

Multi-tenancy in Software as a Service Applications

Eveliina Pakarinen
University of Helsinki
Helsinki, Finland

Abstract—Multi-tenancy is a high level architectural pattern which can be used in cloud computing environment when offering applications using Software as a Service business model. In multi-tenancy pattern service provider hosts a single instance of an application on his or her infrastructure and this application is accessed by multiple tenants who share the resources of the infrastructure. The use of multi-tenancy pattern brings multiple benefits both to the service provider and to the customers but there are also some complexities that affect the use of multi-tenancy. Although multi-tenancy is a popular paradigm it is still a relatively new topic in scientific literature and the research domain of multi-tenancy is not yet mature which can be seen from the lack of industrial experience reports on multi-tenancy. In this paper the key concepts of multi-tenancy in SaaS applications are presented and some concerns that need to be taken into account when developing a multi-tenant SaaS application are discussed. A couple of recently proposed example frameworks for developing multi-tenant applications are also presented in order to describe the current situation in multi-tenancy research.

Keywords: multi-tenancy, Software as a Service, architectural pattern

I. INTRODUCTION

Software as a Service (SaaS) is a software delivery and business model where an application is provided as an on-demand service for multiple users through Internet [1, p. 1]. In SaaS business model the service provider maintains the application and offers the software as a service to the customers [2, p. 1]. By using software offered by a third party companies can use various IT services without maintaining or purchasing their own IT infrastructure [2, p. 1].

Multi-tenancy is a high level architectural pattern which can be used when offering applications as Software as a Service in cloud computing environment [3, p. 1]. In multi-tenancy pattern the service provider hosts a single instance of the software product on his or her infrastructure and multiple customers, so called tenants, access the same instance of the software [2, p. 1]. A tenant is the organizational entity which rents a multi-tenant SaaS solution [2, p. 2]. A tenant groups typically multiple users of the same organization and these users are the stakeholders in the organization.

There are multiple benefits for the service provider when using multi-tenant architecture pattern when implementing SaaS applications. The first benefit is that the application deployment becomes easier because only one application instance has to be deployed [2, p. 1]. In multi-tenant model multiple customers access the same software instance and they do not need own dedicated instance of the software. This means that the customers share the same hardware resources when using multi-tenant SaaS application [4, p. 1]. That

increases and improves the utilization rate of the hardware which is the second benefit of the multi-tenant model [2, p. 1].

These two benefits reduce the software delivery costs for the service provider which can help improve the profit margin [5, p. 1]. The reduced delivery costs enable that software provider can offer the service to the customers at lower service subscription costs [5, p. 1]. That makes multi-tenant applications interesting for customers in the small and medium enterprise segment [2, p. 1].

In addition to the benefits of the multi-tenancy there are also some complexities that come with the multi-tenancy. Challenges can arise in the application development, deployment and management phases [5, p. 1]. Challenges that arise are for example application performance, scalability, security, zero-downtime and maintenance [2, p. 3].

Although multi-tenancy is a popular paradigm it is still a relatively new topic in scientific literature [3, p. 1]. The term multi-tenancy was explicitly mentioned for the first time in a scientific paper in year 2006. Since then many definitions for multi-tenancy have been proposed [3, p. 5]. Also many solutions related to multi-tenancy have been proposed in the multi-tenancy research over the years but there has been very few industrial reports about experiences on multi-tenancy [3, p. 4-5]. This indicates that the research domain of multi-tenancy is not yet mature and that the solutions have not yet been implemented or evaluated. The high amount of proposals and the low amount of industrial reports can also indicate that there is a lack of cooperation between industry and academia in this domain [3, p. 4-5].

In this paper the key concepts of multi-tenancy in SaaS applications are presented and some concerns that need to be taken into account when developing a multi-tenant SaaS application are discussed. A couple of recently proposed example frameworks for developing a multi-tenant applications are also presented in order to describe the current situation in multi-tenancy research.

This paper is organized as follows. In Section 2 the research methods for data collection for this paper are introduced and the research questions are presented. In section 3 an introduction to multi-tenancy and SaaS is given. In section 4 an example architecture framework for multi-tenant architecture is presented. In section 5 the answers to the research questions are discussed. A conclusion is presented in section 6.

II. RESEARCH METHODS

Data the work is based on how was the data collected and how was it decided what to include? What kind of data there is? Why was this particular data chosen?

Research questions! Here or should they be in introduction?

What is the problem for which this seminar paper provides an answer?

III. INTRODUCTION TO MULTI-TENANCY AND SAAS (SOFTWARE/ARCHITECTURE/PATTERNS)

(overview of the research field?)

Basic introduction to multi-tenancy and SaaS because they are not so well known (?). Background information for the user who does not know anything about multi-tenancy.

Maybe comparison of multi-tenancy and multi-user? Figure where multi-tenancy/multi-instance/multi-user are compared.

Maturity levels of SaaS. (Maybe a brief discussion about SaaS and other XaaS services/paradigms?)

Levels of multi-tenancy or different interpretations of multi-tenancy.

Challenges of multi-tenancy

IV. RESULTS: EVALUATION/VALIDATION/REVIEW OF A PROPOSED ARCHITECTURE PATTERN/Framework FOR MULTI-TENANT SAAS APPLICATION

(or maybe comparison to some other framework? How has it evolved through years?)

Answers to the research questions (should be here?)

New results

Neutral analysis based on the data. New ideas/new things/new viewpoints based on the material my unique way to combine the materials to something new?

V. DISCUSSION

How well the research questions were answered?

Discuss my answers to research questions

Limitations that affect the validity of the results

Related work or comparison to other work (maybe)

VI. CONCLUSION

Main points, message

Probably some mention about future work or what should be researched next

REFERENCES

- [1] S. Kang, S. Kang, and S. Hur, "A design of the conceptual architecture for a multitenant saas application platform," in *2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering*, May 2011, pp. 462–467.
- [2] C.-P. Bezemer and A. Zaidman, "Multi-tenant saas applications: Maintenance dream or nightmare?" in *Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE)*, ser. IWPSE-EVOL '10. New York, NY, USA: ACM, 2010, pp. 88–92. [Online]. Available: <http://doi.acm.org.libproxy.helsinki.fi/10.1145/1862372.1862393>
- [3] J. Kabbeldijk, C.-P. Bezemer, S. Jansen, and A. Zaidman, "Defining multi-tenancy: A systematic mapping study on the academic and the industrial perspective," *Journal of Systems and Software*, vol. 100, pp. 139 – 148, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0164121214002313>
- [4] S. Pal, A. K. Mandal, and A. Sarkar, "Application multi-tenancy for software as a service," *SIGSOFT Softw. Eng. Notes*, vol. 40, no. 2, pp. 1–8, Apr. 2015. [Online]. Available: <http://doi.acm.org.libproxy.helsinki.fi/10.1145/2735399.2735412>
- [5] C. J. Guo, W. Sun, Y. Huang, Z. H. Wang, and B. Gao, "A framework for native multi-tenancy application development and management," in *The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (CEC-EEE 2007)*, July 2007, pp. 551–558.