A Secure and Flexible Data Infrastructure for the VPH-Share Community

Siegfried Benkner*, Chris Borckholder*, Marian Bubak†, Yuriy Kaniovskyi*
Piotr Nowakowski†, Dario Ruiz Lopez§ and Steven Wood‡

*University of Vienna, Research Group Scientific Computing, Vienna, Austria
†Institute of Computer Science / Cyfronet AGH University of Science and Technology, Krakow, Poland
‡Scientific Computing & Informatics, Sheffield Teaching Hospitals NHS Foundation Trust, Sheffield, UK
§Atos Origin S.A.E. / IT Research, Madrid, Spain

***Abstract*—The European VPH-Share project develops a comprehensive service framework with the objective of sharing clinical data, information, models and workflows focusing on the analysis of the human physiopathology within the Virtual Physiological Human (VPH) community. The project envisions an extensive and dynamic data infrastructure built on top of a secure hybrid Cloud environment. This paper presents the data service provisioning framework that builds up the data infrastructure, focusing on the deployment of data integration services in the hybrid Cloud, the associated mechanism for securing access to patient-specific datasets, and performance results for different deployment scenarios relevant within the scope of the project.**

I. Introduction

The Virtual Physiological Human Initiative (VPH-I) develops a methodological and technological framework for researching pathological and physiological processes in the human body. As part of this initiative, the ongoing VPH-Share project establishes a novel data infrastructure that will offer researchers comprehensive tools for collecting, structuring, disseminating and analyzing data originating from clinical institutions, with the ultimate goal of improving the chances of accurate diagnosis and successful treatment.

Datasets exposed by the infrastructure are typically multi-scale and multi-modal in both the qualitative and quantitative dimension. The scope of coverage ranges from micro to macro biology, including patient data, medical imagery and biomedical records. Aside from the technical challenges of exposing and accessing numerous, distributed and potentially heterogeneous dataset sources, a number of legal and ethical requirements must be met. These requirements mainly concern the privacy of patient-specific clinical data. The data has to be anonymized so it cannot be traced back to the patient, while access has to be restricted to authorized members of the VPH community.

The VPH-Share Data Publication Suite [1], developed in context of the Dataset Service Environment (DSE), offers tools to de-identify, semantically annotate and expose dataset sources. The Java-based DSE is built upon the Vienna Cloud Environment [2] and the Open Grid Services Architecture Data Access and Integration (OGSA-DAI) [3] framework. It provides two types of data integration services: Dataset Services used to expose heterogeneous dataset sources, and Virtual Dataset Services used to mediate data across several Dataset Services. Virtual Dataset Services enable the creation of specialized data spaces, where associated data are unified through a global virtual schema. Whereas Dataset Services are preferably deployed in the vicinity of the dataset source, either on the same server or network, Virtual Dataset Services are provisioned within a Cloud environment for scalable data mediation across the VPH-Share platform. To safeguard data and applications, the platform implements a custom-built service security framework that allows application services to be wrapped and its end-to-end communications secured in a generic and transparent manner.

The VPH-Share Cloud that hosts the services of the platform – called Atmosphere [4] – follows a hybrid architecture and merges the private Clouds situated in Krakow (Poland), Sheffield (UK) and Vienna (Austria). These Cloud systems are contributed by the corresponding partners of the project. In addition, the public Europe-based Amazon EC2 [5] in Ireland is used to extend the capabilities of the Atmosphere. The different Clouds provide the Atmosphere with a high flexibility and scalability in terms of its computing resources.

Following a broad overview of the hybrid Cloud environment of the VPH-Share project in Section II, which also briefly touches on the services that build up the data infrastructure, the paper discusses two central aspects of the project: security and performance. Section III describes the security measures that are implemented for restricting access to the critical functions of the research platform, while Sections IV and V detail the testbed for the data infrastructure and provide performance results based on different Cloud sites of the Atmosphere.

II. VPH-Share Cloud Environment

The Dataset Services are deployed on the hybrid Atmosphere Cloud. The Cloud platform enables groups of users to gain authorized access to a variety of computational and data services deployed on distributed hardware resources. Most of the technologies underpinning the platform are exclusively implemented as so-called Atomic Services – applications or other essential elements of platform logic, which encapsulate the data they operate on and provide secure interfaces to access them. In this sense the function of the Atmosphere provides a

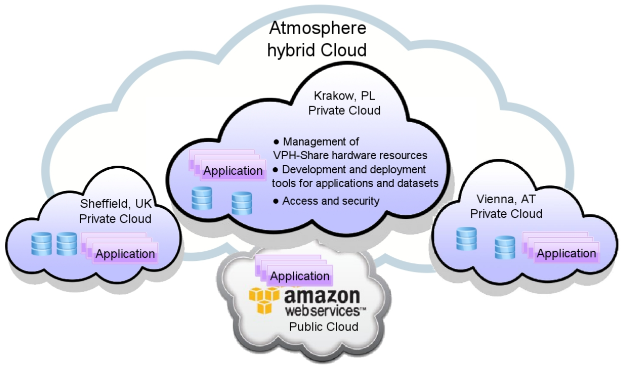

Fig. 1. Schematic overview of the Atmosphere Hybrid Cloud plaform.

persistent hosting infrastructure on which to deploy, instantiate, access and manage VPH services (which are understood as applications, or components thereof), fulfilling the specific needs of researchers. In addition to the application hosting, the goal of the Atmosphere is to provide end users with control interfaces, enabling them to interact with the exposed tools in a secure and convenient manner. With the help of Atmosphere, these tools are meant to be exposed to a wider community of users and potential collaborators.

The framework divides users into application providers (developers), domain scientists and administrators. The Cloud platform covers the entire application development life-cycle, from inception to scalable exploitation, assisting each participating class of users at each step of the design, deployment and enactment process. At the same time, the Atmosphere forms a bridge between the world of Cloud middleware services, which are typically difficult to access for inexperienced users, and the familiar OS environments, in which standalone scientific applications are deployed and accessed.

The platform is implemented as a series of modules that operate together on a dedicated host (the core host) and Cloud middleware stacks required for accessing and managing physical resources. The platform also includes an internal registry containing all metadata pertaining to Atomic Services and their instances, with embeddable UI extensions that can operate in a standalone mode or be imported into a portal. Provided they share a common registry, all core components of Atomic Services can be instantiated multiple times and deployed on an external host. This feature offers the platform the required scalability, enabling it to grow with the number of instantiated services and clients.

The Atmosphere is able to merge and use multiple Clouds. By doing so, it provides an elastic hosting infrastructure that offers numerous compute, storage and network resources visualized in Fig. 1. The exposed API endpoints of the individual Cloud sites are used to deploy services and provide statistics of the current resource load. This allows Atmosphere to monitor and manage system load. Since Atmosphere is compatible with the Amazon EC2 Cloud, it can use the infrastructure provided by this large hosting platform, as illustrated in Section V.

A. *Dataset Services*

Dataset Services that expose datasets and build up the data infrastructure are provided within an integrated service framework [6]. The framework offers SOAP and RESTful interfaces based on the Apache CXF Web Service Framework, and provides access to both relational and semantic datasets. Two distinct classes of services are defined within the scope of the data infrastructure. Dataset Services can expose heterogeneous dataset sources, e.g., relational DBMS or their RDF-triple-store counterparts, whereas Virtual Dataset Services transparently integrate multiple Dataset Services unifying and exposing their datasets as a single virtual one. The interface for both service types is identical, making operation calls to them similar. The querying language is dependent on the resource that is requested. Relational resources are queried via SQL, while RDF resources are queried via SPARQL. Virtual Dataset Services implements the mediation engine, built on top of OGSA-DAI's Distributed Query Processing (DQP) [7] extension for querying distributed relational datasets. Virtual Dataset Services also include a semantic processor for federated SPARQL queries based on [8] to allow a unified access distributed semantic datasets. The semantic representation of the data enables the use of convenient knowledge discovery mechanisms, while mediated datasets provide a domain specific unification of the data.

The concept of Atomic Services envisions a collection of many virtual appliances – preconfigured virtual images – that constitute the toolbox for working with applications via the Cloud. In the context of Atomic (Virtual) Dataset Services, this provides the user with the ability to utilize the capabilities, security and the flexibility offered by Atmosphere to its full extent.

III. Secure Data Access in the VPH Cloud

VPH-Share is protected from unauthorized access by a security framework that incorporates a number of security features as a wrapper around the Atomic Services. To grasp the scope of the requirements of the security framework, it is important to note that VPH-Share integrates a dynamic set of heterogeneous applications, each of them having own security requirements. The security framework cannot address all security requirements specific to each integrated application, especially since new applications with new requirements can be added dynamically. The VPH-Share security framework resolves the security issues resulting from the integration of these applications into a common and publicly accessible framework, leaving the responsibility of fulfilling application-specific security requirements to service developers.

The VPH-Share portal uses the OpenID authentication standard for authorizing users. It allows users to log-in using the credentials obtained from a OpenID Identity Provider, which is BiomedTown [9]. The biomedical research community of BiomedTown gathers participating users such as data providers, developers and clinicians while forming a trust model between them and the virtual organization of VPH-Share. As such, BiomedTown is also responsible for managing

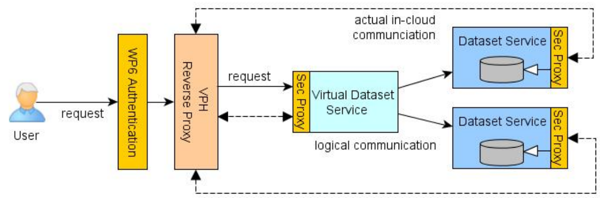

Fig. 2. Cloud communication on a request to a Virtual Dataset Service. In-Cloud communication of participating data and evaluation nodes is routed through the reverse proxy, while the security proxy wraps individual Dataset Services.

VPH-Share user accounts. A login request or application call is followed by a query to BiomedTown, where the portal either grants or confirms access to the platform.

Securing individual Atomic Services involves managing secure communications using two types of proxies. A reverse proxy of the Atmosphere maps and forwards incoming requests from public IP address of Atomic Service instances to their private IP addresses. It does not expose the private IP address of the Atomic Service, leaving it hidden in the network. The reverse proxy operates in tandem with a security proxy preinstalled on every Atomic Service instance, as depicted in Fig. 2. This second component safeguards each VPH-Share relevant service through authorization procedures. The security proxy is configured using XML configuration files. Each instance retrieves its configuration files from the Atmosphere Cloud API and upgrades the local configuration of the security proxy. The configuration files are cached locally in each Atomic Service, but the configuration is updated periodically.

In addition to access restrictions, the security framework has to protect external communications conducted with the services. This is achieved by encrypting all communications between service entities and clients. The encryption takes place on the HTTP level by using the SSL protocol over HTTP (HTTPS). This implies that the reverse proxy of the Atmosphere decrypts the message and forwards the message, encrypting it again prior to contacting the service recipient. Dealing with SSL can be easily performed by configuring the HTTP SSL module of the Nginx server used by the Atmosphere.

Atmosphere provides tools for declaring a number of endpoints (TCP- or HTTP-based) through the VPH-Share portal. Atomic Services then use the declared endpoints for secure external communications. The endpoints can be searched with tools provided by the VPH-Share portal and used to build application workflows involving multiple services. When dealing with Atomic Service instances, the security proxy intercepts all HTTP traffic by listening to endpoint ports configured in the Atmosphere. Two tasks are performed at a service request: the decryption of messages and checking the authorization. The authorization check of a given service is based on the security attributes of the user provided along with the request into the VPH-Share platform. The proxy inspects the HTTP header attributes of the incoming request and validates the integrity of the message by analyzing the signature of the field containing it. If the request is valid, the HTTP message is redirected to a local host address of the Atomic Service. Once the service produces a result, it is encrypted on its way back to the reverse proxy that delivers it to the requester. The presented solution allows any HTTP-based services, including REST- and SOAP-based services, to be secured in a transparent manner.

A. *Security-aware Dataset Services*

Following the implementation of the security framework for the Atmosphere, Atomic Services operate in a protected environment. While each instance is a secured encapsulation of service functionality, the question arises how services can interoperate, while still being able to uphold the authentication of the user. This is achieved through delegating the authentication information across the involved services. In the following, we differentiate security measures according to the service type.

Dataset Services act as fine-grained data nodes that process requests directed to data information systems they wrap, e.g. passing a SQL query to the actual dataset source through the JDBC interface. These services do not require delegation of authentication as they do not consume additional Atomic Services, but rather the dataset source contained within the same server or network.

Virtual Dataset Services act as data mediators. In contrast to Dataset Services, they usually do not process requests from a locally available dataset source but rely on interoperation with other Dataset Services in the VPH Cloud for aggregating data received from the actual datasets. To enable communications through the security framework a Virtual Dataset Service "impersonates" the user for every adjacent operation call to the underlying Dataset Services. For this purpose a delegation of the user authentication information – in form of a security token – is accomplished.

While the data itself is streamed, the processing of a distributed query is realized through asynchronous communication between the participating data and mediation nodes. The credentials provided by the BiomedTown security token are to be shared with each involved node for a two-way communication between them. At the same time, credentials must be associated with a single request to maintain a user session and preclude leaking critical information to a false recipient. The request will fail altogether should the user not have permissions to access all datasets involved in the distributed dataset query, as it is assumed that the partial delivery of datasets may cause inconsistencies within the data.

Security mechanisms are mostly transparent to a Dataset Service, however, handling the security token while processing a distributed query in Virtual Dataset Service is a crosscutting concern for both service types and requires modifications in the OGSA-DAI middleware. The general flow of the delegation of credentials is depicted in Fig. 3.

Should the security proxy grant the user access to the Dataset Service, then the request is forwarded to the corresponding service frontend (either SOAP or HTTP). For this

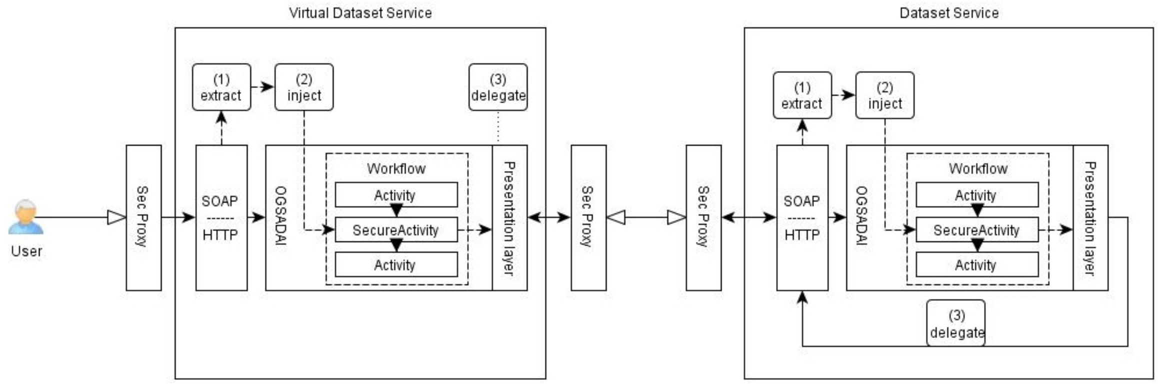

Fig. 3. Security token delegation while processing a request in OGSA-DAI. (1) The token is extracted by the accessed frontend, (2) the OGSA-DAI engine injects the token into secured activities, (3) activities pass the received token to the presentation layer to access remote services.

service type this alone is sufficient to authorize the request execution, as the validation of the provided security token is handled by the security proxy.

In the case of a Virtual Dataset Service, the frontend is responsible for handing the security token to an appropriate context within the OGSA-DAI engine. To achieve this, we extend security context interfaces already existent in the OGSA-DAI framework. The security token extracted from the HTTP request header by the frontend interface is used to instantiate a custom security context for the service.

When executing a distributed query using DQP, the workflow of OGSA-DAI defines tasks (called activities) that communicate with remote data or mediation nodes on request to obtain or deliver data from or to a remote data node. These activities were modified to implement the existing marker interface for expressing the security context within the extended activity task. The interface signalizes the OGSA-DAI engine to push the available security context into the activity upon initialization. Consequently, a custom-built security context is injected and used to authenticate and authorize requests to remote services, which, in turn, use the security token passed along with the request to respond asynchronously. To impersonate requests, the security context populates the HTTP headers of subsequent requests with given credentials upon completion of the modified activities.

We have implemented the described delegation of user authentication in context of distributed query execution, enabling secured, asynchronous, multi-way communication – and thus secure (Virtual) Dataset Service interoperation for the VPH data infrastructure. Our aim was to minimize modifications in core OGSA-DAI to a bare minimum through the use of available extension points, keeping the components of Dataset Services loosely coupled and the dissemination of credentials on a need-to-know basis. In particular, the forwarding of credentials through a security context is workflow-agnostic and, in principal, not restricted to data federation use cases. In addition, the described delegation of authentication information allows us to associate the execution of a request with its initiator, even if it is part of a distributed query. This enables auditing capabilities for the data infrastructure.

IV. Data Infrastructure Testbed

In this paper we extend the scenario and performance results presented in [6], where we described the Dataset Services and their performance in relation to the query type. The results of the following experiments demonstrates the performance in a series of tests executed on top of the VPH-Share data infrastructure and involving different Clouds of the Atmosphere.

The setup has been chosen according to the requirements of the project. Two related dataset sources, originating from the Sheffield Teaching Hospitals containing clinical data are exposed using two separate Atomic Dataset Services within the Vienna private Cloud. Atomic Virtual Dataset Services on the other hand are deployed into the following three Clouds of the Atmosphere:

- Local (Vienna): The service is deployed in the same Cloud as the Dataset Services. The Virtual Dataset Service instance is assigned 2 vCPUs and 1GB memory;
- Remote (Krakow): The service runs on a remote Cloud. The instance is assigned 1 vCPUs and 1GB memory;
- Commercial (Amazon EC2): The service runs on European Amazon EC2 instances the specification of VMs types can be viewed in [10].

An important question we are addressing with regard to the Amazon EC2 is whether the query execution is largely dominated by the transfer time of data and other network-related issues, or if more expensive and better equipped virtual instances can achieve higher performance. The answer may give a strategic direction on how to best leverage the Cloud infrastructure.

For testing purposes, the available data were substantially inflated through random generation to provide a realistic measure of the performance on the amount of transported data. The client, placed in Vienna, is set to receive results in the WebRowSet XML format, which further inflates the amount of transferred data. Each request aggregates both dataset sources through a table join returning the following three classes of data:

- small - 1.000 rows, 497.211 bytes (485 KB)
- medium - 10.000 rows, 2.334.666 bytes (2.22 MB)

- large - 100.000 rows, 18.980.031 bytes (18.1 MB)

The client records the time in milliseconds until the final result is received. Both, the Dataset and Virtual Dataset Services are tested to provide a measure of the overhead of data mediation and the performance in relation to the remotely located services. Each test was carried out 30 times, recording average query execution times. The first two executions were excluded, to ensure that service contexts have been fully initialized.

V. PERFORMANCE RESULTS

Due to the remote location of the tested Virtual Dataset Services and different networks they are situated in, the performance of the data infrastructure will strongly depend on the current bandwidth and latency, which may change during the course of the day. The results presented here were acquired trying to minimize this hazard by executing the tests in a time span of under 4 hours. Nevertheless, the measurements are meant to give the reader a basic idea of what to expect when executing queries against the presented data infrastructure.

Table I provides a reference in terms of local request and data fetching, as all services are deployed in the same network in Vienna. In this first experiment, the Dataset Services exposing the dataset sources return results according to the data classes defined above. The Virtual Dataset Service aggregates data from both Dataset Services by applying restrictions to fetch the first half of the data from one and the last half from the other dataset source. This filtering is relayed and executed directly in the DBMS, thus having a minimal impact on the performance. Median times differ only slightly from the average times presented above, as the variation in recorded was rather low and no significant outliers were recorded.

TABLE I
PERFORMANCE COMPARISON BETWEEN DATASET AND VIRTUAL DATASET SERVICES

Data size	Dataset service	Virtual Dataset Service
Large	1599ms	3494ms
Medium	259ms	442ms
Small	132ms	283ms

Performance results of the Dataset Services executing data requests within one Cloud. Execution times roughly double when two dataset sources with the same data size are mediated.

The results show that the query execution time roughly doubles when aggregating the two datasets. This overhead was expected due to the orchestration of the data mediation, temporary data storage, transformation and aggregation.

Fig. 4 provides performance results comparing Atomic Virtual Dataset Services deployed on the local, remote and commercial Clouds of the Atmosphere. The chart is sorted by performance. Due to the pay-per-use concepts of public Clouds it is important to take the pricing (as of August 2013) of the corresponding instances into account for the discussion.

While the difference in performance between the dedicated local and remote Clouds is minimal, the difference between the two instances deployed in private Clouds and that of the Amazon EC2 instances is fairly significant. As expected, private Clouds such as the one in Vienna and Krakow typically have less traffic than a popular commercial Cloud. Nevertheless, the question on whether having more resources in an instance can improve the performance of the Virtual Dataset Service can now be answered. The performance is mainly driven by the transfer-rate, however a careful selection of a suitable instance type may provide a significant benefit.

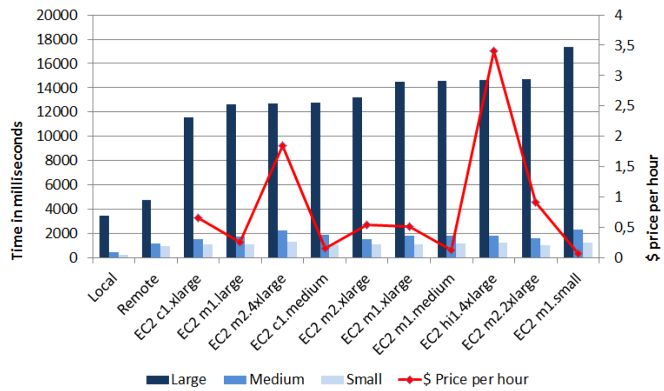

Fig. 4. Performance tests on different Atmosphere Cloud sites: Local - Virtual Dataset Service positioned near the dataset sources (Vienna), Remote - away from the dataset sources (Krakow) and on the commercial Amazon EC2 cloud instances. The chart is sorted by performance, while the red line indicates the price of the corresponding Amazon EC2 instance.

Amazon EC2 instances used in the tests are considered to be used in the Atmosphere. They are categorized by Amazon into several instance types [10]. The m1-type are general purpose instances. These are cheap and have a moderate overall network performance. The m2-type instances are still general purpose, but memory optimized instances, while the c1-type instances are equipped additional compute resources. Finally, the hi-type instance is a special instance type optimized to work in a cluster. The instances within the types range from small to xlarge denoting the increasing amount of resources they are equipped with. It is important to note, that the larger an instance, the better the network performance it is provided with as well. This is reflected by the first three top performers. The results also show that the top performer is the high-computing instance c1.xlarge. In fact, a request to the Virtual Dataset Service will spawn several dozen threads for processing, making high-computing instances suitable hosts these services.

Memory-optimzed instances on the other hand are less beneficial. Additional tests were executed to gain insight into the performance in relation to memory allocation. The working memory of the JRE was held to 512MB throughout all instances in order to ensure comparable results. However, further tests showed that allocating 1GB or 256MB to the JRE yielded similar to the ones presented here.

The hi1.4xlarge instance has a remarkable result. This extensive cluster-type instance, designed for I/O intensive applications and equipped with high computational capabilities, was not a top performer at all. While cluster-type instances within Amazon share a high bandwidth and low latency network connection, we cannot draw benefits of it because the presented services are not used as a cluster application.

VI. Conclusion

The VPH-Share project develops a comprehensive Cloud-based service framework for the sharing of data, information, models and workflows on human physiopathology. VPH-Share leverages different Clouds through the Atmosphere, unifying private and public IT resources. The VPH-Share dataset service environment provides support for accessing and integrating distributed heterogeneous dataset sources.The associated security framework restricts access to critical clinical data. Virtual Dataset Services implement security token delegation to allow transparent access to the underlying dataset sources. All concepts described in this paper were implemented, deployed and tested.

Performance results show that computational instances are more suited for deployment of the Virtual Dataset Services, as these deal with aggregation, filtering and general processing of data. Due to the fact that performance is mainly driven by the data-transfer rate, different instances provide a marginal difference in performance. A better approach is to seek a cheap, balanced instance in terms of CPU capacity, memory and bandwidth, rather than utilizing more expensive hosts.

Future work will be focused on implementing a comprehensive policy-driven access control system for the data infastructure.

VII. Related Work

Over the last years, many projects emerged with the aim to establish a solid and secure clinical data infrastructure as a foundation for their research platform. We discuss a few of the many related research efforts for establishing clinical data infrastructures.

The @neurIST project [11] was creating a secure data grid infrastructure for supporting the research and treatment of cerebral aneurysms. The data infrastructure was based on data integration and data federation services, secured through a similar security framework In addition to the security measures adopted by @neurIST, the VPH-Share security framework is aimed at providing a transparent, generic and configurable endpoint security for their application services, regardless of the service communication interface. While @neurIST aimed at building its own trusted virtual organization, VPH-Share reuses BiomedTown to help foster and cement trust relationships with existing VPH users.

The European Project DebugIT [12] is constructing a Cloud infrastructure for integrating and federating distributed clinical dataset sources with a major focus on semantic integration, as well as security and privacy. In contrast, the VPH-Share project offers an extensive Cloud environment, providing users with generic tools that support the whole life-cycle of dataset sources, including selection, annotation, provisioning and deployment of the data into a secure research network.

In addition, the European Project Hypergenes [13] built a data warehouse infrastructure for supporting unified access to project relevant datasets, while the European Project Health-e-Child [14] created a Grid-enabled network for sharing biomedical data on the basis of Grid technologies.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement #269978 (VPH-Share).

References

[1] S. Benkner, J. Bisbal, G. Engelbrecht, R. D. Hose, Y. Kaniovskyi, M. Koehler, C. Pedrinaci, and S. Wood, "Towards collaborative data management in the VPH-Share project," in *Proceedings of the Intl. Workshop on Cloud Computing Projects and Initiatives, in conjunction with Euro-Par 2011*. Bordeaux, France: Springer, Aug 2011.

[2] M. Koehler and S. Benkner, "VCE - A Versatile Cloud Environment for Scientific Applications," in *The Seventh International Conference on Autonomic and Autonomous Systems (ICAS 2011)*, Venice/Mestre, Italy, May 2011.

[3] M. Antonioletti, M. Atkinson, R. Baxter, A. Borley, C. Hong, P. Neil, B. Collins, N. Hardman, A. C. Hume, A. Knox, M. Jackson, A. Krause, S. Laws, J. Magowan, N. W. Paton, D. Pearson, T. Sugden, P. Watson, and M. Westhead, "The design and implementation of grid database services in ogsa-dai: Research articles," *Concurrency and Computation : Practice and Experience*, vol. 17, no. 2-4, pp. 357–376, 2005.

[4] P. Nowakowski, T. Bartynski, T. Gubala, D. Harezlak, M. Kasztelnik, M. Malawski, J. Meizner, and M. Bubak, "Cloud Platform for VPH Applications," in *8th International Conference on eScience 2012*. Chicago, USA: IEEE, Oct 2012.

[5] Amazon EC2 Web Services, "http://aws.amazon.com/ec2/," 8 2013.

[6] S. Benkner, C. Borckholder, M. Bubak, Y. Kaniovskyi, R. Knight, M. Köhler, S. Koulouzis, P. Nowakowski, and S. Wood, "A cloud-based framework for collaborative data management in the vph-share project," in *Proceedings of the Intl. Workshop on Cloud Computing Projects and Initiatives, in conjunction with AINA*. USA: IEEE CPS, March 2013. [Online]. Available: http://eprints.cs.univie.ac.at/3674/

[7] M. N. Alpdemir, A. Mukherjee, A. Gounaris, N. W. Paton, P. Watson, A. A. Fernandes, and D. J. Fitzgerald, "OGSA-DQP: A Service for Distributed Querying on the Grid," in *Advances in Database Technology - EDBT 2004*, ser. Lecture Notes in Computer Science, E. Bertino, S. Christodoulakis, D. Plexousakis, V. Christophides, M. Koubarakis, K. Boehm, and E. Ferrari, Eds. Springer Berlin / Heidelberg, 2004, vol. 2992, pp. 3923–3923.

[8] *SPARQL 1.1 Federated Query*, W3C Std., Rev. 1.1, March 2013. [Online]. Available: http://www.w3.org/TR/sparql11-federated-query/

[9] BiomedTown portal, "http://www.biomedtown.org/," 8 2013.

[10] Amazon EC2 Instances, "http://aws.amazon.com/en/ec2/instance-types/," 8 2013.

[11] S. Benkner, A. Arbona, G. Berti, A. Chiarini, R. Dunlop, G. Engelbrecht, A. F. Frangi, C. M. Friedrich, S. Hanser, P. Hasselmeyer, R. D. Hose, J. Iavindrasana, M. Koehler, L. L. Iacono, G. Lonsdale, R. Meyer, B. Moore, H. Rajasekaran, P. E. Summers, A. Woehrer, and S. Wood, "@neurist: Infrastructure for advanced disease management through integration of heterogeneous data, computing, and complex processing services," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 6, pp. 1365–1377, November 2010.

[12] D. Teodoro, R. Choquet, E. Pasche, J. Gobeill, C. Daniel, P. Ruch, and C. Lovis, "Biomedical data management: a proposal framework," in *Stud Health Technol Inform*, 2009.

[13] European Project Hypergenes, "http://www.hypergenes.eu," 8 2013.

[14] A. Branson, T. Hauer, R. McClatchey, D. Rogulin, and J. Shamdasani, "A data model for integrating heterogeneous medical data in the health-e-child project," *CoRR*, vol. abs/0812.2874, 2008.