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"

2. *Paper screening*—Consists of a check for completeness, relevance, and compliance to the inclusion and exclusion criteria. Included papers are peer reviewed academic papers. Excluded are non-English papers and duplicates not identified in the previous step.
3. *Filtering on title and year*—Deletion of papers written before 2000 because the term multi-tenancy in this field was non-existent before that year. Papers describing multi-tenancy unrelated to IT (e.g., related to housing) are excluded.
4. *Filtering on abstracts*—Papers that merely use the term but do not actively discuss multi-tenancy are removed as well.
5. *Filtering on full text*—The final selection was based on the criteria that the paper must either explicitly state a multi-tenancy definition or refer to one instead.

The results of conducting all five steps were systematically logged in a central database accessible by all authors. After each step, 10% of all papers have been selected by querying every 10th entry in the database, and checked for inter-rater agreement by all authors. If a paper was rated differently by another author, the discrepancy was discussed and corrected. When more than one discrepancy was identified, the step was redone. This inter-rater agreement check was done in order to ensure construct validity of the data gathering (Eisenhardt, 1989).

2.2. Industrial literature collection

The gathering of industrial literature (i.e., blogs), was performed in order to provide a sanity check for the academic literature. The results were not used explicitly for the construction of the multi-tenancy definition or research agenda, but serve to examine potential different interpretations of multi-tenancy between industry and academia. For the industrial perspective of this survey, we have mirrored the process of the Systematic Mapping Study for scientific literature. We use the same three phases that Petersen et al. (2008) describe for the traditional SMS, being:

1. *Search execution*—Consists of dataset retrieval from using the search query. We use the same search query as for the scientific literature, but this time applied it to the traditional Google search and the Google Blog search (www.google.com/blogsearch). The search string used was:

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

   The search results are limited to the first 300 results of the traditional Google Search and to 100 of the Google Blog search. This cut-off is instigated to keep the results manageable, but we also found that around these thresholds the search results become decreasingly relevant (e.g., the traditional Google search started returning results that were not-related to multi-tenancy in the area of computer science).
2. *Website categorization*—The first 100 entries of the traditional Google search are screened and subsequently the second and fourth authors of the paper established an initial categorization of the web sites that were encountered. The categorization is first performed by both authors independently, after which the initial sets are compared and discussed. Based on discussion, the final set is constructed. Having a website categorization, makes it easier to understand the importance of multi-tenancy in industry and how we could learn from these web sites when considering how multi-tenancy is defined and used in industry.
3. *Inter-rater agreement*—The categorization of the websites is done by the second and fourth author. Both of them categorize half of the website entries. In order to achieve inter-rater agreement 10 websites entries from the second author and another 10 from the fourth author were exchanged and re-classified by the other.
4. *Investigation of full text*—Because a web site typically does not have the same structure as a scientific paper, we screened the full text of each web site in full in order to determine (1) whether the search result is within the scope of this study and (2) in which category the website should be placed. The scope was determined to be everything related to IT.

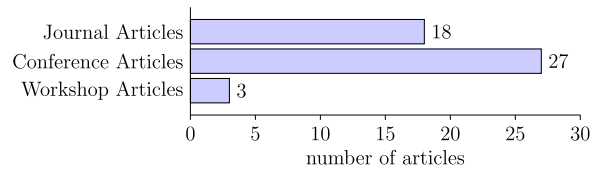

Fig. 1. Publication outlets for academic articles on multi-tenancy.

Whenever differences existed in the classification done by the second and fourth author, an agreement is reached through discussion. The classification result and similar classifications are adjusted according to the new joint interpretation.

3. Classification

3.1. Academic literature classification

In this section, the analysis of the academic literature is illustrated. An overview of the results per phase in the systematic mapping study is presented below, followed by a top-down approach for the literature analysis.

1. *Search execution*—The search resulted in 1371 papers. After duplicate removal based on title, a database of 761 papers was created.
2. *Paper screening*—This phase resulted in 672 applicable papers.
3. *Filtering on title and year*—Resulted in 259 applicable papers.
4. *Filtering on abstracts*—After filtering, 92 applicable papers were identified.
5. *Filtering on full text*—This resulted in 48 applicable papers.

After checking for the inter-rater agreement in each step, small discrepancies between the raters were found. None of the steps, however, had a discrepancy larger than one paper, which meant none of the steps had to be redone. The small level of discrepancy can be explained by the fact both authors are knowledgeable in the area of multi-tenancy and already knew many of the papers published within this domain.

Different publication types are discussed in Fig. 1, showing an overview of the different paper publication outlet types. Conferences clearly play a dominant role in publishing papers on multi-tenancy (27 papers), followed by journals (18 papers). Only three papers were found in workshops.

To further investigate the state of the art in the scientific literature an analysis on the research was performed as well as classification by research type. This overview is useful for identifying gaps in current literature. To classify the type of research approach, six existing distinct research categories were used (Wieringa et al., 2009). An overview of these type of research approaches is presented in Table 1.

Papers were classified using an evolutionary approach, where subjects are selected based on title, abstract and keywords. Papers are categorized and categories are evolved throughout the review using splitting and merging. The analysis of the results focuses on presenting the frequencies of publications for different research categories. An overview of popular and less popular categories can

Table 1
Categorization of 48 papers, listing the number of occurrences (*N*) for each type of paper encountered.

Category	Description	*N*
Solution proposal	Proposes a solution with arguments for its relevance without an evaluation in practice but a proof-of-concept is acceptable.	26
Validation research	Investigates an existing solution and validates it by using a sound scientific approach.	10
Evaluation research	Investigation of a problem or implementation of a technique in practice.	6
Philosophical paper	Introduces a new view on a subject, a new concept, conceptual framework.	5
Experience paper	Explains why or how something has been done in practice. For example lessons learned from projects.	1
Opinion paper	Contains an author's opinion on a subject.	0

be used to identify gaps and possibilities for future research. It also provides a picture about the nature of the scientific material and the maturity of the field. The results from this analysis are depicted in Table 2. Please note the last research category (i.e., Opinion Paper) is not included in the table, since no papers were part of this category.

The list of topics is based on the abstracts of the papers and the keywords listed. It is possible one paper discusses multiple topics, in which case it is listed on all of these topics. A paper, however, is always part of only one research category.

3.2. Industrial literature classification

This section presents the results of the industry literature gathering per phase, followed by a discussion of the analysis.

1. *Search execution*—Among the results were a number of scientific papers, all of which were also part of our search for scientific literature. After removing duplicates, this resulted in 371 entries.
2. *Website categorization*—Eight categories were identified, as shown in Table 3. The first half of the websites was categorized by the second author, the second half was categorized by the fourth author.
3. *Inter-rater agreement*—To validate the choice of categories and evaluate the categorization process, a random sample ($N = 12$) of websites was categorized by both the second and fourth author and compared afterwards. Small changes existed in the classification, mainly due to different interpretation of the categories. In 75% (9/12) of the cases, both authors completely agreed on the categorization (average of 2.33 categories per website). In the three other cases, they at least partly agreed on the categorization. Considering a website can be categorized in a subset of unknown size of 8 different categories, we considered this to be a good level of inter-rater agreement.
4. *Investigation of full text*—All of the 371 entries appeared to be relevant to the concept of multi-tenancy in IT.

Table 3
Categorization of 371 Google search entries, listing the number of occurrences (*N*).

Category	Description	*N*
Non-corporate blog	A software engineer or technology expert writing about multi-tenancy. No (corporate) affiliation is mentioned or could be retrieved.	117
Corporate blogs	White papers mentioning multi-tenancy. This category consists of web pages that are either hosted by a corporation or that explicitly state that the author or text was written from a specific company's perspective. It does not directly advertise the services of the company with regard to multi-tenant technologies, but it describes the company's vision on multi-tenancy.	84
Howto	Web page describing how to implement multi-tenancy. No corporate affiliation or link to a specific product is mentioned.	82
Advertisement	Web page advertising a product or service related to multi-tenancy.	81
Evangelism	Web page containing a strong opinion either in favor or against multi-tenancy	79
Definition	Web page containing a definition (or a discussion on the definition) of multi-tenancy	38
Support forum	Forum discussing multi-tenancy. This forum can be product-specific or product-agnostic. Some support forums are hosted by corporations, others are hosted by StackOverflow, Google Groups, etc.	36
Product manual	Web page describing how to use a multi-tenancy oriented product or service. This category of websites can be linked to a specific product or service.	18

As mentioned in Section 2.2, we started out by analyzing the first 100 entries returned by Google to create an initial categorization of search results. Small changes to the categorization were made while analyzing all search entries. The final categories that we ended up with are listed in Table 3.

Table 3 also describes the criteria that we used for the categorization process. Note that we tried to distinguish "corporate opinions" from "individual opinions" as much as possible, hence

Table 2
Multi-tenancy research topics per research category.

	Evaluation research	Solution proposal	Validation research	Philosophical paper	Experience paper	Total
SaaS	4	19	6	2	1	32
Architecture	4	13	7	3	1	28
Implementation	2	8	2	2	1	15
Database	–	4	6	2	1	13
Balancing &placement	2	6	2	3	–	13
Variability	1	8	1	–	1	11
Infrastructure	1	5	3	1	–	10
Industry evaluation	1	4	1	2	1	9
Quality assurance	1	6	1	–	–	8
Platform development	–	4	2	1	–	7
Security	–	3	1	2	–	6
Standards	–	3	–	2	–	5
Total	16	83	32	20	6	

the many different categories. From the initial search results we removed duplicates, and excluded 14 academic papers and dead website links. This resulted in 371 search entries being investigated, divided over the aforementioned categories. An overview can be seen in Table 3. It should be noted that some search results were categorized in multiple categories, for example, a corporate blog might also contain an explicit advertisement for the product being described.

4. Observations

This section presents a set of observations, based on the results of the Academic and Industrial result classification. All observations were discussed among all four authors and adapted if needed. The observations do not aim to provide a complete list, but rather give a representative illustration of the multi-tenancy domain.

4.1. Academic paper results

Based on the paper classification in Section 3.1, the following observations are made:

Observation 1: Conference oriented—As Fig. 1 shows, around 56% of all research papers on multi-tenancy are published in conference proceedings, compared to 37.5% in journal publications and only around 6.5% in workshop proceedings. The accent on conference publications in not uncommon in the IT domain, but the lack of workshop publications is striking. One such distribution could indicate a very mature research domain, but considering the novelty of multi-tenancy and number of papers published this is unlikely. A more plausible cause is that the domain of multi-tenancy research has no strong community yet and workshops still have to be formed, causing researchers to submit results to conferences and journals, which often have a broader scope.

Observation 2: Many proposals, lack of experience— Table 2 shows a strong emphasis on solution proposals and only one paper reporting on industrial experiences. This imbalance indicates that the research domain is still not mature, and that most of the solutions proposed have not yet been implemented or evaluated. The large difference can also signal the lack of cooperation between industry and academia.

Observation 3: Architecture and SaaS play a big role— Unsurprisingly, the topics of SaaS (32 papers) and architecture (28 papers) are addressed a lot in multi-tenancy research. Multi-tenancy is clearly positioned as an architectural tactic for online software. Since SaaS and architecture refer to the entire software stack, this observation also shows that research focusses on the complete software product instead of just one level (e.g., Database).

4.2. Blog post results

We did a full reading of three categories of web pages, being web pages or blog posts in the categories *non-corporate blog*, *corporate blog*, *definition* and *evangelism*. This reading gave us an impression of some of the advantages, disadvantages and/or issues that practitioners see or have with multi-tenancy. We have translated the impression that we thus got into the following observations:

Observation 1: Different multi-tenancy levels—Some practitioners make a distinction between multi-tenancy at the level of the *infrastructure* (multiple operating system instances on the same physical hardware), at the level of the *platform* (different applications and/or tenants on the same instance of the operation system) and at the *application* level (a single run-time stack is shared with multiple tenants). While not every blog post or website is perfectly clear on this, we observe that most websites on multi-tenancy are actually about the infrastructural or platform level application of multi-tenancy.

Observation 2: Cloud-based nature—For many practitioners multi-tenancy is *evident* in a cloud-based setting (IBM, 2011). This points at two distinct issues with how multi-tenancy is perceived by practitioners. First, a cloud environment is—by its very purpose—a shared platform environment, which in turn indicates that multi-tenancy is seen by many as another way of saying *Platform as a Service* or PaaS. Indeed, in a PaaS setting, tenants can rent a piece of shared platform which can consist of an operating system and standard server applications like a web server, a database, etc. Secondly, in some cases, practitioners were also considering multi-tenancy at the level of software in a cloud-based setting. In this context, practitioners were considering that Software as a Service offerings can be offered more efficiently if the underlying platform is elastic.

Observation 3: Configurability of multi-tenant applications—Configurability, or variability, of multi-tenant applications is seldomly mentioned. This raises two interesting points:

- As discussed in Observation 1 this may hint at a greater awareness of multi-tenancy at the infrastructural or platform level, where configurability might not be so much of an issue.
- There is no apparent need for the configurability of multi-tenant software applications, which might indicate that most applications are actually *multi-user* applications or applications that share resources but that do not offer (advanced) forms of configurability.

When customization is discussed, it is clear that customization should lead to a tailored experience for each tenant and that customization should be done by configuring application metadata. As such, configurability requires no programming. Another important point mentioned is that customizations for one client should not affect other clients.

Observation 4: Multi-tenant database—A number of websites explicitly mention the database as being multi-tenant. In this situation different applications share a single database. When a single multi-tenant application is using the database, some web site authors express concern about data separation, i.e., making sure that tenants do not get access to another tenant's data.

5. Definition

A total of 43 different definitions was extracted from the academic literature with the aim of finding the best definition for use in the multi-tenancy domain, that describes the relevant elements, but also at all levels at which multi-tenancy is possible.

Identification. The 43 definitions were identified by manually searching through papers for terms such as "we define multi-tenancy" or "multi-tenancy is defined as". A common observation from these definitions is that these are typically poorly formulated and only applicable at one level of the software stack or infrastructure. An example: "A multi-tenant cloud system allows multiple users to share a common physical computing infrastructure in a cost-effective way" (Du et al., 2010). This definition is not generic, but refers specifically to a "system". Its strong points are the "common physical computing infrastructure" and its emphasis on "costs", one of the main drivers of multi-tenancy. Another definition is "Multi-tenancy allows a single application to emulate multiple application instances" (Azeez et al., 2010). This definition speaks specifically of an application, thereby excluding for instance hardware resources or databases.

Word frequency analysis—An analysis of frequent occurrences of terms was performed to find the main concepts in multi-tenancy definitions. The results of this analysis can be found in Table 4. Obviously, relevant aspects of multi-tenancy are the fact that

Table 4
Word frequency analysis.

Word	Occurrence
Instance	26
Application	24
SaaS	22
Multiple	21
Infrastructure	20
Single	18
Software	15
Customer	15
Share	13
Database	12

something (single) is being shared among multiple customers, that it takes place on several levels (system, service, application, database, and infrastructure), and that it changes traditional modes of service or software delivery. To clarify, we have conceptualized a system, such that we can reuse it for the definition later in Fig. 2. The dotted boxes are parts of the system that are not influenced by software level multi-tenancy. Efforts exist to apply multi-tenancy at the middle-ware level (Strauch et al., 2013), but we did not explicitly analyse this, for the sake of creating a high level general definition.

Checklist. A checklist containing five criteria was constructed for use in this research, in order to assess the quality of all definitions. The list is based on five principles discussed by Copi and Miller (1972). Furthermore, for each definition we attempted to establish whether it was abstract enough to play a part on all three levels (service, database, and infrastructure). The criteria were formulated as follows:

- A definition must set out the essential attributes of the thing defined.
- Definitions should avoid circularity.
- The definition must not be too wide or too narrow. It must be applicable to everything to which the defined term applies (i.e., not miss anything out), and to nothing else (i.e., not include any things to which the defined term would not truly apply).
- The definition must not be obscure.
- A definition should not be negative where it can be positive.

Several definitions were selected to establish a baseline for the multi-tenancy definition in this paper, based on the criteria mentioned above. First, the definition given by Rimal, Choi, and Lump is "multi-tenancy is when common resources and a single instance of both the object code of an application and the underlying database are used to support multiple customers simultaneously" (Rimal et al., 2009). The definition includes relevant aspects of multi-tenancy, such as "multiple customers" and "common resources" and it speaks of all three levels on which multi-tenancy can play a part (database, service, and hardware

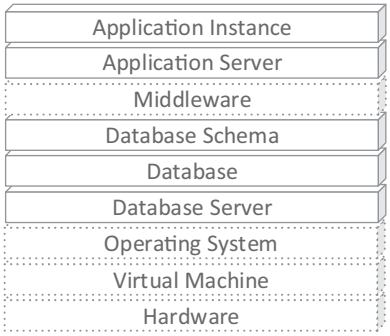

Fig. 2. Software stack. The different system levels where multi-tenancy can be applied to share resources.

Table 5
Difference overview for multi-tenant, multi-user, and multi-instance systems.

Multi	Shared resources	Configurable at runtime
-tenant	Yes	Fully
-user	Yes	Partly
-instance	Possibly	Possibly

resources). However, the definition lacks a goal statement (what is the advantage of multi-tenancy?). Another definition is given by Guo et al.: "In a multi-tenant enabled service environment, user requests from different organizations and companies (tenants) are served concurrently by one or more hosted application instances based on the shared hardware and software infrastructure." (Guo et al., 2007). This definition too addresses only two levels, but adds multiple instances of the software. Finally, an interesting definition is "Multi-tenancy aims to enable a service environment that user requests from different tenants are served concurrently by the least number of hosted service instances running on the shared hardware and software infrastructure" (Li et al., 2008) which focuses on reducing costs by sharing resources. Based on the definitions stated above we define multi-tenancy as follows:

Definition: *Multi-tenancy is a property of a system where multiple customers, so-called tenants, transparently share the system's resources, such as services, applications, databases, or hardware, with the aim of lowering costs, while still being able to exclusively configure the system to the needs of the tenant.*

This definition caters to different needs. To begin with it mentions the most common terms used to identify multi-tenancy (with the sole exception of "instance", but more on that later). Furthermore, it embraces any kind of system and its layers, from a complete service system with multiple instances (like Salesforce.com), to a simple hard drive that is shared among different end-users. Thirdly, it provides the main aim for applying multi-tenancy in a context, being the reduction of costs by sharing resources and achieving scalability. The words "single" and "instance" have been deliberately avoided, such that a qualifier can be used to determine whether we are speaking of single-instance or multiple-instance. The definition prescribes that when someone assigns the property multi-tenant, it is assigned to a system, service, database, or hardware resource, to clarify on what layer the multi-tenancy aspect applies. Although a small detail, it must be noted that multi-tenancy is written with a dash in 75% of the definitions.

There are several clarifications that can be made with the definition at hand. First, the word "transparently" refers to the fact that it is generally unknown to customers and end-users that another customer or end-user is using the same resources, otherwise the definition would be applicable to any web application that is open to multiple users (Google.com, Facebook, etc.).

A question that is frequently asked is what the differences are between multi-tenant, multi-user and multi-instance systems. The answer is that multi-instance systems do not necessarily need shared resources: a new system can be generated or deployed for each new user. Multi-tenant and multi-user systems, however, always share resources on one or more levels of the software stack. Multi-tenant systems share resources and allow only for mass-customization by using variability. Multi-user systems are only partly multi-tenant and offer the same invariable functionality to all customers. Please see Table 5 for an overview of these differences.

6. Research agenda

In order to structure and guide future research in the area of multi-tenancy for both academics and practitioners, this section presents the major future research topics identified in current research on multi-tenancy. The "future work" sections of all final

papers identified in the systematic mapping studies were analyzed to extract potential future research topics. For this search all sections named "future work", "further work", "discussion" and "conclusion" were included. Also, all papers were searched entirely, using the keyword "future". First, all topics mentioned in the relevant sections were listed, after which synonyms and issues that were closely related were merged to overarching research themes. Classification and merging of the topics was performed by two researchers separately, after which the results were compared and discussed. This way, 23 issues were identified, which were categorized in four research themes. The analysis is based on the 48 papers that were collected in the structured mapping study. Every call for future work identified in the papers reflects a potentially strategic theme in the domain of multi-tenancy. Each of the themes below states the number of papers that address the theme and mention a specific call to action to researchers and practitioners.

Quality assurance (6)—Compliance to Service Level Agreements (SLAs), performance, monitoring, all are mentioned in the current body of multi-tenancy literature as important issues to address in future research. Most issues within this topic are similar to important research challenges in the domain of SaaS (Zhang et al., 2010). This can be explained by the fact that multi-tenant software is always hosted in a SaaS environment, causing challenges in this domain to influence the multi-tenancy domain as well.

Call: An investigation into how customization of the multi-tenant application affects quality, e.g., in terms of performance. Can one general SLA be upheld, or should each tenant get a tenant-specific SLA?

Industry validation (4)—Some papers reported on multi-tenant prototypes created, but all were missing a real validation. Because of this, a high number of papers call for industrial application of multi-tenant solutions. Applying prototypes in real industrial settings and performing more multi-tenancy related case studies can greatly enhance the validity of multi-tenancy research and is therefore considered to be a major topic in future research.

Call: With industrial multi-tenant solutions being developed right now, a next step for researchers is to work closely together with industry to validate research ideas on actual multi-tenant software systems.

Balancing & placement (4)—Although all customers in a multi-tenant environment theoretically are served from one instance of a software product, in practice, load balancing is needed between servers. This means identical servers are used to serve one software product in case this can no longer be done using one server. Specific tenants need to be placed on a specific server, but determining the best placement is a difficult task.

Call: There might be opportunities to develop better load balancing algorithms that take into account the historical usage of the application by the different tenants. Specifically, the load balancing can be targeted at looking at the different time zones in which the tenants are operating.

Database (4)—Four papers in the systematic mapping study explicitly mentioned database related issues as an important future research direction. Areas of interest include parallelism, locking, replication and partitioning.

Call: A major point of concern that we noted in the blog posts is data isolation, i.e., making sure that the data of individual tenants is shielded for other tenants. As such, an investigation into how to isolate and partition the data is a logical next step. Additionally, developing tests to make sure that data isolation is working correctly is also an interesting avenue for future work.

Three additional themes were identified, but were not sufficiently highlighted to count towards a valid collection of research themes. Although these themes were not emphasized by a sufficient number of authors, we mention them here briefly, to provide insight into other issues that are relevant. First, two papers mention the development of and research on multi-tenant platforms as an important next step in multi-tenancy research. The *development of a multi-tenant platform (2)* enables other researchers and developers to more easily deploy and test multi-tenant applications. Such a platform (ie. Salesforce (Fisher, 2007)) is likely to stimulate multi-tenancy research and development. The *call* in this context would be the need for an open platform available for multi-tenant applications. Researchers and industry should work together in designing, developing, and maintaining such a platform. Secondly, *security (2)* is a recurring theme in future work (Zhang et al., 2010), where papers specifically focus on the fact that different organizations, each having their own confidential data, are typically deployed on the same server and use the same instance of a software product. This increases the risk of data accidentally being queried by the wrong tenant. This leads to a *call* for increased attention to security in multi-tenant systems than it already does in multi-instance and multi-user systems. Finally, a theme that only occurs once in the literature that we surveyed, but poses a relevant challenge is *variability (1)*. Since multi-tenant software is almost exclusively used in a setting in which multiple different organizations use the same instance of a software product, variability is an important research topic. Variability is the ability of a software product to offer different configurations to organizations hosted on one instance of a software product. The definition of multi-tenancy presented in this paper also mentions 'varying customers', inducing the need for variability (Kabbedijk and Jansen, 2012). The corresponding *call* is that there should be more awareness on the importance of variability in multi-tenant software.

7. Threats to validity

Since conducting a systematic mapping study is a largely manual task, most threats to validity relate to the possibility of researcher bias, and thus to the concern that other researchers might come to different results and conclusions. One remedy we adopted is to follow, where possible, guidelines on conducting systematic mapping studies as suggested by Budgen et al. (2008) and Petersen et al. (2008). The question of whether an article or blog post should be included in the mapping study is sometimes debatable. Following the advice of Kitchenham (2004), we enforced this criterion by utilizing predefined selection criteria that clearly define the scope (also see Section 2).

A potential threat to the validity of the interpretation of the results is researcher bias in the selection and filtering of the articles and blog posts. Our countermeasures were (1) the systematic logging of all data related to the screening and filtering steps in a database accessible by all authors of the paper and (2) randomly selecting 10% of all papers after each selection or filtering step to determine the inter-rater agreement for that subset of papers. If a paper is rated differently by another author, the discrepancy was discussed. Finally, this research assessed results published up to 2012, so the landscape of multi-tenancy could have evolved slightly in the mean time. This is identified as a threat to validity.

8. Conclusion

A total of 761 research papers and 371 industrial blogs on multi-tenancy have been analyzed in order to get a complete overview of the multi-tenancy domain. The results show that most papers propose a solution related to multi-tenancy, but almost no papers report on industrial experiences while implementing multi-tenancy, providing some insight into the maturity of the domain. The blog analysis shows multi-tenancy is a popular topic

and most blogs are written by individuals instead of corporations. Based on the research results a comprehensive definition for multi-tenancy is proposed (**SubRQ1**), positioning multi-tenancy as an architectural principle of a system where multiple varying customers and their end-users transparently share the systems services, applications, databases, or hardware resources, with the aim of lowering costs. We call for this definition to be used in future research on multi-tenancy to further structure results and communication. No clear difference on the interpretation of multi-tenancy *between* academia and industry was observed, but we did see a significant difference *among* academia and industry (**SubRQ2**). For future research we listed 4 themes (**SubRQ3**), meant for the guidance of future research and providing a roadmap within the domain of multi-tenancy. The main research question (**RQ**) is answered by the complete drawing of the current multi-tenancy domain from both the academic and industrial perspective, together with the directions for steering the domain from this point on.

Appendix A. Systematic mapping study paper list

Here a complete list of all final papers identified within the systematic mapping study is presented in alphabetical order of first author.

Arya, P.K., Venkatesakumar, V., Palaniswami, S., 2010. Configurability in saas for an electronic contract management application. In: Proceedings of the 12th International Conference on Networking, VLSI and signal processing (ICNVS'10), University of Cambridge, UK, pp. 210-216.

Azeez, A., Perera, S., Gamage, D., Linton, R., Siriwardana, P., Leelaratne, D., Weerawarana, S., Fremantle, P., 2010. Multi-tenant soa middleware for cloud computing. In: Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on. IEEE, pp. 458-465.

Bakshi, K., 2011. Considerations for cloud data centers: Framework, architecture and adoption. In: Aerospace Conference, 2011 IEEE. IEEE, pp. 1-7.

Bezemer, C.-P., Zaidman, A., 2010. 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). ACM, pp. 88–92.

Bezemer, C.-P., Zaidman, A., Platzbeecker, B., Hurkmans, T., t Hart, A., 2010. Enabling multi-tenancy: An industrial experience report. In: Software Maintenance (ICSM), 2010 IEEE International Conference on. IEEE, pp. 1–8.

Cai, H., Wang, N., Zhou, M.J., 2010. A transparent approach of enabling SaaS multi-tenancy in the cloud. In: Services (SERVICES-1), 2010 6th World Congress on. IEEE, pp. 40–47.

Cai, H., Zhang, K., Zhou, M.J., Gong, W., Cai, J.J., Mao, X.S., 2009. An end-to-end methodology and toolkit for fine granularity SaaS-ization. In: IEEE International Conference on Cloud Computing. IEEE, pp. 101–108.

Domingo, E.J., Ni no, J.T., Lemos, A.L., Lemos, M.L., Palacios, R.C., Berbís, J.M.G., 2010. Cloudio: A cloud computing-oriented multi-tenant architecture for business information systems. In: Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on. IEEE, pp. 532–533.

Du, J., Gu, X., Reeves, D.S., 2010. Highly available component sharing in large-scale multi-tenant cloud systems. In: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing. ACM, pp. 85–94.

Fehling, C., Leymann, F., Mietzner, R., 2010. A framework for optimized distribution of tenants in cloud applications. In: Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on. IEEE, pp. 252–259.

Foping, F.S., Dokas, I.M., Feehan, J., Imran, S., 2009. A new hybrid schema-sharing technique for multitenant applications. In: ICDIM 2009. Fourth International Conference on Digital Information Management. IEEE, pp. 1–6.

Grund, M., Schapranow, M., Krueger, J., Schaffner, J., Bog, A., 2008. Shared table access pattern analysis for multi-tenant applications. In: IEEE Symposium on Advanced Management of Information for Globalized Enterprises. IEEE, pp. 1–5.

Guo, C.J., Sun, W., Huang, Y., Wang, Z.H., Gao, B., 2007b. A framework for native multi-tenancy application development and management. In: The 9th IEEE International Conference on E-Commerce Technology. IEEE, pp. 551–558.

Guo, C.-J., Sun, W., Jiang, Z.-B., Huang, Y., Gao, B., Wang, Z.-H., 2011. Study of software as a service support platform for small and medium businesses. In: New Frontiers in Information and Software as Services. Springer, pp. 1–30.

Jacobs, D., Aulbach, S., et al., 2007. Ruminations on multi-tenant databases. In: BTW. Vol.103, pp. 514–521.

Jiang, X., Zhang, Y., Liu, S., 2010. A well-designed saas application platform based on model-driven approach. In: Grid and Cooperative Computing (GCC), 2010 9th International Conference on. IEEE, pp. 276–281.

Kang, S., Kang, S., Hur, S., 2011. A design of the conceptual architecture for a multitenant saas application platform. In: Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on. IEEE, pp. 462–467.

Kangarlou, A., Xu, D., Kozat, U.C., Padala, P., Lantz, B., Igarashi, K., 2011. In-network live snapshot service for recovering virtual infrastructures. IEEE Network 25 (4), 12–19.

Kong, L., Li, Q., Zheng, X., 2010. A novel model supporting customization sharing in SAAS applications. In: International Conference on Multimedia Information Networking and Security (MINES), 2010. IEEE, pp. 225–229.

Kwok, T., Nguyen, T., Lam, L., 2008. A software as a service with multi-tenancy support for an electronic contract management application. In: IEEE International Conference on Services Computing, SCC'08, 2008. Vol.2. IEEE, pp. 179–186.

Lee, J., Hur, S.J., 2011. Level 2 SaaS platform and platform management framework. In: 13th International Conference on Advanced Communication Technology (ICACT), 2011. IEEE, pp. 1177–1180.

Li, X.H., Liu, T.C., Li, Y., Chen, Y., 2008. Spin: Service performance isolation infrastructure in multi-tenancy environment. In: Service-Oriented Computing-ICSOC 2008. Springer, pp. 649–663.

Li, X.-Y., Shi, Y., Guo, Y., Ma, W., 2010. Multi-tenancy based access control in cloud. In: International Conference on Computational Intelligence and Software Engineering (CiSE), 2010. IEEE, pp. 1–4.

Lin, H., Sun, K., Zhao, S., Han, Y., 2009. Feedback-control-based performance regulation for multi-tenant applications. In: 15th International Conference on Parallel and Distributed Systems (ICPADS), 2009. IEEE, pp. 134–141.

Mietzner, R., Karastoyanova, D., Leymann, F., 2009. Business grid: Combining web services and the grid. In: Transactions on Petri Nets and Other Models of Concurrency II. Springer, pp. 136–151.

Mietzner, R., Leymann, F., Papazoglou, M.P., 2008. Defining composite configurable SAAS application packages using SCA, variability descriptors and multi-tenancy patterns. In: Third International Conference on Internet and Web Applications and Services, 2008. ICIW'08. IEEE, pp. 156–161.

Mietzner, R., Unger, T., Titze, R., Leymann, F., 2009. Combining different multi-tenancy patterns in service-oriented applications. In: IEEE International Enterprise Distributed Object Computing Conference, 2009. EDOC'09. IEEE, pp. 131–140.

Ranchal, R., Lilien, L., Bhargava, B., Kim, A., Othmane, L.B., Kang, M., 2010. An approach for preserving privacy and protecting

personally identifiable information in cloud computing. Unknown Journal.

Rimal, B.P., Choi, E., Lumb, I., 2010. A taxonomy, survey, and issues of cloud computing ecosystems. In: Cloud Computing. Springer, pp. 21–46.

Schaffner, J., Eckart, B., Schwarz, C., Brunnert, J., Jacobs, D., Zeier, A., 2011. Towards analytics-as-a-service using an in-memory column database. In: New Frontiers in Information and Software as Services. Springer, pp. 257–282.

Sénica, N., Teixeira, C., Pinto, J.S., 2011. Cloud computing: A platform of services for services. In: ENTERprise Information Systems. Springer, 91–100.

Shi, Y., Luan, S., Li, Q., Wang, H., 2009a. A flexible business process customization framework for saas. In: WASE International Conference on Information Engineering, 2009. ICIE'09. Vol.2. IEEE, pp. 350–353.

Shi, Y., Luan, S., Li, Q., Wang, H., 2009. A multi-tenant oriented business process customization system. In: International Conference on New Trends in Information and Service Science, 2009. NISS'09. IEEE, pp. 319–324.

Shwartz, L., Diao, Y., Grabarnik, G.Y., 2009. Multi-tenant solution for it service management: A quantitative study of benefits. In: IFIP/IEEE International Symposium on Integrated Network Management, 2009. IM'09. IEEE, pp. 721–731.

Siddhisena, B., Warusawithana, L., Mendis, M., 2011. Next generation multi-tenant virtualization cloud computing platform. In: 13th International Conference on Advanced Communication Technology (ICACT), 2011. IEEE, pp. 405–410.

Tang, K., Jiang, Z.B., Sun, W., Zhang, X., Dong, W.S., 2010. Research on tenant placement based on business relations. In: IEEE 7th International Conference on e-Business Engineering (ICEBE), 2010. IEEE, pp. 479–483.

Tsai, C.-H., Ruan, Y., Sahu, S., Shaikh, A., Shin, K.G., 2007. Virtualization-based techniques for enabling multi-tenant management tools. In: Managing Virtualization of Networks and Services. Springer, pp. 171–182.

Tsai, W.-T., Shao, Q., Elston, J., 2010. Prioritizing service requests on cloud with multi-tenancy. In: e-Business Engineering (ICEBE), 2010 IEEE 7th International Conference on. IEEE, pp. 117–124.

Tsai, W.-T., Sun, X., Shao, Q., Qi, G., 2010. Two-tier multi-tenancy scaling and load balancing. In: IEEE 7th International Conference on e-Business Engineering (ICEBE), 2010. IEEE, pp. 484–489.

Wang, D., Zhang, Y., Zhang, B., Liu, Y., 2009. Research and implementation of a new saas service execution mechanism with multi-tenancy support. In: 1st International Conference on Information Science and Engineering (ICISE), 2009. IEEE, pp. 336–339.

Wang, X.F., Dong, P.J., 2009. The multi-tenant data architecture design for the collaboration service system of textile and apparel supply chain. In: WiCom'09. 5th International Conference on Wireless Communications, Networking and Mobile Computing, 2009. IEEE, pp. 1–4.

Wang, Z.H., Guo, C.J., Gao, B., Sun, W., Zhang, Z., An, W.H., 2008. A study and performance evaluation of the multi-tenant data tier design patterns for service oriented computing. In: IEEE International Conference on e-Business Engineering, 2008. ICEBE'08. IEEE, pp. 94–101.

Weissman, C.D., Bobrowski, S., 2009. The design of the force.com multitenant internet application development platform. In: SIGMOD Conference, pp. 889–896.

Wu, M., 2011. Cloud computing: Hype or vision. In: Applied Informatics and Communication. Springer, pp. 346–353.

Xu, X., 2012. From cloud computing to cloud manufacturing. Robot. Comput.Integr. Manuf. 28 (1), 75–86.

Zhang, K., Li, Q., Shi, Y., 2011. Data privacy preservation during schema evolution for multi-tenancy applications in cloud computing. In: Web Information Systems and Mining. Springer, pp. 376–383.

Zhang, S.W., Wang, X.P., 2011. Configuration of multi-tenant applications. Adv. Mater. Res. 219, 1182–1185.

Zhang, Y., Wang, Z., Gao, B., Guo, C., Sun, W., Li, X., 2010. An effective heuristic for on-line tenant placement problem in SAAS. In: IEEE International Conference on Web Services (ICWS), 2010. IEEE, pp. 425–432.

References

Anjum, M., Budgen, D.,2012. A mapping study of the definitions used for service oriented architecture. In: Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering (EASE). IET, pp. 57–61.

Azeez, A., Perera, S., Gamage, D., Linton, R., Siriwardana, P., Leelaratne, D., Weerawarana, S., Fremantle, P.,2010. Multi-tenant SOA middleware for cloud computing. In: Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD). IEEE, pp. 458–465.

Bennett, K., Layzell, P., Budgen, D., Brereton, P., Macaulay, L., Munro, M.,2000. Service-based software: the future for flexible software. In: Proceedings of the Seventh Asia-Pacific Software Engineering Conference (APSEC 2000). IEEE, pp. 214–221.

Bezemer, C., Zaidman, A.,2010. 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). ACM, pp. 88–92.

Bezemer, C., Zaidman, A., Platzbeecker, B., Hurkmans, T., Hart, A., 2010. Enabling multi-tenancy: an industrial experience report. In: 26th IEEE International Conference on Software Maintenance (ICSM).

Budgen, D., Turner, M., Brereton, P., Kitchenham, B., 2008. Using mapping studies in software engineering. In: Proceedings of the 20th Annual Meeting of the Psychology of Programming Interest Group (PPIG), pp. 195–204.

Chong, F., Carraro, G., 2006. Architecture Strategies for Catching the Long Tail. MSDN Library, Microsoft Corporation, pp. 9–10.

Copi, I.M., Miller, R.W., 1972. Introduction to Logic: Study Guide. Macmillan, New York.

Du, J., Gu, X., Reeves, D.S., 2010. Highly available component sharing in large-scale multi-tenant cloud systems. In: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, pp. 85–94.

Eisenhardt, K.M., 1989. Building theories from case study research. Acad. Manage. Rev. 14 (4), 532–550.

Fisher, S.,2007. The architecture of the apex platform, salesforce. com's platform for building on-demand applications. In: 29th International Conference on Software Engineering-Companion (ICSE2007). IEEE, 3–3.

Forbes, 2012, November. Cloud Computing and Enterprise Software Forecast Update. http://goo.gl/K3LnNS

Guo, C., Sun, W., Huang, Y., Wang, Z., Gao, B., 2007. A framework for native multi-tenancy application development and management. In: Proceedings of the 9th IEEE International Conference on E-Commerce Technology, IEEE Computer Society, pp. 551–558.

IBM, 2011, July. Best Practices for Cloud Computing Multi-Tenancy. http://goo.gl/Txzbu4

Kabbedijk, J., Jansen, S.,2012. The role of variability patterns in multi-tenant business software. In: Proceedings of the WICSA/ECSA 2012 Companion Volume. ACM, pp. 143–146.

Kitchenham, B., 2004. Procedures for Performing Systematic Reviews. Tech. rep., Keele. Keele University, UK.

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S., 2009. Systematic literature reviews in software engineering-a systematic literature review. Inf. Softw. Technol. 51 (1), 7–15.

Kitchenham, B.A., Budgen, D., Brereton, P., 2010. The value of mapping studies-a participant-observer case study. In: Proceedings of Evaluation and Assessment of Software Engineering (EASE2010), pp. 195–204.

Kwok, T., Nguyen, T., Lam, L.,2008. A software as a service with multi-tenancy support for an electronic contract management application. In: Proceedings of the IEEE International Conference on Services Computing (SCC'08). Vol. 2. IEEE, pp. 179–186.

Li, X., Liu, T., Li, Y., Chen, Y., 2008. Spin: service performance isolation infrastructure in multi-tenancy environment. In: Proceedings of the International Conference on Service-Oriented Computing (ICSOC 2008), pp. 649–663.

Microsoft, 2012, June. Principles of Sharing in a Shared First World: Multi-Tenancy, urlhttp://goo.gl/x3yybzhttp://goo.gl/x3yybz.

Natis, Y.V., 2008. Reference architecture for multitenancy: enterprise computing in the cloud. Gartner 3, 4–8.

Oracle, 2009, October. Multi-Tenancy (Non Virtualized). http://goo.gl/Ogn8xV

Osipov, C., Goldszmidt, G., Taylor, M., Poddar, I., 2009. Develop and Deploy Multi-Tenant Web-Delivered Solutions Using IBM Middleware: Part 2: Approaches for Enabling Multi-Tenancy. Tech. rep., IBM Corporation.

Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., 2008. Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, pp. 71–80.

Petticrew, M., Roberts, H., 2009. Systematic Reviews in the Social Sciences: A Practical Guide. Blackwell Publishing, Oxford.

Rimal, B., Choi, E., Lumb, I.,2009. A taxonomy and survey of cloud computing systems. In: Proceedings of the Fifth International Joint Conference on INC, IMS and IDC (NCM'09). IEEE, pp. 44–51.
Smith, M.A., Kumar, R.L., 2004. A theory of application service provider (asp) use from a client perspective. Inf. Manage. 41 (8), 977–1002.
Strauch, S., Andrikopoulos, V., Gómez Sáez, S., Leymann, F., 2013. Esbmt: A multi-tenant aware enterprise service bus. Int. J. Next-Gen. Comput. 4 (3), 230–249.
Wieringa, R., Heerkens, H., Regnell, B.,2009. How to write and read a scientific evaluation paper. In: 17th IEEE International Requirements Engineering Conference, 2009. RE'09. IEEE, pp. 361–364.
Wilkes, M.V., 1975. Time Sharing Computer Systems. American Elsevier, New York.
Zhang, Q., Cheng, L., Boutaba, R., 2010. Cloud computing: state-of-the-art and research challenges. J. Internet Serv. Appl. 1 (1), 7–18.

Jaap Kabbedijk is a PhD. candidate at the Department of Information and Computing Sciences at Utrecht University. His research currently focuses on runtime variability in multi-tenant Software-as-a-Service environments, but he has also published several scientific papers in the area of Software Ecosystems and Requirements Engineering.

Cor-Paul Bezemer obtained his Ph.D. degree from the Delft University of Technology (The Netherlands) in April 2014 after succesfully defending his thesis titled 'Performance Optimization of Multi-Tenant Software Systems'. He is currently a postdoc researcher in the parallel and distributed systems group at the Delft University of Technology, taking part in the Tribler project. His main research interests are multi-tenancy, performance analysis and improvement, and interaction analysis.

Slinger Jansen is an assistant professor at the Department of Information and Computer Science at Utrecht University. He is one of the leading researchers in the domain of software ecosystems and co-founders of the International Conference on Software Business and International Workshop on Software Ecosystems. He is lead editor of the book "Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry" and of several others. Besides his academic endeavors he actively supports new enterprises and sits on the boards of advisors of several start-ups, one of which is ThinkEcosystems.com.

Andy Zaidman is an associate professor at the Delft University of Technology, The Netherlands. He obtained his M.Sc. (2002) and Ph.D. degree (2006) in Computer Science from the University of Antwerp, Belgium. His main research interests are software evolution, program comprehension, mining software repositories and software testing. He was the general chair of the 15th Working Conference on Reverse Engineering (WCRE 2008) held in Antwerp, Belgium, program co-chair of WCRE 2009 held in Lille, France and program co-chair of VISSOFT 2014 held in Victoria, BC, Canada. In 2013 Andy Zaidman was the laureate of a NWO Vidi career grant.