BENAIS Research Symposium
"Variability in Multi-tenant SaaS environments"

Jaap Kabbedijk, MSc.
j.kabbedijk@cs.uu.nl
Utrecht University, Faculty of Science
Department of Information and Computing Sciences

March 28, 2011

Research Abstract

Currently a trend is going on in which product software vendors want to offer their product as a service to their customers [7]. This principle is referred to in literature as Software as a Service (SaaS) [4]. Turning software into a service from a vendor's point of view means separating the possession and ownership of software from its use. Software is still possessed and deployed at the vendor, but used by the customer. The problem of moving a software product from different on-premise locations to one central location, is the difficulties with complying to specific customer wishes. In order to serve different customers' wishes, variability in a software product is needed to offer all specific functionality. Using variability, it is possible to deliver software functionality as optional modules, that can bound at delivery time. Applying this principle can overcome many current limitations concerning software use, deployment, maintenance and evolution [8]. It also reduces support costs, as only a single version of the software has to be maintained [3].

Besides concurring to specific customer wishes, a software vendor should be able to offer a service to a large number of customers, each with their own variability wishes, without running into scalability and configuration problems [1]. The solution to this problem is the use of multi-tenancy within the SaaS environment a software product is offered in. Multi-tenancy can be seen as an architectural design pattern in which a single instance of a software product is run on the software vendors infrastructure, and multiple tenants access the same instance [2]. It is one of the key competences to achieve higher profit margins by leveraging the economy of scale [5]. In contrast to a model incorporating multiple users, multi-tenancy requires customizing the single instance according to the varying requirements among many customers [6]. No techniques are currently documented on how to realize the variability needed in multi-tenant SaaS environments. A catalogue is proposed containing a collection of variability realization techniques (VRTs) that can be used to implement variability in existing SaaS systems or to create variable multi-tenant SaaS applications from scratch. The VRTs will be analyzed, based on criteria identified during several large case studies (e.g. maintainability, testability, cost, etc.). By applying the VRTS documented in this catalogue software vendors will be able to offer their products as a service in a multi-tenant manner. This will enable software vendors to offer a lot of different variabilities of a product, without maintainability or scalability problems.

References

[1] 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)*, pages 88–92. ACM, 2010.

[2] C.P. Bezemer, A. Zaidman, B. Platzbeecker, T. Hurkmans, and A. Hart. Enabling multi-tenancy: An industrial experience report. In *26th IEEE Int. Conf. on Software Maintenance (ICSM)*, 2010.

[3] A. Dubey and D. Wagle. Delivering software as a service, 2007.

[4] N. Gold, A. Mohan, C. Knight, and M. Munro. Understanding service-oriented software. *Software, IEEE*, 21(2):71–77, 2005.

[5] C.J. Guo, W. Sun, Y. Huang, Z.H. Wang, and B. Gao. A framework for native multi-tenancy application development and management. *The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services*, pages 551–558, 2007.

[6] T. Kwok, T. Nguyen, and L. Lam. A software as a service with multi-tenancy support for an electronic contract management application. In *Services Computing, 2008. SCC'08. IEEE International Conference on*, volume 2, pages 179–186. IEEE, 2008.

[7] D. Ma. The Business Model of "Software-As-A-Service". In *Services Computing, 2007. SCC 2007. IEEE International Conference on*, pages 701–702. IEEE, 2007.

[8] M. Turner, D. Budgen, and P. Brereton. Turning software into a service. *Computer*, 36(10):38–44, 2003.