A Proposed Case for the Cloud Software Engineering in Security

Victor Chang and Muthu Ramachandran

School of Computing, Creative Technologies and Engineering, Leeds Metropolitan University,
Headinley, Leeds, LS6 3QR, U.K.

Abstract. This paper presents Cloud Software Engineering in Security (CSES) proposal that combines the benefits from each of good software engineering process and security. While other literature does not provide a proposal for Cloud security as yet, we use Business Process Modeling Notation (BPMN) to illustrate the concept of CSES from its design, implementation and test phases. BPMN can be used to raise alarm for protecting Cloud security in a real case scenario in real-time. Results from BPMN simulations show that a long execution time of 60 hours is required to protect real-time security of 2 petabytes (PB). When data is not in use, BPMN simulations show that the execution time for all data security rapidly falls off. We demonstrate a proposal to deal with Cloud security and aim to improve its current performance for Big Data.

1 Introduction

The advantages of adopting Cloud Computing have been discussed in numerous papers [1-2, 4-6, 14]. Although organizations that adopt Cloud Computing acknowledge benefits offered by Cloud services, challenges such as security and privacy remain a scrutiny for organizational adoption. While overseeing the importance of security, the software engineering and development process should always design, implement and test security features. The software engineering process should be robust enough to withstand attacks and unauthorized access. The secured services on offer can be guaranteed to provide benefits for users and service providers [13]. For example, users are safe and protected while using Cloud-based services. Service providers can ensure that their services have high extent of security, so that their customer satisfaction and company reputation can be improved. In order to achieve this, service providers have preventive measures and rescued actions to reduce risk imposed by security breach.

While acknowledging the importance of Cloud software engineering in security (CSES) in the previous paragraph, CSES should be undertaken early in the software development cycle. We explain the design and implementation of the CSES, and describe how we prototype a Business Process Modeling Notation (BPMN) as part of our CSES service to test the performance with a cloud platform that contains 2 PB of data. We review a number software engineering proposals, we notice that none of them provides the full solution from design, implementation and services for Cloud

Chang V. and Ramachandran M..
A Proposed Case for the Cloud Software Engineering in Security.
DOI: 10.5220/0004985600710079
In *Proceedings of the International Workshop on Emerging Software as a Service and Analytics* (ESaaSA-2014), pages 71-79
ISBN: 978-989-758-026-0

security [1-3, 7-8, 13]. This motivates us to propose our Cloud software engineering in security (CSES) solution focusing on design and implementation. Before presenting our proposal, we review a few selected literature. We discuss what should be available in the design and then propose our own implementation by the use of BPMN. The breakdown of the paper is as follows. Section 2 presents a review of existing proposals suitable for our approach. Section 3 describes how BPMN can be used from design to implementation. Section 4 shows how BPMN can be used for implementation and results of simulations. Section 5 presents conclusion of this paper.

2 Review of Existing Proposals Relevant to Our Research

We review a few selected literature that is relevant for CSES described as follows. Gonzalez-Castillo [7] defines further classify software security engineering and its implementation into two major groups: ***software acquisition security*** (which includes the security specifications in all processes to buy, rent, or interchange software to use in an enterprise) and ***systems & software development security*** (which includes the security specifications in all processes to develop information systems). His approach is focused on system security for CSES as shown in Figure 1.

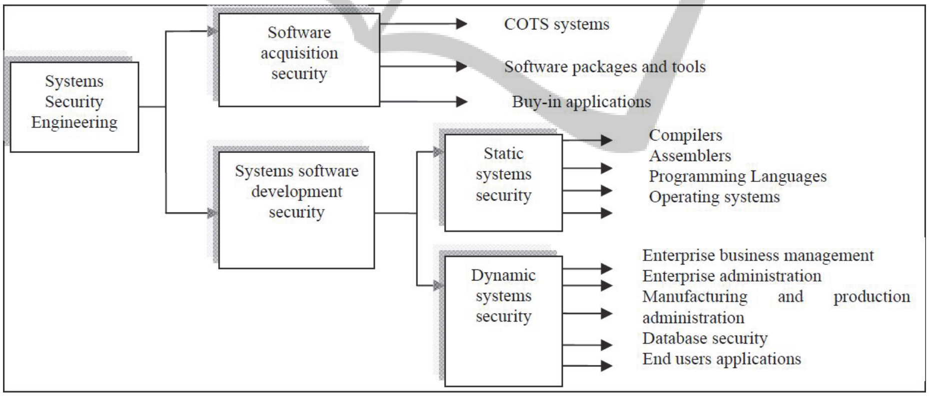
Systems Security Engineering
Software acquisition security
COTS systems
Software packages and tools
Buy-in applications
Systems software development security
Static systems security
Compilers
Assemblers
Programming Languages
Operating systems
Dynamic systems security
Enterprise business management
Enterprise administration
Manufacturing and production administration
Database security
End users applications

Fig. 1. Systems security taxonomy for CSES [7].

In Figure 1, software acquisition deals with COTs (component off-the-shelves), packages, and buy-in tools. Information systems development has been divided further into two main categories known as static and dynamic systems. Static systems security includes compilers (involves specification to develop compilers), assemblers, Programming Languages (PLs), and Operating Systems (OS). Dynamic systems security deals with enterprise business management (take control of business process information), enterprise administration (security specs to develop applications which objective is to provide control of administrative process information), Manufacturing systems, database systems, and end user systems which deal with developing applications for daily activity tools for end users. Design for security calls for specific design rationale and features supporting security explicitly. We can also distinguish furthermore in defining security design where functional design artefacts are created as usual with security. McGraw [8] has identified a number of security techniques against each stage in the software development lifecycle (SDLC). Figure 2 illustrates a set of those techniques. For example, abuse cases, security inspection and security modeling should be conducted as part of the Requirement Engineering (RE) process, security risk analysis should be conducted during design phase, external review and risk based security test analysis should be done during test planning stage, static analysis for security at the code level (this may include code inspection or automated code analysis tools equipped with security), and penetration testing & security breaks should be conducted during operational and field testing.

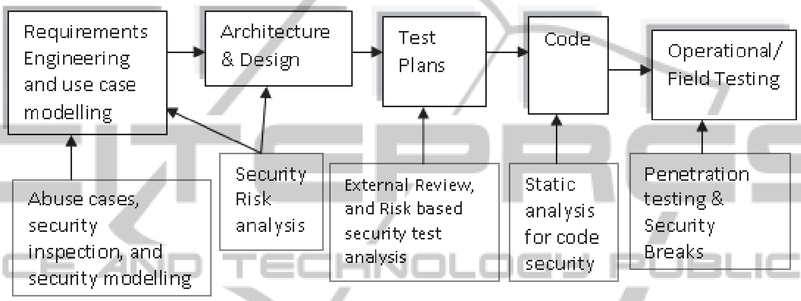
Requirements Engineering and use case modelling
Architecture & Design
Test Plans
Code
Operational/ Field Testing
Abuse cases, security inspection, and security modelling
Security Risk analysis
External Review, and Risk based security test analysis
Static analysis for code security
Penetration testing & Security Breaks

Fig. 2. Software Security Techniques (Build-In Security) [8].

There are different types of security designed for software as follows. Information security is the overall security for the whole enterprise and its network environment. Network security is to make sure the secured transactions and communications take place. Whereas application security is to make sure software systems as a whole is secured in its environment. All the software design work in CSES is suitable for information, network and application security. To demonstrate this concept, Ramachandran [9] proposes the Systems Security Engineering Life Cycle for CSES shown in Figure 3, with the emphasis in having a distinction between classical software engineering (SE) lifecycle vs. systems security engineering lifecycle (SSE) [10-11]. The SSE extracts and specifies security requirements using specific methods in addition to the usual functional modelling conducted during requirements engineering phase of SE lifecycle. This SE lifecycle can be used for CSES as follows. Ramachandran [11] has captured such good practices in the form of software guidelines across software development, reuse, and component based software engineering (CBSE). Software as a Service provides new abstraction for developing and delivering business application as part of the cloud. Service implementation is based on software component for implementing their core logics. Ramachandran [12] has produced a number of service component models for implementing services with build in security.

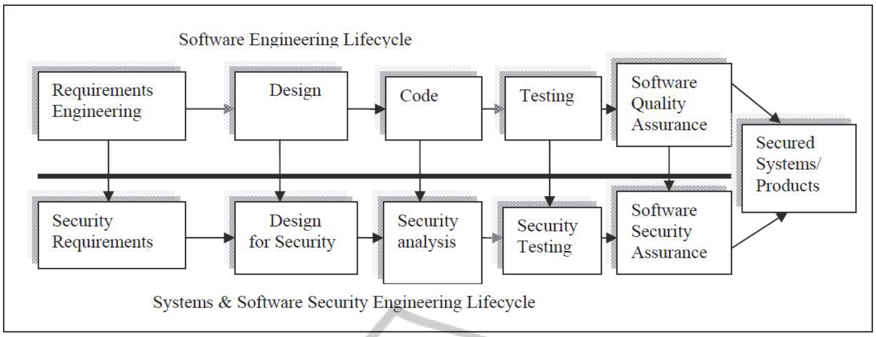
Software Engineering Lifecycle
Requirements Engineering
Design
Code
Testing
Software Quality Assurance
Secured Systems/ Products
Security Requirements
Design for Security
Security analysis
Security Testing
Software Security Assurance
Systems & Software Security Engineering Lifecycle

Fig. 3. Systems Security Engineering Life Cycle for CSES [9-12].

3 Business Process Modeling Notation: From Design to Implementation

This section describes the required steps to move from design to implementation phase in the development of CSES by the use of Business Process Modeling Notation (BPMN) in the Cloud.

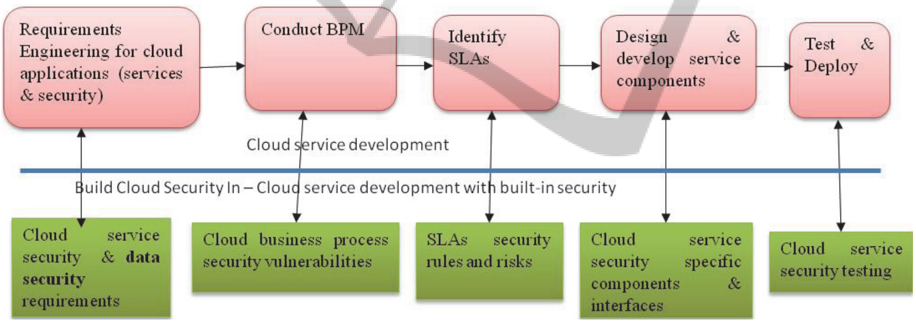
Requirements Engineering for cloud applications (services & security)
Conduct BPM
Identify SLAs
Design & develop service components
Test & Deploy
Cloud service development
Build Cloud Security In – Cloud service development with built-in security
Cloud service security & data security requirements
Cloud business process security vulnerabilities
SLAs security rules and risks
Cloud service security specific components & interfaces
Cloud service security testing

Fig. 4. Cloud security-based service development and integrating data security process with built-in security.

As shown in Figure 4, cloud service development are classified into a number of phases: 1) requirements engineering for cloud services during which time we can identify security related requirements from various stakeholders; 2) conduct business process modelling and simulations (BPM) for each cloud services during which time we can also simulate security aspects and study performance related measures and also introduce a possible number of intrusion and conduct simulations before actual service implementation take place; 3) identify Service Level Agreements (SLAs) identifies a number of service level agreements and regulatory and governance related compliances during this time we should be able to separate out security related SLAs and risks; 4) design and develop services during this phase we can actually implement security related threads that have been carried continuously from all phases, and finally; 5) test and deploy services that are developed with CSES. Additionally, we can:

1. The first step is to apply software security engineering techniques to all identified cloud services. This includes using security analysis tree and various other techniques specified by Ramachandran [10-11].
2. The second step is on identifying BPM which should include software security analysis for each business process identified. This will allow us to identify potential security threats that start with service requirements and business requirements as the input to conduct service security analysis using techniques such as Systems Secure Quality Requirements Engineering (SysSQUARE), and Microsoft Secure Development Lifecycle (SDL). The outcome of this process should yield a set of cloud services security requirements with clear indication of software security issues.

We can use BPM jointly with other framework such as Cloud Computing Adoption Framework presented in our other workshop paper or be used independently in the Cloud. Additionally, different states of user's data must be protected and managed logically and consistently. In order to use BPM to achieve this, additional work for BPM is required. This includes the Business Process Modelling Notation (BPMN) models shown in Figure 5. Several BPMNs (version BPMN2 2012) have been created for each of those scenarios to run a number of possible simulations with various business variability using an open source simulation tool, known as Bonita Soft BOS 5.8 [3]. Simulations of BPMN for Cloud Data Security and Application of the SysSQUARE method can be used to elicit security threats.

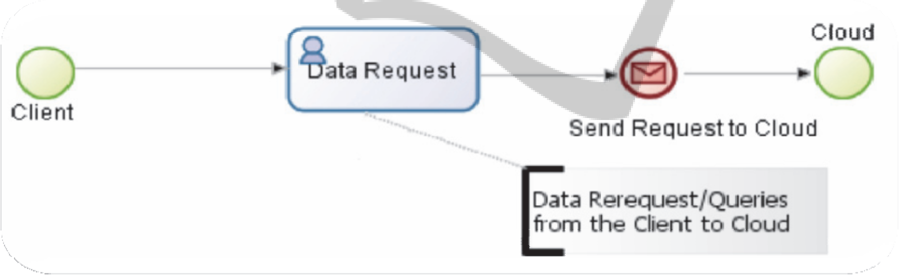

Fig. 5. Data Request Business Process Model for Cloud Security.

The Client of Cloud Computing contains a computer software or/and a computer hardware which dependents on cloud computing architecture to support the application deliveries, or which designed specifically for cloud service deliveries. A Client of Cloud Computing Architecture is an interface of common cloud user through the web browsers or thin terminals. Cloud provider is the one who offers the Cloud Service Delivery Models to Client through the internet. According to our proposed system the client just sends a request to the cloud then the remaining process is being taken care of cloud service provider who consists of Cloud Management Teams, Data Centres/Security pools and the Intrusion Detection Mechanisms. The hardware infrastrcuture is based on Southampton private cloud platform described in [5]. Figure 5 represents the BPMN process of Data Request flow from Client to Cloud. Our proposed solution can opt to use a framework for classifying cloud securities and policies. Cloud security is the key to business sustainability and hence we need to structure security related aspects into a simple framework that helps us to evolve and improve over a time period.

4 Business Process Modeling Notation: Implementation and Results

This section describes how to use BPMN for implementations and the results of simulations in the private cloud environment. Simulating a process allows us to study its behavior for their external events/triggers. Process simulation has been successful in several applications from low-end to high-end systems. Hence, simulating a BPMN model will help us to study business behaviors/performance for various expected and unexpected scenarios. The BPMN simulation process consists of a number of cyclic phases as shown in Figure 6. BPMN starts with an actor called Client with a small circle notation which sends a message to a process (Data Request with rounded square) which task has been devoted to take action based the request and therefore send a message to the cloud (finishing circle). The second phase is to annotate each element in the process and thirdly to create tasks, assign simulation variables (different types of requests both valid and invalid) to process and tasks in that process. Finally, create messages between elements in the process and run a number of simulations.

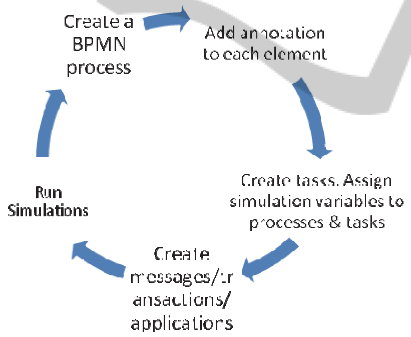
Create a BPMN process
Add annotation to each element
Create tasks. Assign simulation variables to processes & tasks
Create messages/transactions/applications
Run Simulations

Fig. 6. BPMN Simulation Process for CSES.

The data centers are the essential asset of Cloud Computing corporations, these all connect to all applications, storage services and servers. The business relies on the cloud data centers supports the business values and operations and drive maximum efficiencies. The data centers play key roles that need to be managed and planned carefully to meet growing performance requirements/demands from users and applications. The use of BPMN can simulate the daily operations in the data centers, which contain up to 2 petabytes (PB) of data. Figure 7 shows the BPMN model for different states of models for data security. The data center can use this model to study the performances of selected cloud data architecture. This process starts with a data status decision (diamond symbol) passes that data based on that decision to any one of the paths of the cloud storage processes (data at rest, data in use, and data in change/transition). This in turn passed on to a data security pool which is a separate lane with dedicated security processes (such as data security area and data center update) to study security controls in place before it ends.

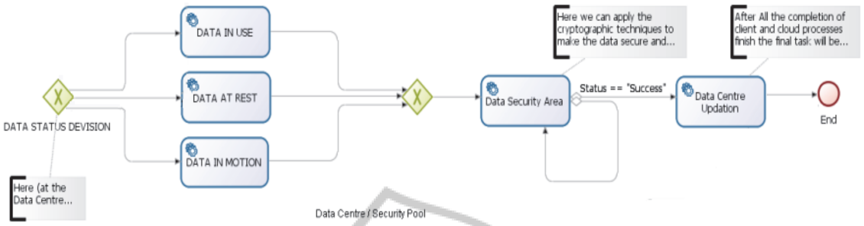
DATA IN USE
DATA AT REST
DATA IN MOTION
DATA STATUS DEVISION
Here (at the Data Centre...
Data Security Area
Status == "Success"
Data Centre Updation
End
Here we can apply the cryptographic techniques to make the data secure and...
After All the completion of client and cloud processes finish the final task will be...
Data Centre / Security Pool

Fig. 7. BPMN model for three types of cloud data security.

Cloud security is the key to business sustainability and hence we need to structure security related aspects into a simple framework that helps us to evolve and improve over a time period. In the implementation to result phase, we use BPMN for raising alarm in data security while all 2 PB of data in the Cloud has been intensively in used.

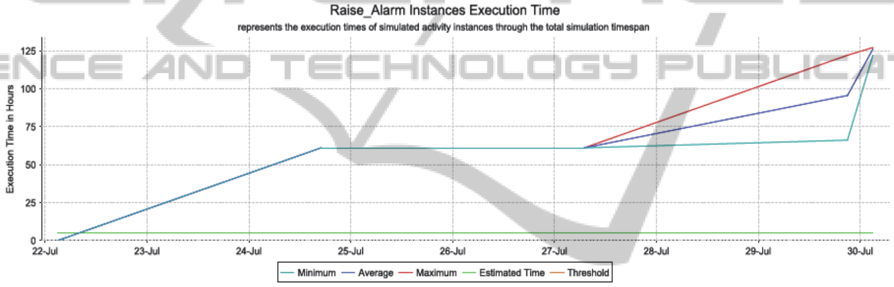

Fig. 8. Data Security Area Peak Access- High execution time when data in use.

Figure 8 shows a graph with peak execution time for entering the data security area of the business process. Results show that increased steadily from 0 to 60 hours between July 22 and the middle of July 24, 2013. The execution time stayed stable at 60 hours between the middle of July 24 and beginning of July 27. Some execution time increased due to the increasing demand sin security. The implications of this result show that data security instances execution time can be high when data was constantly in use. On the other hand, the execution time was less than 2 hours if data was not in use.

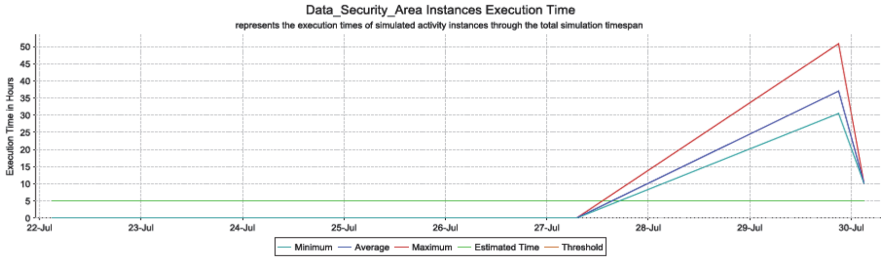

Fig. 9. Raising Alarm by BPMN for 2 PB data.

Figure 9 shows a graph with execution time when the BPMN process raised a security alarm. The execution time rose from 0 to three peaks (51, 36 and 30 hours)between July 27 and right before July 30, before falling to 10 hours of execution time right after July 30, 2013. The increment in execution time was necessary since BPMN alarm checked every single file and instance in 2 PB of data in the Cloud. This explained why such a long execution time was required. We plan to develop algorithms or methods that can optimize the security performance. The execution time to run each BPMN process only takes 2 seconds all the times, which has a very low execution time. This ensures that fast and efficient BPMB process can meet the requirement of business agility.

5 Conclusion

We present our Cloud software engineering in security (CSES) proposal from its system design to implementation phase. We review a few selected literatures and assert that none of them has the solution from design to implementation of Cloud security as yet. We then propose a unique approach to combine the recommended software engineering process with an emphasis on security. We use Business Process Modeling Notation (BPMN) to illustrate design to implementation of a good Cloud service. We use BPMN to demonstrate implementation for CSES with its supporting results. BPMN can be used to simulate the case of raising alarm for protecting Cloud security in real-time. BPMN simulation results demonstrate long execution time of 60 hours of protecting Cloud security of 2 PB. When data is not in use, BPMN can take less than 2 hours of their execution time. We are in the process of developing methods or algorithms to optimize the performance and hope to disseminate our research outcome in the next twelve months.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M. (2010), Above the Clouds: A Berkeley View of Cloud computing. Communications of the ACM, 53(4), 50-58.
2. Binz, T., Breiter, G., Leyman, F., & Spatzier, T. (2012). Portable Cloud Services Using TOSCA. IEEE Internet Computing, 16(3).
3. Bonita Soft (2012) BOS 5.8, Open source BPMN simulation software, http://www.bonitasoft.com/resources/documentation/top-tutorials
4. Chang, V., Walters, R. J., & Wills, G., Cloud Storage and Bioinformatics in a private cloud deployment: Lessons for Data Intensive research, Springer: CLOSER 2012, CCIS 367, pp. 245–264, (2013).
5. Chang, V., Business Intelligence as Service in the Cloud, Future Generation Computer Systems, DOI: http://dx.doi.org/10.1016/j.future.2013.12.028, (2014).

6 Chang, V., Walters, R. J., & Wills, G., The development that leads to the Cloud Computing Business Framework. International Journal of Information Management, 33(3), pp 524-538, June, (2013).

7 González-Castillo, O. Y., Selected Quality Metrics for Digital Passport Photographs, Logos

Verlag Berlin, ISBN 9783-8325-1965-0 (2008).

8 McGraw, G, Software Security: Building Security In, IEEE Security & Privacy, March/April, (2004).

9. Ramachandran, M (2008) Software components: guidelines and applications, Nova Publishers, NY.

10. Ramachandran, M., Software components for cloud computing architectures and applications, Springer, Mahmmood, Z and Hill, R (eds.), (2011).

11. Ramachandran, M., Software Security Engineering: Design and Applications, Nova Science Publishers, New York, USA, 2011.ISBN: 978-1-61470-128-6, (2011).

12. Ramachandran, M (2012) Service Component Architecture for Building Cloud Services, Published: August 20th, 2012, Service Technology Maga-zine Issue LXV, http://www.servicetechmag.com/I65/0812-4.

13. Santos, N., Gummadi, K. P., and Rodrigues, R., Towards trusted cloud computing, In Proceedings of the 2009 conference on Hot topics in cloud computing, (2009).

14. Vouk, M. A., Cloud Computing – Issues, Research and Implementations, Journal of Computing and Information Technology - CIT 16, page 235–246, Volume 4, (2008).