The Journal of Systems and Software 100 (2015) 139–148

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Defining multi-tenancy: A systematic mapping study on the academic and the industrial perspective

Jaap Kabbedijk[a,*], Cor-Paul Bezemer[b], Slinger Jansen[a], Andy Zaidman[b]

[a] Utrecht University, Department of Information and Computing Sciences, The Netherlands
[b] Delft University of Technology, Department of Software and Computer Technology, The Netherlands

ARTICLE INFO

Article history:
Received 18 December 2013
Received in revised form 16 October 2014
Accepted 18 October 2014
Available online 28 October 2014

Keywords:
Multi-tenancy
Systematic mapping study
Definition

ABSTRACT

Software as a service is frequently offered in a multi-tenant style, where customers of the application and their end-users share resources such as software and hardware among all users, without necessarily sharing data. It is surprising that, with such a popular paradigm, little agreement exists with regard to the definition, domain, and challenges of multi-tenancy. This absence is detrimental to the research community and the industry, as it hampers progress in the domain of multi-tenancy and enables organizations and academics to wield their own definitions to further their commercial or research agendas.

In this article, a systematic mapping study on multi-tenancy is described in which 761 academic papers and 371 industrial blogs are analysed. Both the industrial and academic perspective are assessed, in order to get a complete overview. The definition and topic maps provide a comprehensive overview of the domain, while the research agenda, listing four important research topics, provides a roadmap for future research efforts.

1. Introduction

An ongoing growing influence of cloud computing and Software-as-a-Service (SaaS) can be observed in the enterprise software domain (Forbes, 2012). One of the key features of SaaS is the ability to share computing resources in offering a software product to different customers. To benefit from this ability, the architecture of SaaS products should cater for the sharing of software instances and databases. A popular architectural style for achieving this is known as Multi-Tenancy. The concept of multi-tenancy, within the software architecture community, is usually referred to as the ability to serve multiple client organizations through one instance of a software product and can be seen as a high level architectural pattern in which a single instance of a software product is hosted on the software vendor's infrastructure, and multiple customers access the same instance (Bezemer et al., 2010). The specific method for sharing instances (e.g., reentrancy or queueing) is generally not specified within the multi-tenancy pattern. Multi-tenancy allows for the customization of the single software instance according to the varying requirements of many customers (Kwok et al., 2008), contrasting with the multi-user model in which there is no substantial variability (Bezemer and Zaidman, 2010). Also, multi-tenancy is one of the key factors for achieving higher profit margins by leveraging the economies of scale (Guo et al., 2007).

Multi-tenancy has evolved from a number of previous paradigms in information technology. More concretely, starting in the 1960s companies performed *time-sharing*, they rented space and processing power on mainframe computers to reduce computing expenses; often they also reused existing applications (Wilkes, 1975). Around 1990 the *application service provider* (ASP) model was introduced, where ASPs hosted applications on behalf of their customers. ASPs were typically forced to host applications on separate machines or as separate processes (Smith and Kumar, 2004). Finally, the multi-user model is most-known from popular consumer-oriented web applications (e.g., Facebook) that are functionally designed as a single application instance that serves all customers (Bezemer and Zaidman, 2010). Multi-tenant applications represent a natural evolution from these previous paradigms. Similarly, around the year 2000, Bennett et al. (2000) set out a vision for service-based software applications, in which they note a number of essential ingredients for what we now call multi-tenancy, namely: demand-led provisioning of software services and a high degree of personalization of software.

In the domain of software (and hardware) systems, the topic of multi-tenancy in scientific literature appeared relatively recently, with the first explicit mention of the term in a paper by Chong

* Corresponding author. Tel.: +31 302536311.
E-mail addresses: j.kabbedijk@uu.nl (J. Kabbedijk), c.bezemer@tudelft.nl (C.-P. Bezemer), slinger.jansen@uu.nl (S. Jansen), a.e.zaidman@tudelft.nl (A. Zaidman).

http://dx.doi.org/10.1016/j.jss.2014.10.034
0164-1212/

and Carraro (2006) in the MSDN Library. Within multi-tenancy, the hardware and software infrastructure is shared and a hosted application can serve user requests from multiple companies concurrently (Guo et al., 2007). Multi-tenancy is regarded a key attribute of well-designed SaaS applications by Chong and Carraro, who developed a commonly used maturity model of SaaS that distinguishes four maturity levels. The last two maturity levels in this model describe multi-tenancy, rendering it as a requirement for a mature SaaS application. Multi-tenancy is not confined to specific resources, but is applicable at different levels in a system's architecture, for example on a database or instance level. As a result, various approaches to a multi-tenant architecture are possible (Osipov et al., 2009; Natis, 2008).

Most academics and practitioners agree multi-tenancy enables software vendors to serve multiple customers from a single online product, but specific implementations differ significantly, leading to an indistinct understanding of the different levels to which multi-tenancy can be applied. This varying definition of multi-tenancy is confusing *among* academics and practitioners, but it also complicates the communication *between* them, caused by the different understanding of multi-tenancy among them. Oracle, for example, looks at multi-tenancy primarily from a database perspective (Oracle, 2009), while Microsoft looks at multi-tenancy more from a functional perspective (Microsoft, 2012).

The goal of this paper is to chart and bridge these varying definitions and the views from both industry and academics on multi-tenancy. First, there is a need for an overview of the different definitions of multi-tenancy, followed by a clear analysis of what is shared among the different definitions. Having such an overview will improve the understandability of multi-tenancy and allows parties to be more aware of the varying nature of the definitions on multi-tenancy at this moment. Establishing common ground also allows us to define research challenges to guide future research in the domain of multi-tenancy. This paper aims at satisfying these needs by performing a structural search in academic literature and blog posts, as described in Section 2. All search data is analysed (Section 3) and an overview of the results can be found in Section 4. The different perspectives on multi-tenancy emerging from the results are synthesized to one overarching definition (Section 5). To structure future research, a research agenda containing seven areas of interest is proposed (Section 6), followed by a conclusion and discussion in Section 8.

2. Research method

In order to get an overview of the current state of multi-tenancy literature and get insight on the interpretation of multi-tenancy from different perspectives a set of research questions has been constructed. The main research question (RQ) is as follows:

RQ: *How to characterize multi-tenancy?*

The main research question is addressed by answering the sub research questions (SubRQs) listed below. Each question focuses on a different perspective on the characterization of multi-tenancy.

SubRQ1: *What comprehensive definition for multi-tenancy can be constructed based on current literature?*
Rationale: Multi-tenancy is not a new concept, and many different definitions already exist. Since these definitions may reflect different perspectives on a software product and focus on different elements, an overall definition should be developed.

SubRQ2: *How is multi-tenancy interpreted in academia and industry?*
Rationale: The use or understanding of the concept of multi-tenancy in industry could differ from the common use in academia. This possible chasm between academia and industry inhibits cooperation and communication between both domains. To examine this, not only academic papers are analyzed, but also 300 internet blog results are used to be able to compare uses in both domains.

SubRQ3: *What future research topics can be defined based on current literature?*
Rationale Since the domain of multi-tenancy research is rather young and scattered, there is a need for guidance on future research. Several research topics are distilled from the academic literature.

The questions are answered based on the academic papers and public blogs aggregated by the systematic search and selection process that is followed in this research. Two different datasets are gathered and analyzed using a Systematic Mapping Study (SMS) approach. The first dataset is gathered from within the academic domain, while the second dataset is composed from blogs from the industry domain. An SMS is the appropriate method when trying to answer a general research question on a certain topic (Kitchenham et al., 2010) and provides a detailed overview of the topic. A previous paper by Anjum and Budgen (2012) was used as a guideline for reporting the mapping study.

2.1. Academic literature collection

In order to identify, evaluate and interpret the available literature relevant to a particular topic in an unbiased, objective and systematic way, common practice is to perform a Systematic Literature Review (SLR) (Budgen et al., 2008). The proper execution of an SLR is still something that is not done frequently in the field of Software Engineering (SE) (Kitchenham et al., 2009). This is probably caused by the fact that an SLR is time-consuming and should be performed rigorously within a mature research domain. However, if little evidence exists or the topic is too broad or scattered, then a Systematic Mapping Study (SMS) is the appropriate method (Kitchenham, 2004). An SMS is used to map the field of a certain topic, instead of answering a specific research question (Petticrew and Roberts, 2009). Since the research domain of multi-tenancy is not mature yet and initial search shows definitions differ significantly, this study uses an SMS to get a overview of the concept of multi-tenancy. This paper presents an SMS in which the different perspectives on multi-tenancy are examined.

The systematic mapping study was performed according to the phases described by Petersen et al. (2008). First, a search for relevant publications was performed, second a classification scheme was constructed, and third, the publications were mapped. The details of the different steps are described below. The first phase consisted of literature retrieval. The steps and the resulting dataset size are as follows:

1. *Search execution*—Dataset retrieval from using the search query on the following databases: ACM, CiteSeerX, IEEE, ISI, Science Direct, Scopus, SpringerLink, and Wiley. Since Google Scholar aggregates from all the databases listed, it was excluded from the search to minimize the number of duplicates. The search has been performed using the following keyword query:

   "multi-tenancy" OR "multi-tenant" OR multitenancy OR multitenant OR "multi tenancy" OR "multi tenant"