

An OSGi Based SaaS Architecture that Supports Code Level Customization

1. Team Members

15855860	CHEN, Xiao Yu	erickwebsun@gmail.com
14885271	Sharma, Chandan	chandan202.alive@gmail.com
14854389	LI, Mao Chuan	mao.chuan.li@gmail.com

Table 1 Team Members

2. What is the project about?

The project is about Cloud computing, specifically in the field of SaaS – Software as a Service.

SaaS is a new software delivery model in which end users access the software services on internet, normally with a single web browser instead of buying a hard copy of the software and running it on his own computers(Kulkarni et al., 2013). This model has greatly reduced the costs of both software providers and consumers, thanks to the new cloud computing technologies, especially virtualization, and the transition of operation and maintenance tasks from end users to SaaS providers(Bezemer & Zaidman, 2010).

One of the most important characteristics of SaaS software is Multi-tenancy support that one instance of the software could be used to serve multiple groups of users with different functions, as if each of them thinks they are using the system solely without awareness of any other users. In this context, we call the group of users as a tenant.

To satisfy the above requirement and challenge, the software has to support high degree configuration and customization of the software on the fly without affecting any tenants on the same system, meanwhile keeping their data isolated from each other.

Nitu (2009) abstracted 4 levels of configurability that a SaaS software platform usually has:

1. User Interface – the look and feel of a software, such as the logo of the software should be customized for each tenant.
2. Workflow – the behaviour of the application should be customized to meet the tenant's specific requirements.
3. Data – the data model of each tenant might be slightly different, so the database structure should allow the tenants extend with more attributes (in this case, only extension is allowed).
4. Access Control - each tenant should be able to create users and assign different roles on them.

As a team after a lively discussion, we concluded there should be as a 5th level of configuration: CODE, because SaaS can never provide all features required by all tenants. In this project, we will mainly focus on the configurability of SaaS software, and try to create a novel architecture that allows the 5th CODE level extension.

3. What are you going to investigate?

- First, we will look at OSGi platform to see how it supports model programming, and how to construct independent services on it;
- Secondly, we will look at Java platform security, and study how we can utilize the security features to protect and limit the codes/bundles;
- Thirdly, we will investigate the OSGi security model;
- Lastly, we need devise a method for tenant users to deploy their code to the OSGi based SaaS platform.

4. What is your research problem?

Our research question is **“How to allow SaaS tenants to extend the SaaS software platform with new CODE to best suit their requirements without sacrificing the platform security?”**

No one software is perfect in this world, there must be a missing feature in a SaaS platform for a tenant. Some software like Firefox designed an architecture allowing users to develop add-ons and easily integrate the new features to extend the Firefox to satisfy the various requirements. SaaS software is no exception, but all SaaS softwares are reluctant or repulsive to enable this extensibility, primarily because of the security concern.

We think extensibility is a hard problem to solve, but it is inevitable in a modern software design, especially a SaaS software which is to serve hundreds of thousands of different kinds of users.

5. Who is going to do what?

All of us three have different backgrounds and skill set, so we will take different responsibilities of the project:

- Mao Chuan – is good at Java programming and OSGi platform, so he will be responsible for the architecture setup;
- Chandon – is good at model construction and data persistency, so he will use his Turing machine model to verify the system;
- Xiao Yu – has rich experience of web development, so he will help design the web pages.

Overall we will design the architecture as a team, and implement the core functions according to the skill sets. Since the project is heavily dependent on Java programming language, so learning Java would be an important task for both Chandon and Xiao Yu, because the coding work is expected a lot.

6. What is the plan (Gantt chart)?

At this time, we don't have a clear task list to complete the project, and not sure what the final prototype could be. We would like to follow an agile development method and try to finish the following user stories as many as we could in this semester. The user stories are managed online with trello.com website. With more knowledge we achieve on the project, the list may grow and vary.

<https://trello.com/b/xuOfwXII/an-osgi-based-saas-architecture-project>

1. As a SaaS provider, I need to start up a single instance of OSGi application so that I can provide services to all my tenants;
2. As a SaaS provider, I need to create and publish an Add-on specification for tenants to extend the platform and suit their own requirements.
3. As a SaaS provider, I need to publish all my services tenants could call directly on my platform.
4. As a SaaS provider, I need to validate the application from tenant in depth and make sure it will not impact the other tenants nor cause any damage to my SaaS platform.
5. As a SaaS provider, I need give clear feedback to tenant for the deployment of their applications.
6. As a SaaS, I can find and download the source code of the SaaS platform, and the Add-on specification.
7. As a SaaS tenant, I can create an Add-on application in compliance with the specification;
8. As a SaaS tenant, I need to sign my Add-on application.
9. As a SaaS tenant, I can deploy my Add-on application to SaaS platform from a web page or any other way;

7. What methodology are you going to follow?

As above mentioned, we shall follow an agile development methodology to develop the project. With the modern social media WhatsApp, we can synchronize our status and keep in touch at any time. With Trello online task management system, we can manage our tasks in a central place, and each of the tasks could be assigned to each team member. Everyone can have a general picture of the project status.

For source code management, we will use online github.com VCS system to manage the project source code.

8. What have you done already and what are the next-steps?

We have studied in depth the concept of SaaS with the help of dozens of SaaS related slides on internet. Also we looked through some SaaS configuration related articles and understood the state of the art of SaaS configurability.

An initial study of Java security has been conducted, and relevant OSGi security material has been found, so that the assumed target architecture seems feasible in theory.

Next we will further examine the Java and OSGi security model and devise an Add-on application specification, and find a java source code surveillance tool to help parse and validate the application tenants might upload. In the meantime, we will finalize the architecture design of the system, and decide the minimal set of services we should deliver for the project.

9. References

- Bezemer, C.-P., & Zaidman, A. (2010). *Multi-tenant SaaS applications: maintenance dream or nightmare?* Paper presented at the Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE), Antwerp, Belgium.
- Kulkarni, G., Shelke, R., Palwe, R., Khatawkar, P., Bhuse, S., & Bankar, H. (2013, 4-6 July 2013). *Multi-tenant SaaS cloud*. Paper presented at the Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on.
- Nitu. (2009). *Configurability in SaaS (software as a service) applications*. Paper presented at the Proceedings of the 2nd India software engineering conference, Pune, India.