Containing the Cloud: Security Issues in a Large Scale Observational Pharmacovigilance Research Project

Jeffery L. Painter
Mathematics Department, North Carolina State University, Raleigh, NC, USA

Abstract— *The Observational Medical Outcomes Partnership (OMOP) is a public-private partnership designed to help improve the monitoring of drugs for safety. A software model for analysis of disparate data sources must take into account security, repeatability and efficiency in the transmission and communication of results. Each data provider has an individual stake in the ownership of their data and care must be taken to minimize the possibility of data compromise in the use of this data for regulatory purposes. An evaluation of system security must also be taken into account. Since this system will be comprised of several data providers and data consumers, care must be taken to evaluate the critical points of data access privilege and while maintaining the overall goals of sharing knowledge with the community.*

Keywords: cloud computing, security, OMOP, pharmacovigilance

1. Introduction

The Observational Medical Outcomes Partnership (OMOP) is a public-private partnership designed to protect human health by improving the monitoring of drugs for safety and effectiveness. The partnership is conducting a two-year research initiative to determine whether it is feasible and useful to identify and evaluate safety issues of drugs on the market.

The Partnership's methodological research will be conducted across multiple disparate observational databases (administrative claims and electronic health records) and plans to engage in collaborations with qualified organizations in a number of different ways: The Partnership is funding data provider organizations to participate in the initiative, either as a Research Core contributor by providing de-identified patient-level data into OMOP's centralized IT research environment, or as a distributed partner conducting the analysis within its organization and reporting back aggregate summary results to the Partnership.

Although this project is still in its initial research phase, requirements engineering has begun and while many of the technical requirements of the application have been well defined, securely handling the data transformation, transmission and analysis have not been looked at in great detail. The nature of the application requires that data be provided from multiple sources, transformed into a format that meets the Common Data Model (CDM) as defined by the OMOP initiative and then be made available for statistical analysis and review. In Figure 1, we see that the data is first in its raw form from the original provider, which may be coded in one of multiple coding schemes, such as MedDRA or ICD-9, CM. The Common Data Model provides a target to which all the data must be transformed in order to be used in the data analysis process of the application. Then a unified approach can be applied which allows for the comparison of a single hypothesis against several different data sources.

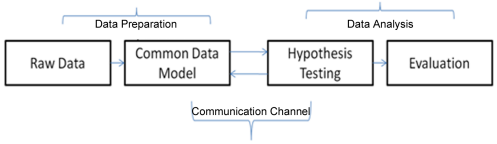

Fig. 1: OMOP Data Model

One method of accomplishing this goal is called the data centric model (or rather the application and data will live in the same space). The second method would be a distributed model in which each of the partners would run their own local instance of the OMOP system and report the evaluation results back to a central location. Yet a third method would be a hybrid approach, where the computational components of data analysis would be conducted in the cloud environment while the data still resides primarily with the original data provider.

The data centric model presents OMOP with the disadvantage of having to house and maintain each data provider's individual data. This opens up the possibility of data breach in addition to the cost of both housing and curating the data over the long term. Each of these issues could be overcome with proper diligence and man power, however, due to the cost of transmitting such large amounts of data (several hundred gigabytes per provider), it is not really a feasible plan when looked at under the cost estimations proposed in (1). Until such time that the cost to transmit data (both in terms of time and actual monetary cost) can be reduced, this model lacks sufficient benefit to be considered.

The distributed model also presents several issues relating to security in that the actual computations are conducted outside the reach of the OMOP initiative. While it would be possible to engineer systems which could mitigate the opportunities for interruption, modification or fabrication of results acquired from the distributed model, it would be more

advantageous (and cost effective) to move the computational component to the cloud environment.

The advantage of a proposed hybrid method is that it yields a higher degree of security and transparency to all parties involved. The original raw data still remains in the hands of the partners, while the analysis portion is conducted in a highly scrutinized yet central location, where all results will be coordinated and made available to the OMOP members and this is the model we will focus on for the remainder of this analysis.

2. Current Practices

Cloud computing is not an entirely new concept (everyone remembers Sun's campaign that "the Network is the Computer"), but the solidification of cloud computing is a relative newcomer to modern software development techniques. Since it is still so new, the focus of any research for developing cloud based applications is fairly thin. Therefore, this analysis builds on looking at the state of security methods found in traditional web application development in relation to how those same issues must be dealt with while developing a cloud computing application. We also look to the lessons learned from years of research focused on Grid computing, a close relative to cloud computing. In fact, Grid computing is focused on the ability of "allowing all components of our information technology infrastructure - computational capabilities, databases, sensors, and people - to be shared flexibly as true collaborative tools" (2). The Grid model has a rich history of research and expertise which should be looked at with careful consideration in regards to cloud computing.

The main idea of cloud computing is that the computational capabilities of large data centers can be shared among any number of individuals and cost is calculated similar to any standard utility, such as electricity, water, or natural gas (3). The idea of computing as a utility has reached a climax which now makes it a viable platform for large scale application development. But the relative newness of this platform still has several unresolved issues relating specifically to the area of security in application development.

Additionally, we will look at the various paradigms which currently exist in the cloud computing models, since there is not a single cloud as of yet from which all cloud services are deployed (1; 4). The heterogeneity we currently see is due in part to some of the current failings in having true development environments predefined for cloud computing as we see in traditional software development (such as IDEs and testing tools for a well understood pattern of software development).

2.1 Cloud Computing Models

Cloud computing has hit the mainstream. However, many still question whether or not it is a viable platform for delivering applications and Leavitt states that "IT departments are still wary of it because they don't control the cloud-computing platform" (5). However, controlling the platform is less of a concern as is the ability to be able to make efficient use of all the resources available. This depends in part on the type of application one wants to design and which vendor has the right mix of both computation and storage products to meet the needs of developers.

There are several offerings in the venue of cloud computing. The big three currently include Amazon Web Services (AWS), Google's cloud environment called AppEngine and finally Microsoft's environment called Azure (1; 4). Each of the platforms offers a variation on the concept of cloud computing and the development methodology for each is quite varied. We also find that "a major challenge of moving applications to the cloud is the need to master multiple languages a nd operating environments" (3). Data storage alternatives are also available with varying degrees of sophistication, availability and data duplication among each of the current providers.

There are essentially three levels of computing in the cloud which include:

1) Virtual Machine - Low Level Control
2) Application Framework - High Abstraction
3) Hybrid Model - Mid-level control of hardware

The first level is the one most easily accessible to traditional software developers. The virtual machine model essentially gives the user access to a complete server implementation (including operating systems from UNIX/Linux variants to MS Windows Server operating systems).

The application framework model essentially falls into the realm of developing web applications. The provider maintains an API which the application developer writes to while the framework infrastructure automatically grows or shrinks depending on demand of the application. This model works well for those who have little prior knowledge of demand peaks for an application and wish to deploy something quickly. It however limits you to the functionality provided by the framework, and this model tends to be more proprietary by imposing such controls. However, the benefit is the only learning curve required to overcome is learning to write to a single programming API.

The hybrid model combines the flexibility of a full-fledged programming environment running in a virtual machine which is then managed automatically by the providers own methods for controlling resource utilization. In this model, you write the application targeted toward a particular virtual machine platform that executes the code, and as demand for computation increases, the infrastructure will automatically allocate the resources required. However, this also requires a higher level of programmer sophistication and a thorough understanding of parallelizing code (1).

AWS falls under the virtual machine model, providing the illusion of an infinite supply of compute power, but

requires the demand of managing the acquisition and release of those resources on the developer himself (1). While this may seem like a big hurdle to overcome, the level of control allotted by the AWS is unsurpassed. The user is free to install any software or external applications (such as a statistics package) to run in the cloud environment. This level of flexibility and control gives the AWS platform the most freedom in terms of being able to run a large scale scientific application.

The Google AppEngine is categorized under the second level of cloud computing - application frameworks - where Google will scale and shrink the resources allocated to your application as needed based on user demand of the system. The development of any application on the Google platform is restricted then to running exclusively on their servers, and additionally the application is further restricted in that it must conform to the model of a traditional web application (4). For large scale scientific computing, the control and precision required to address intense computational needs would not be available, nor would the introduction of custom libraries or 3rd party applications for statistical analysis be allowed. Each of these drawbacks impede the selection of this paradigm for our research needs.

The next platform is Microsoft's Azure. "Microsoft Azure aims to provide an integrated development, hosting, and control Cloud computing environment so that softare developers can easily create, host, manage and scale both Web and non-web applications through Microsoft data centers" (4). This follows the third compute model which is closer to a hybridization of the Amazon and Google systems. It handles many of the low level infrastructure management automatically, but requires that applications be targeted to the .Net CLR (Common Language Runtime) environment. For existing applications, this means migrating from running an application in your own data center to the cloud is virtually seamless. However, you are forced into following the Microsoft development stack for software development which includes having to use a variation of the MS SQL Server for data storage. This lock in to a vendor specific tool set and cloud environment makes it a less attractive alternative (1).

2.1.1 Security in Cloud Computing Models

AWS has provided several technical briefs describing case studies of HIPPA compliant applications being developed and deployed on their cloud infrastructure which at this time is the only one of the three providers willing to make this type of claim (6; 7). Their security model is well documented, simple and easy to understand. They provide an environment that is locked both internally and externally from access except by the creator of the virtual machine via remote secure shell connection.

It should be noted however, that not all agree on the security models implemented by AWS. Viega laments that "unfortunately, Amazon currently lets a user have only one set of credentials per account. This makes it difficult to run applications in multiple pieces, with each piece administered separately either by business function or geography" (8).

The Google AppEngine security model is not described as thoroughly in the literature. The level of security provided by an application deployed on their cloud is primarily dependent on how well the security issues are addressed by the creators of the software to be implemented.

Microsoft's Azure platform again provides application deployment targeted to the .Net framework, which also means that security issues are primarily focused around the actual development of the application which you will deploy on their servers.

While each of the cloud computing models security requirements must be taken into account, the biggest threat to a large scale scientific application such as OMOP is arguably the model implemented for data access since data is of primary concern. There are currently two models under consideration for OMOP. The first is a data centric model, which is to say that all the partner data would actually be "uploaded" to the cloud application for standardization and analysis in the OMOP system. The second model is a distributed model, where each provider maintains ownership of their data and only portions are made available to the OMOP system as necessary.

Both have a cost/benefit association. However, after initial review and due to the fact that much of the partner data is dynamic, the initial recommendation of a data centric model fails to meet several basic criteria for a cloud computing application. Of primary concern is not the storage of the data, but the time to transmit such large databases into the cloud. Frequent updates of the data would make it impractical to make continuous updates to the data stores in the cloud, whereas a distributed model would solve all of these issues. However, the distributed model introduces a higher degree of complexity to the overall system.

Trust is the main concern in all web based applications, and ensuring that the data providers concerns and data ownership are protected while also providing the freedom required to probe these systems for the appropriate data for analysis is a balancing act that requires careful consideration and planning to implement in a software system that performs reliably and securely.

2.2 Security Models

In addition to security in the cloud, we must also pay attention to an overall security model to be enforced on the application itself (that is in the application layer - not the cloud layer). The three models under consideration included (1) centric model (2) distributed model and (3) a hybrid model for data access.

Again, the centric model forces a higher burden on the OMOP system in terms of mitigating breaches in data access,

conforming to HIPPA standards for patient data privacy in addition to the cost of storing and maintaining said data.

The distributed model reduces this burden from the OMOP project itself; however, it opens up the system to an added dependency of extending the computation away from the cloud and back on to the individual partners. Without the ability to know in advance the number of analyses to be conducted and the computing resources necessary to accomplish said analyses, this also proves less than desirable.

Maro, Platt et al describe a design for a "distributed health data network that allows secure remote analysis of separate data sets, each derived from a different medical organization's or health plan's records" which address the ability for "data holders to retain physical control over user of their data, thereby avoiding many obstacles related to confidentiality, regulation, and proprietary interests" (9). They argue that by providing this level of control to the data providers, it encourages more participation and involvement of private sector organizations to share much needed information with research organizations such as the OMOP initiative. However, they do not include a fully regimented description of how this might be accomplished. However, this method again seems to imply that they are looking towards a distributed model, which we have already noted has deficiencies.

Given the focused goal of the OMOP system to evaluate safety issues of drugs on the market, this is an opportune time to dive further into how this might be accomplished by way of cloud computing.

Since trust is the biggest concern in most web based applications, by making use of the hybrid model suggested earlier, we can help to ensure that the data providers concerns over data ownership are protected while also providing the freedom required to probe these systems for the appropriate data for data analysis.

The hybrid model proposal is simply this. A multi-tiered architecture shall be constructed such that a secure communication channel is created between the data provider and the OMOP system. Authentication to the OMOP system will be facilitated either through secure access mechanisms or some type of public key authentication, where the keys are vetted by the OMOP system itself.

When a request for data pertaining to a certain investigation is made, the data types will be transmitted in the common data model to an instance of a data collector running at each of the participating data provider's location. The request for data will be processed and the transformed data from the provider will then be returned to the cloud application for further analysis. Each data set retrieved will be denoted by its provider before results are drawn for comparison. In this way, it will be possible for all results to be traceable back to the original source without the need to retain any of the original raw data from the source provider.

This is in line with much of the security mechanisms discussed in Grid application design. By maintaining strict policies on key generation, expiration and control, we can help to ensure that the data being provided is authentic and can be traced back to its original sources if need be. Additionally, the transmission of the data will be conducted through a secure encrypted channel such as the HTTPS protocol.

To further enhance the security model for data transmission, all data requests will originate from the OMOP system in the cloud following a Pull Authorization as described by Chakrabarti. "In the pull model, the users provide the minimum credentials to the access controller and it is the responsibility of the controller to check the validity of the user based on policies of the system" (10).

In this way, the OMOP cloud application can maintain its own set of privileged access instructions separate from the originating data providers. It provides a way for the control of data to be handled at a centralized location where access and control can be logged and monitored for compliance with all federal policies and regulations.

2.3 Data Security in Cloud Computing

While we are recommending the hybrid model for development to limit the liability of the OMOP initiative in respect to data security, there is still the need to store the transformed data in the cloud for analysis.

Therefore, we must consider the issue of data security in cloud computing. As stated before, each cloud provider also includes the ability to store data in each of their respective platforms. Each of the storage models are given a more thorough analysis in (1).

Both the Microsoft and Google data storage options are proprietary and the ability to demonstrate whether they are feasible options for parallelized scientific computations remains unclear. The Amazon model allows the most flexibility and also provides data replication across their infrastructure making a minimum of three copies of any and all data (1).

Data retention policies will help to insure that data breaches do not occur (both internal to the system and external) and that transparency in the use of data for analysis is made clear to all participants in the OMOP system. The data management policy should be maintained within the OMOP system control and access management component which will automatically enforce said policies without a strict override from authorized personnel or the data providers own agreement.

Trust is a primary concern in any scientific application, and ensuring that our stakeholders' concerns over data ownership are protected while also providing the freedom required to probe these systems for the appropriate data will help to insure the system is compliant with all regulations and has a clear mechanism to log and audit data access and manipulation.

Kaufman raises this same issue when she asks "is security solely the storage provider's responsibility, or is it also incumbent on the entity that leases the storage for its application and data?" (11). Therefore, the onus of data security (in terms of data access and analysis) is transferred primarily to the design and implementation of the OMOP control system which will reside in the cloud.

2.3.1 Limitations of Cloud Storage

The greatest benefit to be offered by cloud computing and storage is the transfer of risk from the application developer and service provider to a larger entity such as Amazon, Google or Microsoft. These platforms offer data management services which "can scale to huge amounts of data and large number of requests" but "stop short of providing transaction guarantees even on a single row" (12). The limitations identified by Das, Agrawal and Abbadi will need to be addressed for making use of the cloud computing environment for data storage.

Das, Agrawal and Abbadi note that "to satisfy the scalability requirements of web applications, designers have sacrificed the ability to support distributed transactions" (12). Because of this, and due to the nature of our application, transactional calculations will need to most likely occur in memory rather than relying on traditional database storage techniques. To mitigate discrepancies in data and the risk this might have towards identifying safety issues in drug data, special provisions will need to be created which can effectively deal with manipulating large amounts of data without the traditional SQL server environment to guarantee atomic operations. It is possible that an in memory database solution may be possible, but this should not be counted on. The ElasTraS system described in (12) provides some advice on how to effectively deal with these problems and should be looked at in closer detail in the future.

2.4 Authentication Methods

To enable use of any secure online application, a user or system must authenticate their access to that system in order to make use of its resources. According to Chakrabarti, this problem is usually addressed using three different mechanisms based on either a shared secret, public key encryption or a third party certificate system.

In order to decrease the burden on the end user, the system should employ methods which make secure authentication both easy and accessible. Certificate authentication provided by third party (CA) authorities, while very secure, actually opens the door to a less secure system in the case that security certificates are left exposed and unprotected on remote systems. For this reason, it is generally more accepted to use a public private key option or to enforce the use of strong passwords (10).

Therefore, for this application, we recommend the use of a strong password mechanism for user access to the OMOP User Interface and a combination of a public/private key system for enabling the communication channel between the data providers and the OMOP System controller. In the next section, we describe an account policy model which could also facilitate the lifecycle management of the public/private keys for the individual data providers.

2.5 Account Policies

There will be a need for automated account policy management for the OMOP system to be a success. User access for both research and maintenance of the system requires different levels of authorization and control. From the cloud computing perspective, access is limited by means of either secure shell access or other proprietary access mechanisms which any application developer is at the mercy of their cloud provider to accept and defend whether or not it meets the criteria set forth in their account policies for the overall application (1). However, the application layer must also implement some access control mechanisms of its own which is where this discussion will focus.

There are many models for user, group and role based authentication. However, where many of these models are lacking is in the area of user administration, policy enforcement and control. Aikema, Kiddle and Simmonds proposed a Virtual Organization (VO) model for account policy management in grid environments (13) which is both robust and timely in light of implementing a large scale multi-user system such as OMOP proposes.

Virtual Organization (VO) is a management system which "allows groups of users to access the system, without requiring that accounts be pre-created for all users who might attempt to connect to this system" (13). Their model of user and attribute associations is the core component of distributing resources to individuals who need it. In the cloud, in theory there are no restrictions on the resources available to users. There are still costs measurements which must be taken into account. A single OMOP analysis can take between 48-96 compute hours based on examples that we have run previously. This amount of compute time can add up quickly when you are examining potentially hundreds of drug-interaction hypotheses. In order to mitigate the potential cost/benefit of cloud computing, one must consider priority and resource control in addition to cost control. The VO model allows varying levels of users to be created each of whom can be associated with an attribute of compute time allocated for any individual study.

Control of these policies must be guarded by administrative privileges. In addition, care needs to be taken that analyses of data is not duplicated, increasing the cost of use of the OMOP system in the cloud environment. By implementing a multi-user policy that both limits resources while maximizing results, the OMOP system can hope to achieve a more secure operating environment and make the best use of the available resources.

The VO system helps to establish policies regarding "account lifecycle, creation, access, expiry, and cleanup" (13). The nature of a collaborative system such as this makes clear that results obtained by one user should be preserved regardless of whether or not that user still has access to the system. Therefore, while job creation and generation of results may be controlled by a single individual, those results should not be tied to the user's individual account. And when a user is no longer participating in the partnership, it is necessary to plan an access policy that includes attributes defining the lifecycle and expiry properties in order to reduce the burden and potential security risks of leaving open old accounts.

3. Proposed Architecture

Here we will propose an architecture to be considered to assist in identifying potential security vulnerabilities. Figure 2 shows a high level overview of the minimum components required to support the requirements of initiating an analysis from a researcher and interacting with partners to access the data necessary to complete an analysis.

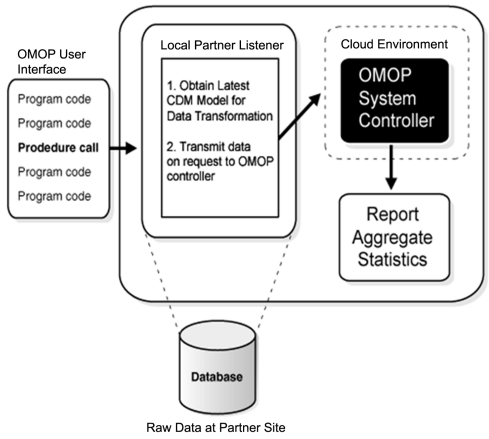

Fig. 2: OMOP System Proposed Architecture

The application will be driven by a user interface where calls are made to perform an analysis of some drug in order to investigate potential safety issues. This web application layer will be exposed to the internet and creates a potential security threat. Data may be intercepted, modified or manipulated in some manner which raise a question of the reliability of results produced by the application.

While implementing strict security controls in web based applications present difficult problems to solve, there are several standard web practices which are well described in the literature (4; 14) and should be followed regarding this aspect of the application. It is assumed here that the user interface portion of the application will also be delivered via the cloud infrastructure. However, it is distinguished as a separate component here for clarity.

After an initial drug investigation is requested, the data required for processing is identified via the common data model and a request to pull that data will be transmitted to one or more data providers. Inclusion and exclusion criteria will then be applied to the target data provider's databases in order to prepare a data set to be transmitted back to the OMOP System controller which resides in the cloud. A separate application will reside at the partner location called the "Local Partner Listener" (LPL). As described in (9), each of the partners will be held accountable for maintaining proper security protocols on their end to protect their data assets. The connection between the LPL and the OMOP system controller must be encrypted to prevent unauthorized data access during transmission. This may be accomplished by utilizing the HTTPS transport protocol.

After the data set is delivered to the OMOP System controller, a policy must be associated with that data set as defined by the LPL. This would indicate the lifespan of the data, whether or not it may be used in future analyses and if the data set may be shared with other partners in the OMOP system or if it should be placed in a "data silo" where it will remain until the end of its lifecycle.

Once the data is transmitted, the OMOP System controller will create a process to analyze the data and allocate the necessary resources to accomplish the task. The system will update an internal monitor to indicate the job is in progress and prevent a duplicate job from being initiated.

At job completion, the results will be published back to the user who initiated the analysis and the results will also be made available back to the data provider. If additional data sets were marked as "shareable" then those results will also be made available to each of the participating partners. Summary statistics will be generated and compiled to compare against each of the data sets and published to the OMOP results section which will be accessible only to those authorized to review safety analysis results.

Each of these steps highlights several layers of access and control mechanisms that need to be put into place in order to maintain a secure and reliable system. Data access and review should be logged and auditable for every calculation and manipulation performed in order to comply with current laws and regulations.

This ability to capture a history of each data analysis is necessary to maintain trust and traceability in the system. In another proposed large scale public health research system, the developers note that "logs contain the certified identity of an investigator, the identity of the trusted agency who certified investigation, and the time of the query" (15). They further emphasize that "a single institution cannot turn off logging or hide disclosures without coming under immediate scrutiny" (15). Trust in the system comes with ensuring secure access is maintained and accountability will always be at the heart of the overall application design.

4. Discussion

Future work will be limited by the failure of many of the cloud computing infrastructures to provide a facility to debug and test applications fully in a cloud environment. The weakest link may not be in the cloud itself, but in the communication between the cloud and outside systems. By providing developers with the tools to integrate security into their applications from the start, cloud providers can help to insure that the cloud will be a reasonable target large scale scientific applications. Latency in transmitting data in and out of the cloud also impedes the development of many "real time" applications from making use of large data sets. However, with improvements in bandwidth these concerns may lessen over time.

"Combining distributed data, computation, models and instruments at unprecedented scales can enable transformative research. The analysis of large amounts of widely distributed data is becoming commonplace" (16).

Even though it is still in the research phase, the OMOP initiative is making moves to become a secure and reliable large scale scientific application. This includes coordinating with the current members of the OMOP partnership and discussing the privacy concerns of both the data providers and the research scientists that will be making use of the application. A clear policy outlining access control, data privileges and overall maintenance of the application must be developed.

5. Conclusion

This paper discusses many issues of security as it relates to developing a large scale data analysis system in the cloud computing environment. The data distribution models could be applied to any large scientific computing project which is looking at cloud computing as a potential target for software application development.

We identify the potential security risks and mitigation strategies which should be followed in the development of a system designed to work with public and private health record databases. A proposal of architecture for such a system is evaluated for security risks and recommendations to prevent unauthorized data and application access is reviewed.

We believe the state of cloud computing is ready for prime time development, but the lack of clear security infrastructure for application development still leaves ample opportunity for research to continue.

References

[1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia, "Above the clouds: A berkeley view of cloud computing", Tech. Rep. UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.

[2] Ian Foster, Carl Kesselman, and Steven Tuecke, "A security architecture for computational grids", in *5th ACM Conference on Computer and Communications Security*, San Francisco, California, November 1998, Grid and Pervasive Computing, pp. 83–92.

[3] Brian Hayes, "Cloud computing", *Commun. ACM*, vol. 51, no. 7, pp. 9–11, 2008.

[4] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility", *Future Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.

[5] Neal Leavitt, "Is cloud computing really ready for prime time?", *Computer*, vol. 42, no. 1, pp. 15–20, 2009.

[6] AWS, "Amazon web services: Overview of security processes", White paper, Amazon, September 2008, Available online (9 pages).

[7] AWS, "Creating HIPAA-compliant medical data applications with amazon web service", White paper, Amazon, April 2009, Available online (8 pages).

[8] John Viega, "Cloud computing and the common man", *Computer*, vol. 42, no. 8, pp. 106–108, 2009.

[9] Judith C Maro, Richard Platt, John Holmes, Brian Storm, Sean Hennessy, Ross Lazarus, and Jeffrey S. Brown, "Design of a national distributed health data network", *Ann of Intern Med*, , no. 151, 2009.

[10] Anirban Chakrabarti, *Grid Computing Security*, Springer, 2007.

[11] Lori M. Kaufman, "Data security in the world of cloud computing", *IEEE Security and Privacy*, vol. 7, no. 4, pp. 61–64, 2009.

[12] S. Das, D. Agrawal, and A. Abbadi, "Elastras: An elastic transactional data store in the cloud", *in HotCloud*, 2009.

[13] David Aikema, Cameron Kiddle, and Rob Simmonds, "An account policy model for grid environments", in *4th International Conference on Grid and Pervasive Computing GPC 2009*, Geneva, Switzerland, April 2009, Grid and Pervasive Computing, Springer.

[14] C. Brabrand et al, "Powerforms: Declarative client-side form field validation", *World Wide Web J*, vol. 3, no. 4, pp. 205–214, 2000.

[15] Andrew J. McMurry, Clint A. Gilbert, Ben Y. Reis, Henry C. Chueh, Isaac S. Kohane, and Kenneth D. Mandl, "A self-scaling, distributed information architecture for public health, research, and clinical care", *JAMIA*, vol. 14, no. 4, pp. 527–533, 2007.

[16] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers, "Examing the challenges of scientific workflows", *IEEE Computer*, vol. 40, no. 2, pp. 24–32, 2007.