze belangrijkste bevinding is dat alle drie de functionele hoofdgroepen meegenomen moeten worden in het ontwerp. Dit betekent dat elke SG databronnen, berekeningsbronnen en samenwerkingsbronnen moet aanbieden als één geheel. Wanneer dit niet gedaan wordt zal dit resulteren in een minder effectieve SG die in de praktijk minder bruikbaar is. Een andere bevinding is dat het ontwerp van een SG ook flexibel genoeg moet zijn om mogelijke veranderingen te kunnen opvangen, bijvoorbeeld wanneer de e-Infrastructure bronnen of de functionele hoofdgroepen wijzigen door een verandering in onderzoeksmethodes. Om deze flexibiliteit aan te kunnen bieden is het cruciaal dat de functies van een SG worden ingesloten in goed gedefinieerde generieke diensten. Deze diensten moeten herbruikbaar zijn en moeten los van de SG te ontwikkelen zijn. Verder wordt de ontwikkeling van SGs bemoeilijkt door de grote verscheidenheid aan toepasbare technologieën. Het is daarom bevorderlijk om 'best practices' te delen en samen te werken bij het ontwikkelen van SGs. Dit zal ontwerpers en ontwikkelaars van SGs helpen om betere, gefundeerde beslissingen te nemen, en hierdoor sneller en goedkoper SGs te bouwen. De werking en uitvoering van een SGs is vooral gebaseerd op samenwerking tussen het wetenschappelijke domein, de SG en e-Infrastructure experts, die elk financiering nodig hebben. Verder werd duidelijk dat beslissingen die genomen zijn tijdens de ontwerp- en ontwikkelingsfase hun uitwerking hebben tijdens de uitvoeringsfase. Dit komt vooral naar voren wanneer zich fouten voordoen. Daarom is het belangrijk om tijdens het ontwikkelingsproces rekening te houden met de uiteindelijke uitvoering, en tevens mogelijkheden te bieden voor het oplossen van problemen. Tenslotte werd duidelijk dat de beoogde onderzoeksgemeenschap betrokken moet zijn tijdens elk proces: ontwerp, ontwikkeling, en uitvoer. Anders gezegd: de bouw van een SG moet gezien worden als een gemeenschappelijk proces, waar biomedische onderzoekers vanaf het begin bij betrokken dienen te worden. Zo wordt de effectiviteit verhoogd, en ontstaat er een levendig ecosysteem rondom de SGs.

Het volgende onderwerp wat wordt besproken in dit proefschrift is het aanbieden van duurzame diensten voor biomedische wetenschappers in de vorm van SGs. Wij hebben duurzaamheid geanalyseerd met behulp van de Business Model Canvas (BMC) methode. Hierbij bleek dat duurzaamheid twee kanten kent: kosten en baten. Bij SGs zijn de kosten vaak hoog, terwijl de baten laag zijn. Dit komt vooral door hoge kosten van ontwerp en ontwikkeling van SGs. Dit zijn beide complexe processen door de grote hoeveelheid aan domein-specifieke eisen van de onderzoeksgemeenschap. Verder zorgt de constante verandering van zowel eisen, e-Infrastructures, als technologieën er voor dat de ontwikkeling van een SG eigenlijk nooit stopt. Ook de uitvoer van een SG kent hoge kosten omdat een team van deskundigen nodig is uit: het domein, de SG, en de e-Infrastructure. Baten zijn vaak laag omdat de financiering van SG projecten gebeurt vanuit publieke middelen, waarbij de ontwikkeling gericht is op innovatie, terwijl de

uitvoeringsfase vaak niet in de begroting wordt meegenomen en dus ongedekt is. Ook bleek dat onderzoekers meestal niet in staat zijn om te betalen voor het gebruik van een SG via bijvoorbeeld pay-per-use modellen. Om de duurzaamheid van SGs te vergroten dienen kosten te worden verlaagd en baten verhoogd. Hoewel er in dit proefschrift een aantal mogelijkheden wordt genoemd om de duurzaamheid van SGs te vergroten zal verder onderzoek moeten uitwijzen of dit in de praktijk ook werkt. Vervolgstudies zijn dan ook een belangrijke stap om de duurzaamheid en houdbaarheid van toekomstige SGs veilig te stellen.

Als laatste onderwerp bespreken we waarom het organiseren van concepten en technologieën van SGs van belang is. Om aan de eisen van de drie functionele hoofdgroepen (data, berekening, en samenwerking) te voldoen, moet elke SG een set aan essentiële functies aanbieden. Wij hebben deze functies geïdentificeerd en georganiseerd in een referentie model dat wij de naam *Science Gateway Canvas (SGC)* hebben gegeven. Dit referentie model classificeert de functies in groepen en categorieën. De meest prominente functiegroep betreft 'management van middelen' (resource management); dit betreft management van: data, berekening, en samenwerking. Hiernaast zijn er ook nog universele managementfuncties die het management van individuele middelen behelzen, te weten: coördinatie, beveiliging, monitoren, en herkomst. Als laatste is er een categorie voor afleveringsfuncties die worden gebruikt door onderzoekers en externe programma's. Het ontwikkelde referentiemodel helpt nieuwkomers met het identificeren, begrijpen en plaatsen van de verschillende concepten en technologieën die gebruikt worden in SGs. Ook dient het als leidraad voor het identificeren van eisen en functies van SGs. Het ultieme doel van de SGC is het faciliteren van de constructie en oplevering van SGs voor biomedische wetenschappers, om hiermee hun onderzoekspraktijken te verbeteren. Uit de eerste feedback van de SG onderzoeksgemeenschap bleek dat de SGC bruikbaar wordt gevonden, maar dat er nog andere functiegroepen zijn die verder onderzocht moeten worden. Wij hopen dat het aanwenden van de SGC door de SG onderzoeksgemeenschap de kosten van het ontwerpen en ontwikkelen van SGs in de toekomst zal verminderen, waardoor het gebruik van SGs als nuttige gereedschap in biomedisch en ander wetenschappelijk onderzoek, algemeen geaccepteerd zal worden.

In dit proefschrift is meer inzicht verkregen in de essentiële fundamenten van SGs voor biomedisch onderzoek. Dit is gedaan door het organiseren van de eisen van de gebruikers, en ideeën over het ontwerp, ontwikkeling, uitvoering en 'houdbaar houden' van effectieve SGs. Verder hebben we een aantal succesvolle SGs ontwikkeld die gebruikt worden door een grote groep wetenschappers, waarmee we hun biomedische big-data-analyse op e-Infrastructures hebben gefaciliteerd. Als laatst stellen we een referentiemodel voor, dat de essentiële functies van SGs organiseert. Wij denken dat deze inspanningen leiden tot een betere ondersteuning van het ontwerp, ontwikkeling, uitvoering, 'houdbaar houden', én gebruik van SGs voor biomedisch onderzoek.

Acknowledgments

First and foremost I would like to thank my promoter and co-promoter Antoine van Kampen and Sílvia Olabarriaga. Both of you always provided me with your indispensable advice and feedback that not only guided me throughout this research, but also taught me invaluable lessons. Sílvia, I have learned a lot particularly from you as my daily advisor. And most importantly, you have not been only my advisor, but also a very good friend. Thank you for your support, encouragement, and all the essential skills you taught me. Antoine, your critical advice always helped me to stay on track, and your unique perspective taught me new points of view, thank you!

I would also like to take this opportunity to thank the anonymous reviewers of the articles published in this thesis who helped me to improve the quality of my research by their critical feedback. Moreover, I thank the members of the doctorate committee for accepting to evaluate this thesis.

The research presented in this thesis was not possible without various internal and external collaborations. I would like to thank the present and past members of the AMC e-Science group led by Sílvia for their various contributions, especially: Mahdi Jaghoori [MJ: thank you for sharing your unique views on science and programming with me, and for the proofreading and being my paranymph.], Allard van Altena [AvA: thank you for translating the summary to Dutch, and good luck with your doctorate.], Juan Luis Font [JLF: I always enjoy chatting with you, you have one the best senses of humor, thank you for cheering me up.], Mark Santcroos [MS: I miss our forty-two sessions, I learned a lot from our brainstorms, thank you, also thanks for being my paranymph, and good luck with your doctorate.], Jorrit Posthuma [JP: I really enjoyed and learned a lot from our programmings together, thank you.], Vladimir Korkhov, Hurng-Chun Lee, Yassene Mohammed, Ammar Benabdelkader, Souley Madougou, Sara Ramezani, Jalmar Teeuw, Gerbrand Spaans, Mostapha al Mourabit, Kyriacos Neocleous, Evert Mouw, Sytse Geldermalsen, Daniël Westerbeek, and Carsten Byrman.

I would also like to thank the present and past member of the AMC bioinformatics laboratory led by Antoine, especially: Barbera van Schaik, Angela Luyf, Aldo Jongejan, Marcel Willemsen [BvS, AL, AJ, MW: thank you all for your contributions to the e-Bio-Infra gateway.], Umesh Nandal [UN: thank you for teaching me all those yoga moves, and good luck with your thesis.], Gerbert Jansen [GJ: thank you for proofreading the Dutch summary.], Miranda Stobbe, Perry Moerland, Mia Pras, Dicle Hasdemir, Polina

The names presented in this chapter are in no particular order.

Reshetova, Eelke van der Horst, Andrew Gibson, Herman Sontrop, Ishtiaq Ahmad, and Els Natzijl-Visser.

I would also like to thank the present and past members of the AMC Brain Imaging Center (BIC), especially: Matthan Caan [MC: thank you for being such a great collaborator.], Guido van Wingen, Paul Groot, Jordi Huguet, Laura Koenders, and Marise Machielsen.

I would also like to thank the present and past members of the department of Clinical Epidemiology, Biostatistics, and Bioinformatics (Dutch: Klinische Epidemiologie, Biostatistiek en Bio-informatica) (KEBB) of the AMC, especially: Koos Zwinderman, Jammbe Musoro, Wouter Ouwerkerk, Mare van Barreveld, Erik van Iperen, Mareen Datema, Daniel Korevaar, Maurice de Ronde, Yu Han (Sapphire), Simone Aufiera [JM, WO, MvB, EvI, MD, DK, MdR, YH, SA: good luck all with your doctorates.], Michel Hof, Parvin Tajik [MH, PT: thank you for the useful tips about the thesis.], Raha Pazoki, Jérémy Cohen, Rosa Sloot, Annefloor van Enst, Eleonor Ochodo, Teodora Radonic, Iris Kolder, Esther van de Glind, Inge Stegeman, Rob Scholten, René Spijker [RS: thank you for the literature search methods.], Ronald Geskus, Miranda Langendam, Michael Tanck, Susanne de Rooij, Sara-Joan Pinto, Hanni Spitteler, Annette van der Graaff, Gré de Vries, Helen Ottenhoff, and Alexander Popma.

I would also like to thank other AMC colleagues, especially: Renan Sales Barros, Eva Hartkamp, Marjanne Meinsma, Cootje Kusters, Ed Hull, Mohammad Jazaeri, Mohammad Motazacker, Majid Tarahomi, Ramin Sarrami, Rossella Trezza, Leonie de Vries, and Liesbeth de Vries [RSB, RT, LdV, LdV: good luck all with your doctorates.].

I would also like to thank Boy Menist, Peter Sloot, Shantenu Jha, Shahin Rouhani, Coen Schrijvers, Natalie Danezi, Jeroen van Duffelen, Hans van den Berg, Joyce Nijkamp, Jeroen Roodhart, Jan Just Keijser, Tom Visser, and Piter de Boer for their advice and various contributions.

I would also like to thank the members of Amsterdam Center for Entrepreneurship (ACE), European Grid Infrastructure (EGI), and SURFsara for their various contributions.

I would also like to acknowledge COMMIT/ (especially P24 and P23), BiG Grid, and SCI-BUS projects that funded me or my collaborators.

I would also like to thank those who have contributed to the following (open-source) software projects: VBrowser, MOTEUR, DIANE, DIRAC, Spring Framework, Apache Foundation, Glassfish, Liferay, MySQL, WS-PGRADE/gUSE, GitHub, Redmine, and many other projects I benefited from.

Every person who I came into contact with in my life, for sure taught me something one way or another. I truly wish I could show my appreciation by naming every one of them here, but unfortunately that is not possible. Nevertheless, I would like to take this opportunity to thank all of my family members, friends, and past and present colleagues, classmates, teachers, and neighbors. *Thank you all.*

"The greatest gift of life is friendship, and I have received it." says Hubert H. Humphrey. I have received this gift from: Mark & Kalinka, Sílvia & Ruud, Jasleen S., Mohammad Reza S., Mohsen & Betisa, Dawood & Sanaz, Danesh M., Hadi & Mehrnoosh, Ali & Sanaz., Pouyan S., Faraz F., Heiko & Filza, Mike & Karen, Ahmad S., Masoud & Samar, Hossein & Maryam, Amir M., Hamed H., Mohammad P., Saeed T., Hamze & Raheleh, Behnam E., Arman K., Adel G., Danial K. [DK: thank you for designing the cover.], Vahid & Sara, Mohammad Amin & Fatemeh, Abbas M., Naser & Sareh, Saeed

& Andisheh, Ali & Azadeh, Mojtaba & Toktam, Mehdi & Marziyeh, Hodjat & Marziyeh, Mahdi & Samira, Naser & Aylar, Mahdi & Mahdiyeh, Mohammad & Fahimeh, Masoud & Hoda, Amir Hossein & Ayrin, Micheal & Narges, Rory & Behnaz, Hendrik & Rokhsareh, Maryam B., Hassan Ch., Hamid Reza & Marziyeh, Behrooz & Shima, Afshin & Fatemeh, Farzin & Mahsa, Mostafa & Samira, Mohammad & Mahsa, Amin & Parisa, Hadi & Parvin, Navid & Niloofar, Ali & Bita, Abolfazl A., and many more.

Pieter en Monique, ik kan nu goed genoeg in het Nederlands communiceren dankzij jullie. Ik zie jullie echt als mijn eigen familie. Dank jullie wel.

My dear mother, my dear father, thank you both for everything, for all that you have taught me. My dear parents, parents-in-law, sisters, brothers-in-law, sisters-in-law, nieces, and nephew, thank you all for your endless support and love.

My dear Samin, you brought light and joy into our lives, your smile gives me energy, and your eagerness to explore motivates me to continue learning. Thank you sweetie.

My darling Narges, your unconditional support and endless love gives me energy and purpose in life. Thank you sweetheart.

Shayan Shahand
July 2015, Amsterdam

پدر و مادر عزیزم، این پایان نامه ی دکترا هدیه ی ناقابلی است که از طرف فرزند کوچکتان به شما تقدیم می گردد. از شما به خاطر همه آن هایی که به من آموختید، به خاطر محبت و کمک و پشتیبانی بی انتهایتان که همواره باعث دلگرمی من بوده است، بی نهایت سپاسگزارم. پدر و مادر سببی عزیزم از محبت و کمک های همیشگی شما هم سپاسگزارم. خواهران و برادران نسبی و سببی عزیزم، شکوه و حمیدرضا، شیوا و شاهین، شایسته و مسیح، هاجر و مهدی، حامد و زهرا دخت، و مجید و شهین از محبت و کمک های همیشگی شما هم سپاسگزارم. مونا، روشا، آبتین، فاطمه، حورا، و بهار عزیز حضور شما هم همواره باعث شادی و دلگرمی من بوده است.

بر خود لازم میدانم که از سایر اقوام نسبی و سببی، پدربزرگ ها، مادربزرگ ها، خاله ها، عمه ها، دایی ها، و عموهای عزیز، و فرزندان شان نیز بابت محبت و لطف همیشگی شان تشکر کنم.

ثمین عزیزم، با ورودت شادی و رنگ تازه ای به زندگی ما بخشیدی، لبخندت باعث دلگرمی من در زمان نوشتن این پایان نامه بود. ازت ممنونم.

نرگس عزیزم، حمایت، محبت و کمک همیشگی و بی پایانت همواره اطمینان بخش و باعث دلگرمی و شادی من بوده است. ازت سپاسگزارم.

شایان شهند
تیر ۱۳۹۴، آمستردام

PhD Portfolio

Name of PhD student: Shayan Shahand
PhD period: Oct 2011 – Oct 2015
Name of PhD supervisor: Prof. dr. A. H. C. van Kampen
Name of PhD co-supervisor: Dr. S. D. Olabarriaga
Total ECTS: 42

1. PhD training	Year	ECTS
General courses		5.6
AMC World of Science (AMC graduate school)	2011	0.7
Oral Presentation in English (AMC graduate school)	2012	0.8
Project Management (AMC graduate school)	2014	0.6
Explore Program (ACE)	2014	2
Scientific Writing in English for Publication (AMC graduate school)	2015	1.5
Specific courses		7.6
Introduction to Bioinformatics (AMC graduate school)	2011	1.2
Managing Life Science Information (NBIC)	2011	2
DNA Technology (AMC graduate school)	2012	1.2
e-Science (AMC graduate school)	2014	1.2
Functional Programming Principles in Scala (EPFL via Coursera)	2014	2

Continued on the next page

Continued from the previous page	Year	ECTS
Seminars, workshops and master classes		9.1
Master class on fMRI and SPECT	2011	0.9
International Workshop on Science Gateways for Life Sciences (IWSG-Life)	2012	0.8
The São Paulo School of Advanced Science on e-Science for BioEnergy Research (SPSAS)	2012	1.5
The Latin American e-Science workshop (LATAM): Turning Data into Insight	2013	1.2
International Workshop on Science Gateways (IWSG)	2013	0.8
SURFsara Life Science e-Infrastructure Workshop	2013	0.2
SURFsara Data & Computing Infrastructure Event	2014	0.4
SURFsara Life Science e-Infrastructure Workshop	2014	0.2
International Summer School on Trends in Computing	2014	1.5
Medical Informatics Symposium: Healthcare: The Big Data Challenge	2014	0.4
International Workshop on Science Gateways (IWSG)	2015	0.6
International Workshop on the Science of Cyberinfrastructure: Research, Experience, Applications and Models (SCREAM)	2015	0.6
Presentations		5.0
"Gateways that Cover Broader Grid User Spectrum". S. Shahand, A. H. C. Van Kampen and S. D. Olabarriaga. Satellite presentation in IWSG-Life.	2011	0.5
"Many versions of a web interface for data analysis on the Dutch e-Science Grid". S. Shahand, C. Byrman, M. Santcroos, A. H. C. van Kampen and S. D. Olabarriaga. Satellite presentation in P-GRADE Portal User COmmunity WOrkshop (PUCOWO).	2011	0.5
"Front-ends to Biomedical Data Analysis on Grids". S. Shahand, M. Santcroos, Y. Mohammed, V. Korkhov, A. C. M. Luyf, A. H. C. van Kampen and S. D. Olabarriaga. in HealthGrid Conference.	2011	0.5
"Integrated Support for Neuroscience Research: from Study Design to Publication". S. Shahand, M. W. A. Caan, A. H. C. van Kampen and S. D. Olabarriaga. in HealthGrid Conference.	2012	0.5
"A Data-Centric Science Gateway for Computational Neuroscience". S. Shahand, A. Benabdelkader, J. Huguet, M. Jaghouri, M. Santcroos, M. al Mourabit, P. F. C. Groot, M. W. A. Caan, A. H. C. van Kampen and S. D. Olabarriaga. in IWSG.	2013	0.5

Continued on the next page

Continued from the previous page	Year	ECTS
"e-BioInfra Gateway to Facilitate Large-scale Biomedical Data Analysis". S. Shahand, A. H. C. van Kampen and S. D. Olabarriaga. In SURFsara Life Science e-Infrastructure Workshop	2013	0.5
"Science Gateways for Big Biomedical Data Analysis". S. Shahand and S. D. Olabarriaga. In the Medical Informatics Symposium: Healthcare: The Big Data Challenge.	2014	0.5
"Initial steps in analyzing science gateways sustainability through business model canvas: A use case for the computational neuroscience gateway". S. Shahand and S. D. Olabarriaga. Satellite presentation in the 9th Workshop on the Gateway Computing Environments (GCE).	2014	0.5
"Science Gateway Canvas". S. Shahand, A. H. C. van Kampen and S. D. Olabarriaga. in IWSG.	2015	0.5
"Science Gateway Canvas: A business reference model for Science Gateways". S. Shahand, A. H. C. van Kampen and S. D. Olabarriaga. in the 1st Workshop on the Science of Cyberinfrastructure: Research, Experience, Applications and Models (SCREAM).	2015	0.5
(Inter)national conferences		3.4
The Healthgrid Conference	2011	0.8
The Healthgrid Conference	2012	0.8
The Dutch ICT-Research in the Netherlands (ICT.OPEN)	2013	0.5
The Dutch ICT-Research in the Netherlands (ICT.OPEN)	2015	0.5
The 24th International ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC)	2015	0.8
Other		6.8
Bioinformatics Laboratory (bi)weekly seminars	2011:4	2.0
KEBB (bi)weekly seminars	2011:5	1.2
BIC (bi)weekly seminars	2012:4	0.5
Reading Club	2012:4	1.0
COMMIT Day: Kick-Off meeting		0.5
Demo at the ICT.OPEN	2013	0.2
COMMIT Day: COMMIT/TED to you	2013	0.5
COMMIT Day: The Big Future of Data	2014	0.5
Demo at the COMMIT event: The Big Future of Data	2014	0.2

Continued on the next page

Continued from the previous page	Year	ECTS
Demo at the ICT.OPEN	2015	0.2

2. Teaching	Year	ECTS
Lecturing		2.5
Guest lecturer for Introduction to Bioinformatics course (AMC graduate school)	2012	0.5
Guest lecturer for Advanced Cognitive Neurobiology & Clinical Neurophysiology course (SILS)	2013	0.5
Guest lecturer for e-Science course (AMC graduate school)	2014	0.5
Guest lecturer for Biomedical Information Systems Engineering course (MIK master)	2014	0.5
Guest lecturer for Databases and Computer Networks course (MIK bachelor)	2015	0.5
Tutoring, Mentoring		2.0
Jorrit Posthuma, design and implementation of a science gateway for computational neuroscience (Rosemary)	2014	1.0
Allard van Altena, traineeship MIK Master AMC: User-centered design and implementation of a science gateway for In Vitro Fertilization (IVF) data management	2015	1.0

3. Parameters of Esteem	Year
Grants	
Full travel grant for attending the São Paulo School of Advanced Science on e-Science for BioEnergy Research (SPSAS)	2012
Full travel grant for attending the Latin American e-Science workshop (LATAM)	2013
Co-applicant of a 92000€ proposal granted by the High Performance Computing and Networking (HPCN) Fund of UvA	2015

Publications

Peer reviewed full papers

*Publications marked with * are used as the body of this thesis.*

1. * **S. Shahand**, A. H. C. van Kampen, and S. D. Olabarriaga. "Science Gateway Canvas: A business reference model for Science Gateways". In *Proceedings of the 1st Workshop on the Science of Cyberinfrastructure: Research, Experience, Applications and Models*, SCREAM'15, pages 45–52, Portland, OR, USA, 2015. DOI: 10.1145/2753524.2753527 (Used in Chapter 7)
2. M. M. Jaghoori, **S. Shahand**, and S. D. Olabarriaga. "Processing Manager for Science Gateways". In *Proceedings of the International Workshop on Science Gateways*, IWSG'15, Budapest, Hungary, 2015. DOI: 10.1109/IWSG.2015.9
3. M. W. A. Caan, J. Teeuw, **S. Shahand**, M. M. Jaghoori, J. Huguet, A. van Altena, S. D. Olabarriaga. "A Neuroscience Gateway for Handling and Processing Population Imaging Studies". In *Proceedings of the 1st MICCAI 2015 Workshop on Management and Processing of images for Population Imaging, MICCAI-MAPPING2015*, C. Barillot, M. Dojat, D. Kennedy and W. Niessen (Eds), pp.15–22, 2015.
4. * **S. Shahand**, J. van Duffelen, and S. D. Olabarriaga. "Reflections on Science Gateways Sustainability Through the Business Model Canvas: Case Study of a Neuroscience Gateway". *Concurrency and Computation: Practice and Experience*, to appear, 2015. DOI: 10.1002/cpe.3524 (Used in Chapter 6)
5. * **S. Shahand**, A. Benabdelkader, M. M. Jaghoori, M. al Mourabit, J. Huguet, M. W. A. Caan, A. H. C. van Kampen, and S. D. Olabarriaga. "A data-centric neuroscience gateway: design, implementation, and experiences". *Concurrency and Computation: Practice and Experience*, 27(2):489–506, 2015. DOI: 10.1002/cpe.3281 (Used in Chapter 5)
6. **S. Shahand** and S. D. Olabarriaga. "Initial steps in analyzing science gateways sustainability through business model canvas: A use case for the computational neuroscience gateway". In *Proceedings of the 9th Workshop on the Gateway Computing Environments*, GCE'14, pages 5–8, Piscataway, NJ, USA, 2014. IEEE Press. DOI: 10.1109/GCE.2014.16
7. **S. Shahand**, A. Benabdelkader, J. Huguet, M. M. Jaghoori, M. Santcroos, M. al Mourabit, P. F. C. Groot, M. W. A. Caan, A. H. C. van Kampen, and S. D. Olabarriaga.

"A data-centric science gateway for computational neuroscience". In *Proceedings of the 5th International Workshop on Science Gateways*, IWSG'13, Zurich, Switzerland, June 2013. URL: ceur-ws.org/Vol-993/paper12.pdf

8. S. Madougou, **S. Shahand**, M. Santcroos, B. D. C. van Schaik, A. Benabdelkader, A. H. C. van Kampen, and S. D. Olabarriaga. "Characterizing workflow-based activity on a production e-infrastructure using provenance data". *Future Generation Computer Systems*, 29(8):1931–1942, 2013. DOI: 10.1016/j.future.2013.04.019

9. M. Santcroos, B. D. C. van Schaik, **S. Shahand**, S. D. Olabarriaga, A. Luckow, and S. Jha. "Exploring dynamic enactment of scientific workflows using pilot-abstractions". In *Proceedings of the 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, CCGrid'13, pages 311–318, 2013. DOI: 10.1109/CCGrid.2013-.17

10. * **S. Shahand**, M. Santcroos, A. H. C. van Kampen, and S. D. Olabarriaga. "A grid-enabled gateway for biomedical data analysis". *Journal of Grid Computing*, 10(4):725–742, 2012. DOI: 10.1007/s10723-012-9233-4 (Used in Chapter 4)

11. * **S. Shahand**, M. W. A. Caan, A. H. C. van Kampen, and S. D. Olabarriaga. "Integrated support for neuroscience research: from study design to publication". In *Studies in Health Technology and Informatics*, volume 175, pages 195–204, 2012. DOI: 10.3233/978-1-61499-054-3-195 (Used in Chapter 3)

12. S. Madougou, M. Santcroos, A. Benabdelkader, B. D. C. van Schaik, **S. Shahand**, V. Korkhov, A. H. C. van Kampen, and S. D. Olabarriaga. "Provenance for distributed biomedical workflow execution". In *Studies in Health Technology and Informatics*, volume 175, pages 91–100, 2012. DOI: 10.3233/978-1-61499-054-3-91

13. M. W. A. Caan, **S. Shahand**, F. Vos, A. H. C. van Kampen, and S. D. Olabarriaga. "Evolution of grid-based services for diffusion tensor image analysis". *Future Generation Computer Systems*, 28(8):1194–1204, 2012. DOI: 10.1016/j.future.2012.03.007

14. Y. Mohammed, **S. Shahand**, V. Korkhov, A. C. M. Luyf, B. D. C. van Schaik, M. W. A. Caan, A. H. C. van Kampen, M. Palmblad, and S. D. Olabarriaga. "Data decomposition in biomedical e-science applications". In *Proceedings of the Seventh International Conference on e-Science Workshops*, eScienceW'11, pages 158–165, Stockholm, Sweden, Dec. 2011. DOI: 10.1109/eScienceW.2011.7

15. * **S. Shahand**, M. Santcroos, Y. Mohammed, V. Korkhov, A. C. M. Luyf, A. H. C. van Kampen, and S. D. Olabarriaga. "Front-ends to Biomedical Data Analysis on Grids". In *Proceedings of the HealthGrid Conference*, Bristol, UK, 2011. DOI: 10.6084/m9.figshare.1372471 (Used in Chapter 2)

Other

16. EGI Science Gateway Virtual Team. "Science Gateway Primer". In *European Grid Infrastructure (EGI) document repository*, 2013. URL: documents.egi.eu/document/1463

17. **S. Shahand**, M. M. Jaghoori, A. Benabdelkader, J. L. Font-Calvo, J. Huguet, M. W. A. Caan, A. H. C. van Kampen, and S. D. Olabarriaga. "Computational neuroscience

gateway: A science gateway based on the WS-PGRADE/gUSE". In *P. Kacsuk, editor, Science Gateways for Distributed Computing Infrastructures*, pages 139–149. Springer International Publishing, 2014. DOI: 10.1007/978-3-319-11268-8_10

Contributing Authors

Ammar Benabdelkader

Sharp Systems, Amstelveen, the Netherlands.

Matthan W. A. Caan

Department of Radiology and Brain Imaging Center, Academic Medical Center, University of Amsterdam, Amsterdam, The Netherlands.

Jeroen van Duffelen

Amsterdam Center for Entrepreneurship, Amsterdam, The Netherlands

Jordi Huguet

Department of Radiology and Brain Imaging Center, Academic Medical Center, University of Amsterdam, Amsterdam, The Netherlands.

Mohammad Mahdi Jaghoori

E-Science group, Department of Clinical Epidemiology, Biostatistics, & Bioinformatics, Academic Medical Center, University of Amsterdam, Amsterdam, The Netherlands.

Antoine H. C. van Kampen

Bioinformatics Laboratory, Department of Clinical Epidemiology, Biostatistics, & Bioinformatics, Academic Medical Center, University of Amsterdam, Amsterdam, The Netherlands.
Biosystems Data Analysis group, Swammerdam Institute for Life Science, University of Amsterdam, Amsterdam, The Netherlands.

Vladimir Korkhov

Department of Computer Modeling and Multiprocessor Systems, Faculty of Applied Mathematics and Control Processes, St. Petersburg State University, St. Petersburg, Russia.

Angela C. M. Luyf

Bioinformatics Laboratory, Department of Clinical Epidemiology, Biostatistics, & Bioinformatics, Academic Medical Center, University of Amsterdam, Amsterdam, The Netherlands.

Yassene Mohammed

Center for Proteomics and Metabolomics, Leiden University Medical Center, Leiden, The Netherlands.
Proteomics Center, University of Victoria, Victoria, BC, Canada.

Mostapha al Mourabit

Unknown.

Sílvia D. Olabarriaga

E-Science group, Department of Clinical Epidemiology, Biostatistics, & Bioinformatics, Academic Medical Center, University of Amsterdam, Amsterdam, The Netherlands.

Mark Santcroos

RADICAL group, Department of Electrical and Computer Engineering, Rutgers University, New Brunswick, NJ, USA.