A Secure Multi-Tenant Model for SaaS System

Atul Singh
School of Computer science
VIT University Chennai
Campus Chennai, Tamil Nadu-600048

Sharath Kumar J.
Professor, School of Computer science
VIT University Chennai
Campus Chennai, Tamil Nadu-600048

A Muralidhar
Professor, School of Computer science
VIT University Chennai
Campus Chennai, Tamil Nadu-600048

ABSTRACT

World Wide Web is working as a fuel to the systems like SaaS. SaaS shares both the application delivery model and the business model. It provides the customers to access the application using a web browser. it provides better scalability to users. SaaS[2] applications mostly support multi- tenant system to provide the service to many customers at a single instance. It provides users to access the database application server in authentication environment, but it has been found that it is scalable but not much secure for all the customers accessing the same functionalities, it may occur to the information hacked system. So in this paper I am proposing one approach to handle the multi-tenant environment and secure system with the privacy preservation model approach for the multi-tenant support SaaS system to increase the security and scalability.

General Terms

Security, SaaS in Cloud, Emerging Web technologies.

Keywords

Web3.0, SaaS, Multi-Tenancy, Privacy Preservation.

1. INTRODUCTION

In Todays, stage of World Wide Web, circulation of content is no longer just a top-down process. Application & web sites like-Flicker, Wikipedia, you tube and my space, enable user to share and create the content with each other, instead of simply passive consumes it. As compare to the previous web version application development has been simplified such as Google Apple. Going ahead, the World Wide Web is likely to undergo important enhancements due to the extending efforts of diverse ecosystem players, programs products/services vendors, apparatus OEMs, and Operators.

Multiple developments are happening at the same time in an interconnected manner and that they promise to reshape the web considerably.

Today, Web Services and internet have been used generally for Enterprise Integration because it is very effective for connecting the different web applications. The current web technologies named as web3.0, have shown some good efforts to provide facilities of expert systems but it also have limitation such as sometimes slow response, information leaks and consumption etc. for example the Google search engine capabilities previously we can only find the web search key word based data information called as static search but we can find some part of the dynamic data with the help of emerging technology named as web 3.0.dyanamic data finding is called as a deep web searching. So, the technology is improving the user experience, accessibility and responsiveness. Here are some current web technologies.

1.1 Intelligent Application

Search tools will also benefit throw semantic technologies that can process the descriptions added with web documents it will help search programs to identify the content of the search and return the more relevant query based result. Web 3.0[1] provides the development of mashups, which are web pages or applications that se data/functionality for multiple external resources to offer enhanced services to consumers.

1.2 Open Source and Open API Client Based Model

Open Source code encourage developers to do the work in minimum time it helps them to play a more active role in developing new consumer applications like, Google Maps API allows developers to assemble interactive Maps from Google into their own web sites. Developers can thus mare effectively convey location information to their consumers without developing mapping functionalities. Open platform for mobile devices like android platform, in Ajax in JSP are very familiar with better communications policy. In Enterprise Integration, There is an edge information must be exchanged, shared and communicated in a well-defined format and in real-time with the help of new technologies.

1.3 Cloud Computing Based Supply Chain Application

Cloud Computing is based on the “on demand” processing capabilities for web based services and devices. So web 3.0 is further divided the Cloud Computing [5] into SaaS (Software as a Service) and IaaS [3] (infrastructure as a Service). SaaS provide a delivery method of software. it provides the facilities to deploy the software application on the web as a part of service and it also support multi-tenancy with better accessibility, flexibility, reliability and efficient system deployment of SaaS is valuable to the Enterprises integration specifically in business process of dynamic support type in the supply chain.

2. SOFTWARE AS A SERVICE (SaaS)

New web technology made a way to provide the facilities of secure and usable way of data transfer. Web content can be easily provide in many forms of Artificial intelligence, dynamic data, Natural language from taking help of new web technologies, Semantic web, Natural language processing (NLP), Mashups & other technologies are a collection of new web technology called as web 3.0 and it is maintained by real business needs SaaS[2] is being sought for reducing IT-related cost.

SaaS [2] is also known as application on demand or software as a service. Implementation of SaaS delivery model has become a popular way to reduce the time and cost associated and maintaining services for the organizations. SaaS enables many services to customers, to take advantage from an application on the pay basis as per the paid services provided and removes the need to install and run the provided application, on the hardware of the customer.

SaaS maintains and shares both the different models named as the application model and the business model. SaaS provide the facilities to the customers as they can use the application on a subscription based model and it also provide the nature of scalability, so the customer can install and run the provided application on his own hardware. Generally, customers access the application from a web based application browser i.e. internet. SaaS is a subscription based model and in this model all maintenance, upgrade & support are provided by the software provider as a part of the service. The capabilities of application customization are usually facilitate to all the customers in a procedural manner, if available at all.

The SaaS model facilitates the high protection of its property, operational management of the environment for frequently working the software package, and usually a it can be done by providing the subscription fees service for revenue repetition, software package providers have different capabilities and applications can be come in different facilities but SaaS applications support several different customers in a single instance of the provided application called as multi-tenant support system.

SaaS needs an additional security service than any other alternative offered models of delivery system. The current security services related to the development of application that involves a stratified approach. In spite of being the software application delivery model, we cannot implement the security at a single “make or break” purpose. The security should be divided in layers into the servers, the network, the data base and the code. Security must be present in both the forms as intrusion detection and prevention. In this paper, we mentioned the security issues especially to the SaaS business model and assumes an operating data and how it will be useful for the development of application for a multitenant environment.

Generally, the modern SaaS architecture comes in the web-based application form.

a. The communication between the user/customer and the service provider using SSL encryption.

b. The communication from the public internet.

c. Many customers looking for a service provider who can provide the solution built using SOA principles.

3. SAAS SECURITY MODEL

3.1 SaaS Working Model

For changing the authorization module we require significant efforts and testing that causes the risk increment and that must be avoided. Every time it is preferred that SaaS provider does not provide the custom identity management interface for each customer. A better idea is to offer a well-tested predefined standard for industry interface. That may need to lead to next development process from the client but it will decrease the going on maintenance process for the overall platform. The basic SaaS model includes these steps:

1. The customer/user tries to access the SaaS supplier and will need to complete the identifying information. For example: in the event SaaS platform is web based, URL or cookie can be the encrypted data.

2. The authenticated process will be done by the customer’s user directory via a call of web service.

3. The user directory of customer will reply back with acknowledge containing the authentication & authorization information.

4. Result will be the successful completion based on the authentication & authorization.

Once the SaaS provider’s system recognized the user successfully, it is necessary for the users to allow only access the data and functions for that they are permitted to access, called as authorization. SaaS platform should well change or addition to its identification strategies. It is necessary to check whether the authorization module is different from the authentication module or not.

The authorization module should log the performing action in every attempt, for success. The exact procedure of authorization is totally dependent upon the SaaS application software architecture. It is better to validate each and every request, requested the customer/user. For the support of Enterprise integration SaaS introduces the new supporting technology like Multi-Tenancy, SAN (Storage area network), and MySQL etc.

3.2 Multi-Tenancy

It is an architectural concept which is used to maximize the effectiveness and user scalability as well as to offer maximize operational scalability and less support costs for the service provider by providing the common user interface, business logics and databases for all customers and users at all layers of the application, multi-tenancy is considered an architectural ideal but it is not required at all layers to realize benefits. It is best model for lower cost service offering, but it leads to security issues in mining the data. Basically there are three approaches known to handle the multi-tenant data.

1. Separated Database

2. Separated Schema

3. Shared Schema

Separated databases are the simplest method for storing tenant data to data isolation. Generally the application code and other computing resources shared between all of the tenants on a server, but each and every tenant has its own set of data which remains logically isolated from other tenants’ data. We can extend the application data model by using this approach by giving each tenant its own database and it can complete the other restoring data backups when sever gets failed. But its cost is higher to implement because it is premium approach and it leads to relatively high hardware and maintenance requirements and cost.

Shared database and separated schemas, this approach shows the multiple tenants in the same database and with each tenant will be having its own tables set which are grouped into a schema specifically created for the tenant. For the creation of schema we use here the SQL server or MY SQL.

Shared schema and shared database, in this approach the database will be same and same set of tables to host multiple tenants' data. A table can include the records from the multiple tenants they can be store in any order.

Among all the three methods the shared schema approach has the lowest backup and the hardware cost because it provides the facility to serve the largest number of tenants that shares the same database tables but it needs additional efforts in terms of security, for ensuring that tenants can never access other tenants data not even in the unexpected bugs or attacks. This approach is efficient when it is necessary that the application be capable of providing a large number of tenants with a small no. of servers.

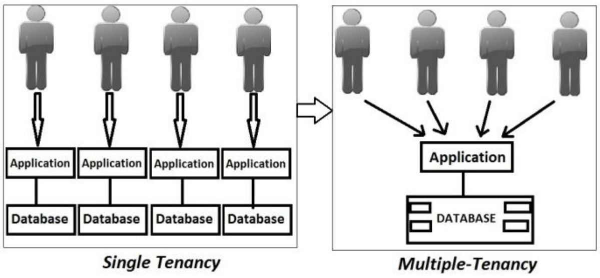

Fig 1: Multi-Tenancy database distribution in SaaS

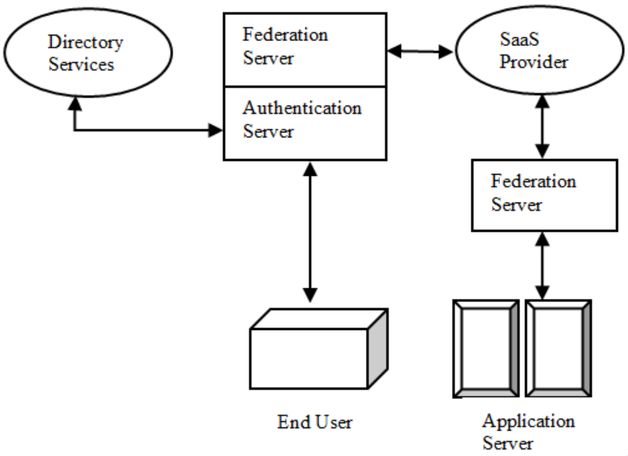

Fig 2: SaaS Architecture

3.3 Privacy Algorithm Approach For SaaS Multi-Tenant System

Society is experiencing exponential growth in variety of data collections. Containing specific information of a person as computer technology, network connectivity and disk storage become more & more reasonable. Data holders operating autonomously and having the limited knowledge are left with confidentiality or national interest. For creating and working with anonymous data to save the data from the others and working privately without any knowledge leak to others, here we propose the method to work and mining the data anonymously.

In this paper we will see how to provide a secure channel to the application database to access the data without conflicting others data tables when we use shared schema approach. One data hiding technique is given below called privacy preservation for data hiding.

Algorithm for the database configuration and encoding decoding system:

1. First the database tables divided into the indexes. Example: index 1 having 1st 200 records and index 2 another 200 records.

2. Indexes are coded using some encryption algorithm (SHA-1, MD5 etc.) Combine in a separated table.

3. One id will be generated for each index to access the new table of coded index.

4. When user identification process completed they will be provided the passcode then we will follow these steps to decrypt and merge the indexes for accessing the tenant's application.

5. Steps:

   i) For each index I_1 and I_2 assign a frequency that will be based on the no. of appearances of the given leading column of the index.

   ii) **For** each joined index M check the passcode for the index

   iii) **If** (index code==passcode).

   iv) **Do** provide the decode key to the user using mobile authentication.

   v) **If** (mobile code == decode code).

   vi) **Do** login and **print**("Access to application database successful").

   vii) **Else** wrong id.

   viii) **Else** not an authorised user

   ix) **End**

6. The server provides the database application access for the particular session and the data hiding technique used by the privacy preservation algorithm for the multi-tenant system, which is given below.

Algorithm for privacy system using private table:

1. Let $\mathbf{C(D_1,...,D_n)}$ is a table of finite no. of tuples and $\mathbf{\{D_1,....,D_n\}}$ is the finite set of attributes of C.

2. Given a table $\mathbf{C(D_1,...,D_n)}$, $\mathbf{\{D_i,...,D_j\}\subseteq\{D_1,....,D_n\}}$ and the tuple $\mathbf{t\in C}$, where $t[D_i,....,D_j]$ is value sequence of $V_i \ldots V_j$, $C[D_i,....,D_j]$ for maintaining duplicate tuples of attributes $D_{i,}.....,D_j$ in C.

3. Find the Quasi identifier written as Q_v , Q_v can be {name, address, Zip, birthdate, gender}, where V can be the voter specified table.

4. So, {name, address, Zip, birthdate, gender} $\subseteq Q_v$ however, {name, address} $\subseteq Q_v$, Because, these values can be appear in external information and can be used for linking.

5. So, in the final table there will be only the quasi identifier attributes are given and rest information will be hide into groups of clusters Example: age =27, can be written in age=25-30, and the age occurs in the groups of 25-30 category.

6. Based on above method we can generate a decision tree of given data set and from that each level can be divided for authorization procedure for data privacy system.

3.4 Algorithmic Approach for SAAS Database privacy

As we know in the privacy preservation algorithm, the database of each user is divided into the clusters of information. So that any other user connected to that service provided by the same vendor, will able see only the out most information the hidden information will not be shown to other users. This proposed method will contain these steps.

1. User sends the request to the server through the web browser the server will reply, and ask for the authentication.
2. The customer will login into the system for the identification of user using its username & password which is provided by the software vendor.
3. Connection will be created, when the customer want to submit the query of finding the relevant data or getting the user information from the software the local authorization server again ask for identification of user.
4. The authorization server will send the mobile authentication code by email or in mobile to identify the customer. If customer replies with the correct information then he/she can be provided the permission.
5. This will inform the server and make the correct entry of log by using the privacy reservation system.
6. Payment will be charged to customer based on the payment mode or subscription.

4. FIGURES

Here figures are given as:

Fig 3: Improved security model for SaaS.

Fig 4: Database and application server operation.

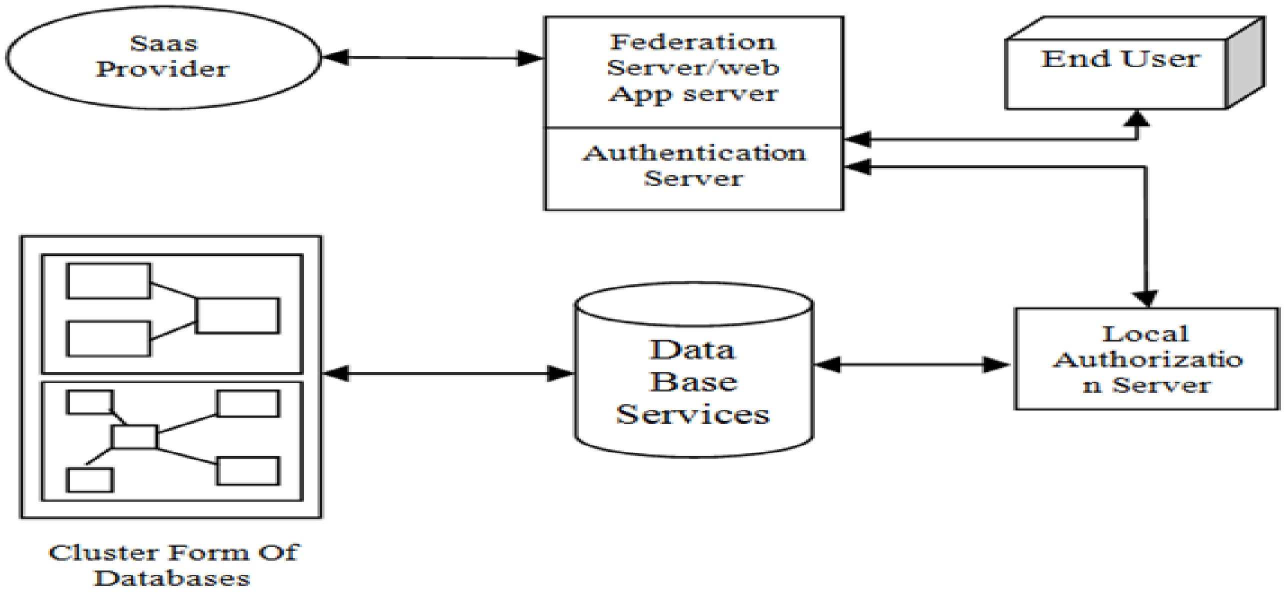
Saas Provider
Federation Server/web App server
Authentication Server
End User
Data Base Services
Local Authorizatio n Server
Cluster Form Of Databases

Fig 3: Improved Security Model For SaaS

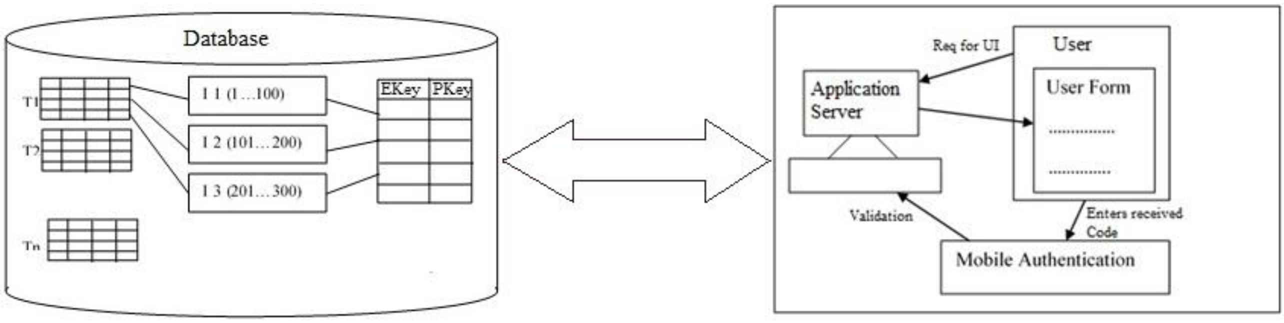

Fig 4: If necessary, the images can be extended both columns

5. CONCLUSION AND FUTURE WORK

In this paper, we propose the security solution for based SaaS system. Security is one of the important components of a SaaS system. It is the core and it must be considered and integrated from the starting day of architecting a SaaS application. For the security, every time monitor your logs. Multi-tenant support system is a very good architecture introduced for providing the scalable, flexible and cost effective system for the SaaS, but as its scalability increases, the data security in the SaaS will be decreased. So, in this paper we are proposing an algorithmic system approach for the SaaS technology with the multi-tenant supportability and security. Using this algorithm approach the SaaS multi-tenant support system will become more power full in security aspects. Here we are proposing few key components: every SaaS enabled system must have the authentication and authorization server to continuous log information that creates a secure channel for customer to vendor interaction or user to user interaction in SaaS system.

This system can be implemented because it will be providing the better privacy system for hiding the information in SaaS and work effectively.

6. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards We would like to sincerely acknowledge the encourageous efforts of authors who provide the path to find something new in web technology and learning opportunity.

7. REFERENCES

[1] Web3.0description-http://www.cruizine.com/?s=web+3.0

[2] Software as a Service Is Gaining Ground, http://www.technologyevaluation.com/Research/ResearchHighlights/CRM/2006/03/research_notes/TU_CR_PJ_03_14_06_1.asp (2006).

[3] http://searchcloudcomputing.techtarget.com/definition/Infrastructure-as-a-Service-IaaS

[4] http://whatis.techtarget.com/definition/multi-tenancy.

[5] http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031.

[6] Software as a service white paper “A scan safe white paper ”(2008).

[7] L. Sweeney:- a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), 2002; 557-570.

[8] Securing a multitenant SaaS application (2008)

[9] Cisco Software-as-a-Service (SaaS) Access Control [Cisco Any Connect Secure Mobility Client] - Cisco Systems.htm.

[10] http://msdn.microsoft.com/en-us/library/aa479086.aspx