



Assurance of Security and Privacy Requirements for Cloud Deployment Models

Yandapalli Khajana Raju

Department of Computer Science & Engineering,
Sarada Institute of Science, Technology and
Management,
Srikakulam, Andhra Pradesh 532404, India.

Smt.Y.Triveni

Department of Computer Science & Engineering,
Sarada Institute of Science, Technology and
Management,
Srikakulam, Andhra Pradesh 532404, India.

ABSTRACT

Despite of the huge benefits of migrating enterprise critical assets to the Cloud, there are challenges particularly related to security and privacy. It is major that Cloud Users understand their security and privacy needs, based on their specific context and select cloud model top fit to support these needs. The literature provides works that focus on discussing security and privacy issues for cloud systems but such works do not give a detailed procedural approach to elicit security and privacy requirements neither methods to select cloud deployment models based on satisfaction of these requirements by Cloud Service Providers. This work advances the current state of the art towards this direction. In particular, we think necessities engineering concepts to elicit and analyze security and privacy requirements and their associated mechanisms using a abstract framework and a systematic process. The work introduces assurance as evidence for satisfying the security and privacy requirements in terms of completeness and reportable of security incident through audit. This allows standpoint cloud users to define their assurance requirements so that proper cloud models can be preferred for a given context. To show our work, we present results from a real case study based on the Greek National Gazette.

INTRODUCTION

Migrating into the cloud certainly gives organization tangible competitive advantages due to significant cost savings, improved degree of scalability, flexibility and resource pooling availability. Moreover, organizations can take advantage of Infrastructure, Platform or

Software as a Service deployment models [1] and a range of service models to choose from – Public, Private, Hybrid and Community. However, there are many uncertainties about the migration process, specifically related to the dependency of an outside provider for the existing business model, data usage and leakage, lack of understanding about the cloud, and many more Security and privacy are major concerns for organizations, which hinder cloud adaption as migrating into the cloud means organizations need to store their sensitive electronic assets into the providers' infrastructure. Existing business applications and data are mostly con-trolled through the provider's infrastructure depending on the chosen model, i.e. Saas, PaaS, IaaS, on which users may not have full/any control. Users' data are generally stored in a multi-tenant platform. This scenario introduces extra security and privacy challenges comparing to the traditional computing environment. Lack of monitoring facility of user data incurs less user confidence on cloud based systems. Techniques [2] to analyze the security and privacy issues in the context of cloud computing are different to those provided by the existing literature for traditional computing environments. It is therefore necessary to develop methods that not only identify and analyse security and privacy requirements but also provide certain assurance that these requirements are met by a specific cloud model before undertaking the migration decision. While such initiative have been put

Cite this article as: Yandapalli Khajana Raju & Smt.Y.Triveni, "Assurance of Security and Privacy Requirements for Cloud Deployment Models", International Journal of Research in Advanced Computer Science Engineering, Volume 5 Issue 4, 2019, Page 8-17.

in place in for tradional IT based systems, the literature fails to provide evidence of a framework that fulfills that objective for cloud based services. This paper provides work towards this direction [3-5].

REQUIREMENT ANALYSIS

A Software Requirements Specification (SRS) is a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as functional requirements. In addition, to using cases, the SRS also contains non-functional (or supplementary) requirements. Non-functional requirements are requirements which impose constraints on the design or implementation

FUNCTIONAL REQUIREMENTS

Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). How a system implements functional requirements is detailed in the system design. In some cases, a requirements analyst [6]

NON-FUNCTIONAL REQUIREMENTS

Requirements define how a system is supposed to be. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals" and "quality of service requirements, "and "non-behavioral requirements. Qualities, that is, non-functional requirements, can be divided into two main categories:

1. Execution qualities, such as security and usability, which are observable at run time.
2. Evolution qualities, such as testability, maintainability, extensibility, and scalability, which are embodied in the static structure of the software system.

HARDWARE REQUIREMENTS

System : Pentium IV 2.4 GHz.

- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.

- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- RAM : 256 Mb

SOFTWARE REQUIREMENTS

Operating system : - Windows 7.

- Front End : - JAVA or JSP
- Database : - SQL SERVER 2008
- Tools :- Eclipse IDE

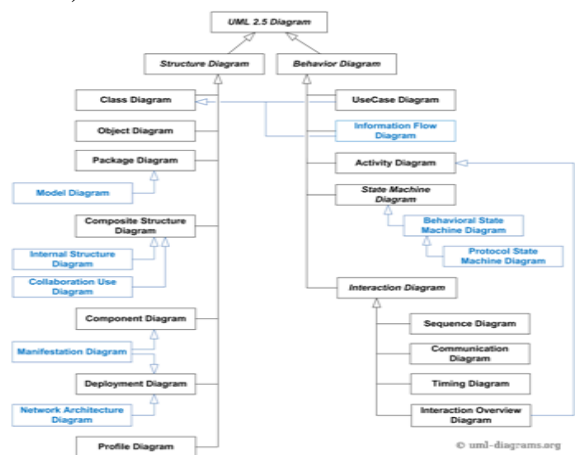
SYSTEM DESIGN

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. he UML has become the standard language used in Object-oriented analysis and design [citation needed]. It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations [7].

UNIFIED MODELING LANGUAGE (UML)

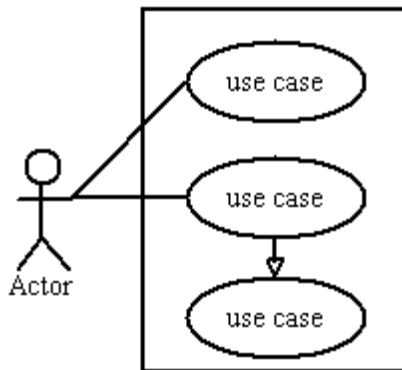
The Unified Modeling Language (UML) offers a way to visualize a system's architectural blueprints in a diagram including elements such as:

- Any activity
- Individual component of the system
- And how they can interact with the other components
- How the system will run
- How entities interact with others (components and interfaces)

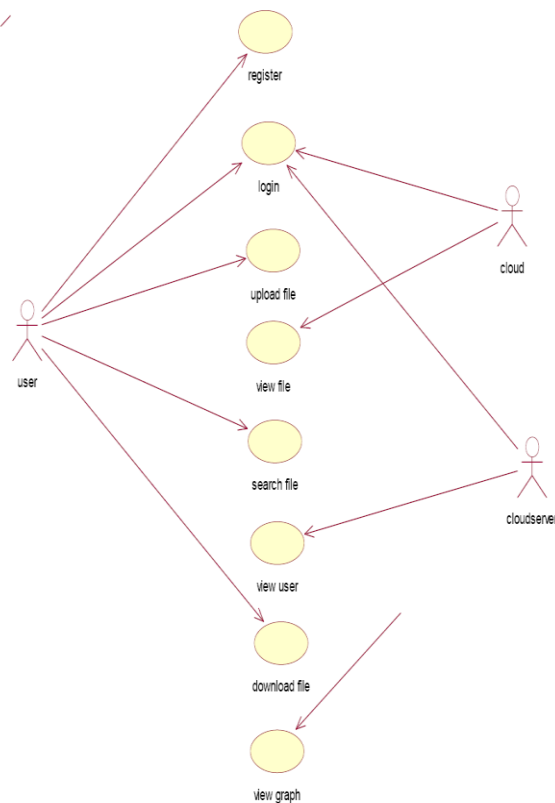


USE CASE DIAGRAM

To model a system the most important aspect is to capture the dynamic behavior. To clarify a bit in detail, dynamic behavior means the behavior of the system when it is running /operating. So only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior [8].



Basic Use Case Diagram



Project Use Case Diagram

INPUT DESIGN

Input Design considered the following things:

What data should be given as input?

How the data should be arranged or coded?

The dialog to guide the operating personnel in providing input.

Methods for preparing input validations and steps to follow when error occurs.

OUTPUT DESIGN

Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements [9].

2. Select methods for presenting the information.

3. Create a document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

Convey information about past activities, current status or projections of the Future.

Signal important events, opportunities, problems, or warnings.

Trigger an action.

Confirm an action.

LANGUAGE SPECIFICATIONS

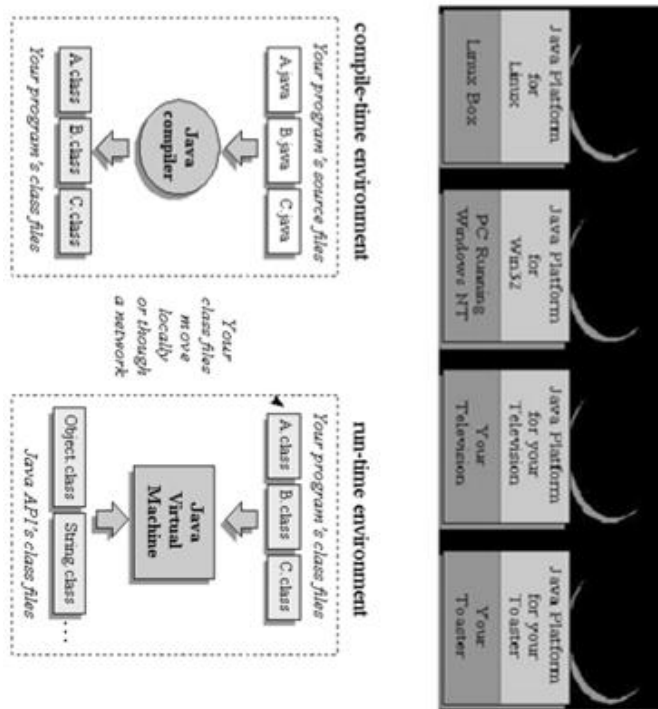
Fundamentally Python is usually used as an interpreted language. This provides very powerful prototyping and live coding opportunities and can also aid in live debugging efforts. It does mean the code can be slower than compiled code, but engineering is always a game of trade-offs.

Python is also a multi-paradigm programming language. Python directly supports both object-oriented and structure-oriented programming. It also embraces many functional programming concepts, especially when

working with groups of things. Overall Python is a very powerful language that can be used in many different ways and in many different areas.

- Python is open source and a broad standard library, which are cross platform compatible on different operating systems like Windows , Mac and Linux.
- Python is portable. Python can run on a wide variety of hardware platforms and has same interface on all platforms with no changes.
- Python provides access interfaces to all major data bases like SQL Server, MySQL, Oracle and PostgreSQL.

Java Technology



Features of Java:

Platform Independence:

The Write-Once-Run-Anywhere ideal has not been achieved platforms usually required, but closer than with other languages.

Object-Oriented:

Object-oriented throughout - no coding outside of class definitions, including main (). An extensive class library is available in the core language packages.

Compiler/Interpreter Combo:

Code is compiled to byte codes that are interpreted by Java virtual machines (JVM).

This provides portability to any machine for which a virtual machine has been written.

The two steps of compilation and interpretation allow for extensive code checking and improved security.

Robust:

Exception handling built-in, strong type checking (that is, all data must be declared an explicit type), local variables must be initialized.

Security

No memory pointers

A program runs inside the virtual machine sandbox.

Array index limit checking

Automatic Memory Management:

Automatic garbage collection

Memory management handled by JVM.

Limitations of C & C++ eliminated:

No memory pointers No preprocessor

Array index limit checking

JAVA CODE

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package design;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class download extends HttpServlet {
SimpleFTPClient client;
/**
 * Processes requests for both HTTP
 * <code>GET</code> and
 * <code>POST</code> methods.

```



```

if (output != null)
fos.write(output);
}
byte[] output = cipher.doFinal();
if (output != null)
fos.write(output);
fis.close();
fos.flush();
fos.close();
}
// <editor-fold defaultstate="collapsed"
desc="HttpServletRequest methods. Click on the + sign on the
left to edit the code.">
/**
 * Handles the HTTP
 * <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error
 occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request,
HttpServletRequest response)
47
throws ServletException, IOException {
processRequest(request, response);
}
/**
 * Handles the HTTP
 * <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error
 occurs
 * @throws IOException if an I/O error occurs
 */
@Override

```

```

protected void doPost(HttpServletRequest request,
HttpServletRequest response)
throws ServletException, IOException {
processRequest(request, response);
}
/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
return "Short description";
} // </editor-fold>
}

```

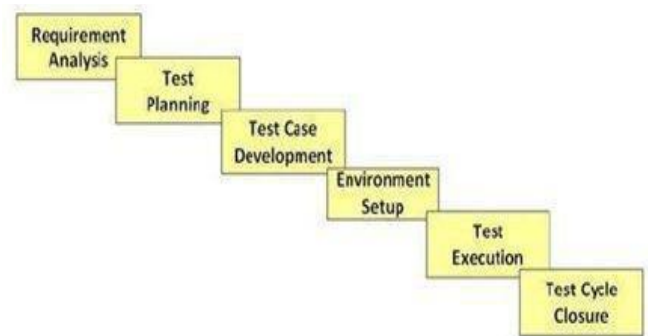
TESTING

Software testing can also be stated as the process of validating and verifying that a software program/application/product:

1. meets the business and technical requirements that guided its design and development;
2. Works as expected; and
3. Can be implemented with the same characteristics.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted.

Software Test Life Cycle



Test Planning

This phase is also called Test Strategy phase. Typically, in this stage, a Senior QA manager will determine effort and cost estimates for the project and would prepare and finalize the Test Plan Activities:

- Preparation of test plan/strategy document for various types of Testing
- Test tool selection.
- Testing Test tool selection

Test effort estimation

- Resource planning and determining roles and responsibilities.
- Training requirement

Test Environment Setup

Test environment decides the software and hardware conditions under which a work product is tested. Test environment set-up is one of the critical aspects of testing process and can be done in parallel with Test Case Development Stage. Test team may not be involved in this activity if the customer team provides the test environment in which case the test team is required to do a readiness check (smoke testing) of the given environment.

Activities:

- Understand the required architecture, environment set-up and hardware and software requirement list for the Test Environment.
- Setup test Environment and test data
- Perform smoke test on the build

Test Execution

During this phase test team will carry out the testing based on the test plans and the test cases prepared. Bugs will be reported back to the development team for correction and retesting will be performed.

Activities:

- Execute tests as per plan
 - Document test results, and log defects for failed cases
- Map defects to

test cases in RTM

- Retest the defect fixes
- Track the defects to closure

RESULTS

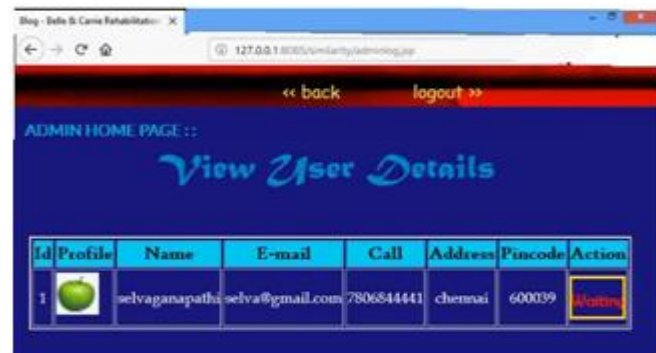
Login Portal



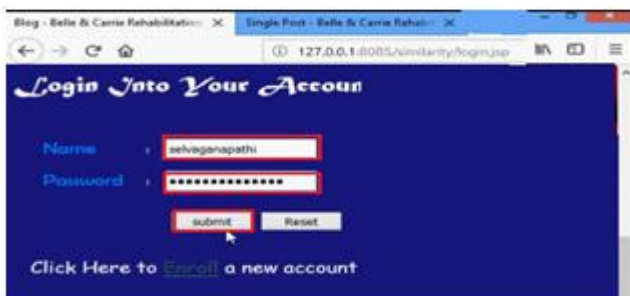
Uploading JSP FILE Using Admin Account:



Creating and Authorize the New User:



Generating Secret Key and downloading the doc from it



Save the below secret key..



CONCLUSION

Cloud migration is one of the most important concerns nowadays for both private and public organisations since due to the recent financial situations every organisation is aiming on cost reductions without losing efficiency and service quality. However, before migrating services, data or infrastructure into the cloud, it is necessary to realise and understand the migration needs and risks that cloud migration hinders. These risks vary among organisations especially due to the variability of information as well as the type of cloud services each organization wishes to use. Finally, the selection of the respective cloud model that will be adopted plays an important role on the potential risks that the organization might face as well. Thus, the role of security privacy is very important for an organization to decide which cloud solution fits best its needs and requirements.

REFERENCES

- [1]. P.J. Bruening, and B.C. Treacy “Privacy & Security Law Report: Privacy”, Security Issues Raised by Cloud Computing. The Bureau of National Affairs, 2009
- [2]. M. Ouedraogo and H. Mouratidis “Selecting a cloud service provider in the age of cybercrime”, Computers & Security, Special issue on Cybercrime in the Digital Economy, vol.38, pp.3-13, Elsevier, 2013
- [3]. P.G. Dorey and A. Leite “Commentary: Cloud computing – A security problem or solution?” Information Security Technical Report, vol. 16, no. 3-4, pp. 89-96, Elsevier, 2011
- [4]. AICPA “Statement on Auditing Standards (SAS) n°70”, from http://sas70.com/sas70_overview.html, 2012
- [5]. NIST, US Government Cloud Computing Technology Roadmap Volume II Release 1.0 (Draft), Useful Information for Cloud Adopters, November 2011



[6]. CSA, Cloud Controls Matrix V.3.0.1,2014<https://cloudsecurityalliance.org/research/cmm/>

[7]. M. Ouedraogo, E. Dubois, D. Khadraoui, S. Poggi and B. Chenal “Adopting an Agent and Event Driven Approach for Enabling Mutual Auditability and SecurityTransparency in Cloud Based Services, In Proceeding of CLOSER 2015

[8]. M. Ouedraogo, D. Khadraoui, B. Rémont, E. Dubois, H. Mour-atidis: Deployment of a Security Assurance Monitoring Framework for Telecommunication ServiceInfrastructures on a VoIP Service. In Proceedings of NTMS 2008: P1-5

[9]. M. Theoharidou, N. Papanikolaou, S. Pearson and D. Gritzalis, “ Privacy Risk, Security, Accountability in the Cloud” proceed-ings of IEEE InternationalConference on Cloud Computing Technology and Science, 2013