Multi-Tenant Architectures in the Cloud: A Systematic Mapping Study

Gözde Karataş, Ferit Can, Gamze Doğan, Cemile Konca, Akhan Akbulut

Abstract—Unlike traditional web applications, cloud-based applications can provide services for large number of tenants using the same hardware and software by implementing multi-tenant architectures. 407 papers have been examined by using systematic mapping method to evaluate the publications related to this architecture, which have been used increasingly in the Software-as-a-Service (SaaS) model. The goal of the study is to determine which storage strategies were used most, which criteria were taken into account in selecting the preferred storage strategy, and the most searched topics under multi-tenant architecture model. Primary researches which are conforming to specified review rules have been obtained from electronic databases (IEEE, ACM, Springer, ScienceDirect, Wiley, Scopus) and classified by research topic and content.

Index Terms—cloud computing,multi-tenancy, multi-tenant architecture, systematic mapping study.

1 Introduction

Cloud Computing, one of the most popular and widely used technologies today, is a service model that enables users to access information system resources efficiently via the network/internet. Users; cover their needs like data storage, processor power and software from outside sources with the concepts of pay-per-use. This model offers a number of advantages for business such as lowering hard- ware, software and maintenance costs, and also high performance and flexibility [1], [2].

In order to take full advantage of cloud computing, services provided to the client should be designed in accordance with the ”multi-tenant” architectural structure which is aimed maximizing resource sharing among users. Service providers reduce service costs tenant per customer by maintaining resource utilization at the highest level [3], [4]. We can describe multi-tenant as an architectural structure that allows all information system resources to be shared by multiple users and sub-users at the same time.

In the second part of the report, information about Multi-tenant architecture and Systematic Matching Study is given. In the third chapter, the details of the research method are explained In the fourth section, the presentation of the obtained results is presented with the aid of tables and graphics. In the last part, the given systematic mapping study was evaluated and the studies to be carried out in the next period are pointed [5], [6].

2 Preliminaries

This section describes the multi-tenant architecture.

2.1 Multi-tenant Architectures

In a multi-tenant architecture, a single software instance is set up on the server running on the service provider infrastructure, and this software receives service by providing access to more than one tenant at the same time. It is used in the realization of the Software as a Service(Saas), which is rapidly spreading especially in small and medium sized enterprises. Tenant refers to persons, institutions or organizations that hire the multi-tenant Saas model [3], [4], [5], [6], [7].

2.1.1 Data Storage Strategy

There are three data Storage Strategies defined between isolated and shared features. These are shown in Figure-1 [8], [9], [10], [11], [12], [13], [14].

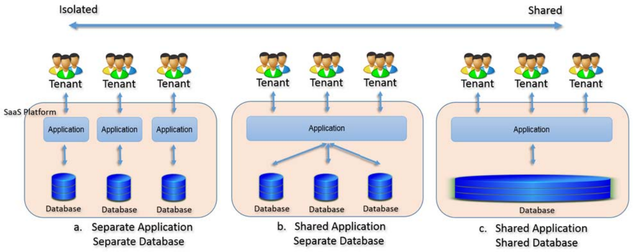


Fig. 1: Data Storage Strategy

Separate Application, Separate Database: In this model every user has its own software and database. All tenants are completely isolated from each other. Maintenance and update costs take too much time and system resources are not use efficiently in this model. *Shared Application, Separate Database:* All tenants are using the same and only one software. Also, all users have their own physically separated databases. Special methods are used so that software can be individually customized for each tenant according to the tenant's wishes. *Shared Application, Shared Database:* Tenants use a common software. Also, this model is divided into two model. These are *Shared Database-Separate Schema* and *Shared Database-Shared Schema*. These are shown in Figure-2.

Shared Database, Separate Schema: In this approach, each tenant has its own separate database tables on the common database. Example is shown in Figure-3. As separate

978-1-5386-1880-6/17/$31.00 ©2017 IEEE

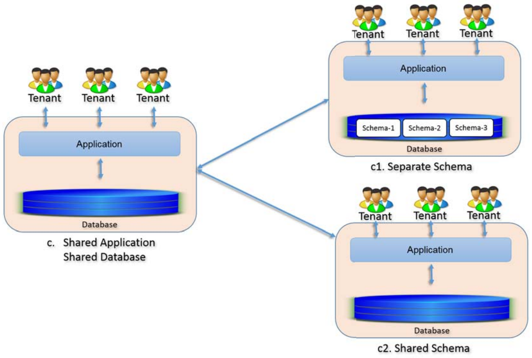


Fig. 2: Separate Schema*Shared Schema

schemes are created for each tenant, the implementation is partly easier and can be easily shaped to suit the needs of a user. The disadvantage is that a large number of tables must be kept.

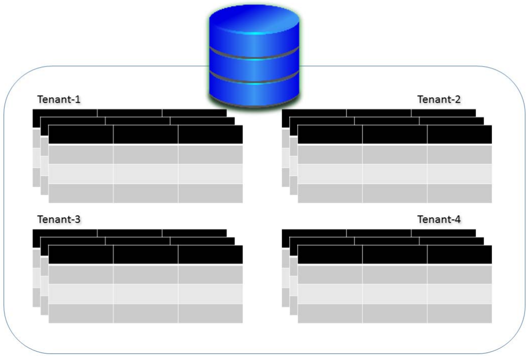
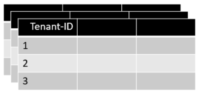


Fig. 3: Separate Schema

Shared Database, Shared Schema: In this approach, there is a common database and common schema for all tenants.



Tenant-ID		
1		
2		
3		

Fig. 4: Shared Schema

This model provides the least use and maintenance cost in terms of the most efficient use of hardware. The disadvantage is that all tenants are in the same schema.

2.1.2 Advantages and Disadvantages of a Multi-tenant Architecture

Efficient use of hardware resources and reduces cost with ease of maintenance and updating are the advantages of the multi-tenant architecture. The disadvantages that the problem caused by a tenant is affected by all other tenants because the whole system works on the same hardware and any mistake made during coding can cause major problems in the future system management.

2.2 Systematic Mapping Study

At the beginning of the 90's, researchers in medicine field have started to use Evidence Based Software Engineering [15]. With the success of this approach, many researchers have reached to the possibility of applying this approach in a short time in different areas. Evidence-Based Software Engineering collects empirical evidence from its primary works and conducts research with the Systematic Literature Review and Systematic Mapping Study. Systematic Mapping Study is a convenient way to match a given topic when there is a small amount of evidence or topic is irrelevant [16].

In this study articles, conference proceedings, and book chapters published in last 17 years have been examined in order to carry out a Systematic Mapping Study on multi-tenant architects in cloud computing.

3 Research Method

Systematic Matching Studies include next steps; identification of research questions, creation of a search string with specified keywords, searching the sources and finding the publications in order to find the relevant articles, determination of inclusion and exclusion criteria, making publication eliminations based on inclusion and exclusion criteria, search for content and answers to research questions, making inferences and sharing study results [17].

3.1 Research Questions

The research questions should reflect the fundamental purpose of the mapping study. They should include all aspects of defined goals. The research questions identified in this study are;
RQ1: What types of publications have been made about multi-tenancy?
RQ2: What kinds of studies have been produced including publication years and sources?
RQ3: Which storage strategy is preferred for multiple tenancy?
RQ4: Which criteria have been taken into account in determining the preferred storage strategy?
RQ5: Which is the most searched title for multi-tenancy systems? What is the purpose of the research?
RQ6: What are the known benefits, damages, and limitations of multi-tenancy systems?

3.2 Search Method

In this study, firstly the keywords to be searched are determined then the search was made with the search string created with these keywords in different data sources. It has been taken into consideration that the search keywords are chosen to provide all possible conclusions about the topic. The key search string used in this study is as follows.

(Cloud AND (SaaS OR Software as a Service) AND (multi-tenant OR Multi-tenancy OR tenant OR tenancy OR tenants)) OR multi-tenant OR Multi-tenancy OR tenant OR tenancy OR tenants

After the determination of search key string, in the "IEEE, ACM, Wiley, ScienceDirect, Scopus, and SpringerLink" data sources, this search string was used to search for articles covering the years 2000-2017. Within these databases, search query have been run with the help of the advanced search section. As a result of these studies, 407 studies were collected. The Exclusion-inclusion criteria in Table 1 are considered in order to be able to extract the articles related to the selected topic.

TABLE 1: Inclusion-Exclusion Criteria

Exclusion Criteria	Inclusion Criteria
EC1: Study is not related to multi- tenancy EC2: Duplicate work EC3: The found publication is related but the paper length is too short EC4: Non-English studies EC5: Publications with only summary information EC6: Short notes, panels, posters and session summaries EC7: Secondary studies	IC1: Primary studies IC2: Publications that are studied together with Software as a Sevice(SaaS) and multi-tenancy

3.3 Data Extraction

In this phase, the number of studies received from each data source via search sequences is IEEE 11451, ACM 30632, ScienceDirect 393, Wiley 71, SpringerLink 355 and Scopus 6199 were found. Subsequently, these studies were examined and eliminated according to inclusion-exclusion criteria. Firstly, according to the summary and entry parts, the items were eliminated, then the whole text was read out among the remaining ones and the number of the final articles was reached.On the other hand,if duplicated related study found,only one of them is selected and the others were eliminated. Additions and subtractions were made by asking other colleagues for undecided studies. After the items were eliminated, the works were reduced to 407 related articles and the research questions were applied to these works.

4 Results

Distribution according to types of publications examined in Figure-5. Most publications are made as an academic article. It has been seen that the most publication after the type of academic article is in the type of Research Report.

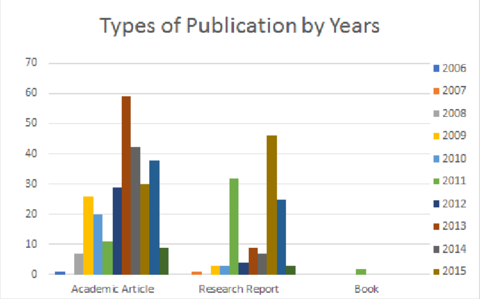


Fig. 5: Types of publications by years

Figure-6 shows the distributions of the publications analyzed by years. Accordingly, there is a regular increase in the number of publications that have been examined since 2008. It is seen that most publications about multi-tenant architects were prepared in 2016.

Looking at the types of publications examined in Figure-7, it is seen that publications in the type of Ideological Suggestion are 58% followed by publications in 31% Experiment and 11% report type.

The preferred storage strategies are examined in the publications examined in Figure-8. Within all storage strategies, Shared software-Shared database storage strategy is the most preferred strategy with 65%. In particular, the shared scheme strategy includes a 35% share within all strategies.

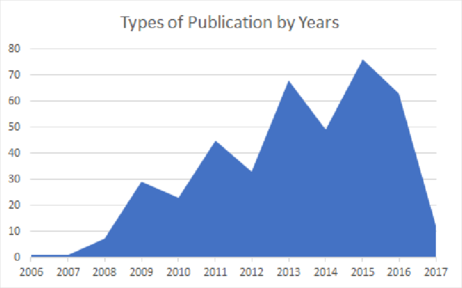


Fig. 6: Distribution of publications by years

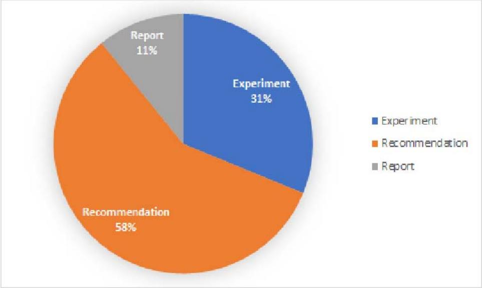


Fig. 7: Types of the publications in reviewed studies

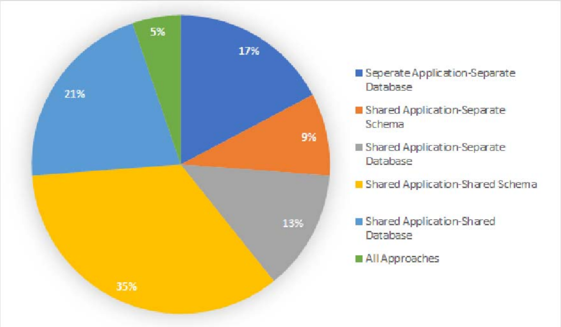


Fig. 8: Preferred storage strategy

When determining the preferred storage strategies, economic reasons come first with 36%, considering the determined criteria. 29% of the publications consider the characteristics of the tenant (number, database size requirement, the number of users per tenant) and 20% publications has safety considerations. The related distribution is shown in Figure-9.

The most research for multi-tenant systems is to provide security issues. Then the programs come to be able to provide customization according to the needs and wishes of the tenants and researches on welding insulation issues.

Multi-tenant systems have many advantages but also have disadvantages. In the examined articles, different solutions to these disadvantages have been produced. Considering the reviewed articles, Multi-Tenant systems have advantages in terms of cost effectiveness, ease of maintenance, flexibility, scalability, resource utilization and also they have disadvantages of security, data confidentiality, data integrity, consistency, resource management, configu-

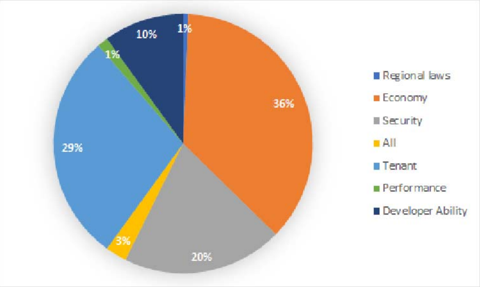


Fig. 9: Criteria considered in determining the preferred storage strategy

ration, access control.

5 Conclusion

Six electronic databases were used in the study, 49101 publications were found, but 407 were used [18]. Suggestions at the end of the research are as follows; Report-type work on multi-tenant systems should be developed, performance studies should be carried out on the storage strategies of the systems and different types of study areas which are suitable for implementing multi-tenancy should be determined and various infrastructures should be prepared.

Finally, in our examinations, we concluded that multi-tenancy is now being applied to a limited number of areas, and that this situation needs to be improved, and that some infrastructure software must be prepared to facilitate implementation. This work is a starting point. In the future, the systematic literature review will be used to try to reach the experimental evidence and try to find out the problems in the application with the questionnaire which will be done in the last stage.

References

[1] Z. Pervez, S. Lee, Y. K. Lee, 2010. ”Multi-tenant, secure, load disseminated SaaS architecture.” In Adv. Com. Tech., Int. Conf. on (Vol. 1, pp. 214-219). IEEE.

[2] W. Su, Q. Liu, C. Lin, S. Shen, 2015. ”Modeling and Analysis of Availability in Multi-Tenant SaaS.”, In Comp. Com. and Net., Int. Conf. on (pp. 1-8). IEEE.

[3] M. Hahn, S. Gomez, V. Andrikopoulos, D. Karastoyanova, F. Leymann, ”SCEMT: A Multi-tenant Service Composition Engine”, in Int. Conf. on Service-Oriented Computing and App., 2014, pp. 8996.

[4] G. Liu, 2010. ”Research on independent SaaS platform.”, In Information Management and Engineering, Int. Conf. on, pp. 110-113. IEEE.

[5] M. Pathirage, S. Perera, I. Kumara, S. Weerawarana, 2011. ”A multi-tenant architecture for business process executions.” In Web services, int. conf. on, pp. 121-128. IEEE.

[6] J. Kabbedijk, C. P. Bezemer, S. Jansen, A. Zaidman, 2015. ”Defining multi-tenancy: A systematic mapping study on the academic and the industrial perspective.” Journal of Systems and Software, 100, 139-148.

[7] K. F. Lee, ”Automatic Speech Recognition: The Development of the SPHINX SYSTEM”, Kluwer Academic Publishers, Boston, 1989.

[8] H. Chen, 2016, ”Architecture strategies and data models of Software as a Service: A review.” In Inf. and Cybernetics for Comp. Social Sys., Int. Conf., pp. 382-385. IEEE.

[9] F. Chong, G. Carraro, R. Wolter, 2006. ”Multi-tenant data architecture.”, MSDN Library, Microsoft Corporation, 14-30.

[10] H. Yaish, M. Goyal, 2013, ”A multi-tenant database architecture design for software applications.” In Comp. Sci. and Engi. Int. Conf., pp. 933-940. IEEE.

[11] W. T Tsai, P. Zhong, 2014, ”Multi-tenancy and sub-tenancy architecture in software-as-a-service (SaaS).” In S.O.S. Engineering Int. Symp., pp. 128-139. IEEE.

[12] W. Su, C. Lin, K. Meng, Q. Liu, 2014, ”Modeling and Analysis of Availability for SaaS Multi-tenant Architecture.” In Service Oriented Sys. Engi. Int.Sym. on, pp. 365-369. IEEE.

[13] C. P .Bezemer, A. Zaidman, B. Platzbeecker, T. Hurkmans, A. T. Hart, 2010. ”Enabling multi-tenancy: An industrial experience report.” In Software Maintenance, pp. 1-8. IEEE.

[14] C. J. Guo, W. Sun, Y. Huang, Z. H. Wang, Z. H., B. Gao, 2007, ”A framework for native multi-tenancy application development and management.” In e-commerce Tech. and the 4th IEEE Int. Conf. on Enterprise Comp., pp. 551-558. IEEE.

[15] B. A. Kitchenham, T. Dyba, M. Jorgensen, ”Evidence-Based Software Engineering”, Proc. of the 26th International Conference on Software Engineering, pp. 273-281.

[16] V. H. S. Pinto, H. J. Luz, R. R. Oliveira, P. S. Souza, S. R. Souza, ”A Systematic Mapping Study on the Multi-tenant Architecture of SaaS Systems, 2016.

[17] A. . Aytekin, E. Tzn, Y. Macit, B. Tekinerdoan, Uygulama Yaam Dngs Ynetimi - Sistematik Eleme almas

[18] http://www.kisa.link/DDa

[19] H. Kopka and P. W. Daly, *A Guide to $\LaTeX$*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

Gözde Karataş received her undergraduate degree from Mathematics and Computer Science Department of Istanbul Kültür University in 2009 and her graduate degree from Computer Engineering Department of Istanbul Kültür University in 2013. In 2015 she completed her master thesis on NoSql Database Testing in Istanbul Kültür University. During her master studies, she worked on distributed databases. Also she has been working at the Department of Mathematics and Computer Science in Istanbul Kültür University as Research Assistant.

Ferit Can received his undergraduate degree from System Engineering of Turkish Military Academy in 2003. He is currently graduate student at the National Defense University. The courses he took are Distributed Systems, Basic Protocols in Computer Networks, Wireless Sensor Networks, Advanced Topics in Software Engineering and Object Oriented Programming Languages and Systems.

Gamze Doğan received her undergraduate degree from Computer Engineering department of Kadir Has University in 2010.She also received her undergraduate degree from Industrial Engineering department as double major. In 2015, she completed her undergraduate thesis on collaborative filtering. Also she has been working at a private bank as Information Security Specialist.

Cemile Konca received her undergraduate degree from Computer Engineering department of Karadeniz Technical University in 2014. Also she has been working at a ERP specialist at a private company. Her research interests lie in the ares of cyber security and computer security.

Akhan Akbulut received his undergraduate and graduate degrees from Computer Engineering Department of Istanbul Kültür University in 2001 and 2008, respectively. In 2013 he completed his Ph.D. thesis on Extending Wireless Sensor Networks to Internet using Cloud Computing in Istanbul University. During his master studies, he worked on medium and large scale e-government, CRM and ERP projects. He has been working at the Department of Computer Engineering in Istanbul Kültür University as Assistant Professor. His research areas are Software Engineering and Cloud Computing.