I.J. Information Technology and Computer Science, **2016, 3, 1-9**
Published Online March 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2016.03.01

A Framework for Multi-Tenant Database Adoption based on the Influencing Factors

Olumuyiwa Matthew
Research Student, University of Wolverhampton, UK.
E-mail: o.o.matthew@wlv.ac.uk

Kevan Buckley and Mary Garvey
School of Mathematics & Computer Science, University of Wolverhampton, UK.
E-mail: K.A.Buckley@wlv.ac.uk; M.Garvey@wlv.ac.uk

***Abstract*—**Multi-tenant databases (MTD) are aspect of computing that has become one of the revolutionary technologies of recent years. This technology has helped to discard the large-scale investments in hardware and software resources, in upgrading them regularly and also in expensive licences of application software used on in-house hosted database systems. A MTD is a way of deploying a Database as a Service (DaaS) for the convenience and benefits of tenants. This concept is good for higher scalability and flexibility but it involves a lot of technicalities. The adoption of a MTD is based on some salient factors which can be grouped into four categories for easy understanding. A survey is presented in this research that involves a focus group of thirty respondents. The result shows the degree of impact each factor has on the decision to adopt a MTD. This paper also considers these factors and develops a framework that will help prospective tenants to take an informed decision about the adoption of the concept.

Index Terms—Multi-tenant, Database as a Service, Scalability, Flexibility, Factor, Framework, Adoption.

I. Introduction

Deploying Database as a Service (DaaS) is gaining momentum with a significant increase in the number of organizations ready to take advantage of the technology. This architecture will reduce the cost incurred in the development and deployment of an on-premises database system both in hardware and software required for such a system. Multi-tenancy is a familiar word that comes from the real world of estate building. An estate building will provide a multi-tenant housing service to any number of tenants [1]. This could be an individual, couple, family or groups of different sizes. Similarly, a multi-tenant database is one which provides database support to a number of isolated and different groups of users. The concept of multi-tenancy was developed from the service providing technology known as Software as a Service (SaaS). SaaS is a form of cloud computing that involves offering software services in an on-line and on-demand fashion with the internet as the delivery mechanism [2]. The SaaS providers interface with the end users by virtue of the provisioning of business application services similar to the ones that have been traditionally self-hosted by the corporate houses [3]. Software as a Service constitutes a fast-growing business model for the sales of software that is based on the principle of outsourcing. With SaaS, a service provider hosts an application or software on its infrastructure and delivers it as a service to several organizations. An organization, also referred to as a tenant, subscribes for the service and accesses it across the Internet through standard web technology [4].

A Multi-tenant Database refers to a principle where a single instance of the database runs on a server, serving multiple client Database Management System (DBMS) organisations (tenants). A MTD is one which provides database support to a number of separate and distinct groups of users, also referred to as tenants. A tenant is simply any logically defined group of users that requires access to its own set of data. This definition was substantiated by Bezemer et al [5] as an architectural pattern in which a single instance of the software is run on the service provider’s infrastructure, and multiple tenants access the same instance. This concept provides the ability of a system to provide database management services to different users or customers without causing interference with each other’s processes. This reduces the effort made in production and the cost incurred in the development. Cloud computing users can discard the following difficulties, which include: The large-scale investments in hardware and software platforms; upgrading them regularly and the expensive licenses of application software used to run business processes, related transactions and decision-support systems [3].

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 and databases based on a scalable, shared hardware and software infrastructure [6]. Such a database system must be able to maintain or even increase its performance or efficiency level under larger operational demands. This multi-tenant database system is a new technology that can be implemented in both host based and cloud based environments.

Today many companies want to outsource their data to a third party which hosts a Multi-tenant Database System to provide a data management services. Each company is called a tenant. The multi-tenant data management system spreads the cost of hardware, software and professional services to a large number of tenants and thus significantly reduces per-tenant cost by increasing the scale. Thus the multi-tenant database system provides excellent performance, low-space requirements and good scalability [7]. With multi-tenancy, there is no need for customers to be separated on site systems for their personal applications since such services and resources could be rendered to them through a service provider at a much more reduced cost.

A single instance of the software running on a single server will serve multiple client organisations, this is a principle also used in software architecture and is also referred to as multi-tenancy. Multi-tenancy can minimize hardware/software costs and human costs per tenant. Multi-tenant database systems have been exploited to store, manage, and retrieve data of tenants. A service provider hosts the multi-tenant database system and each tenant subscribes to the services by doing necessary configuration, loading data to the data centre, and then interacts with the services through a standard method, e.g., Web Services. Thus, the cost of ownership of database applications and the maintenance costs are transferred from the individual tenant to the service provider [8].

There are still some factors that need to be considered before allowing data to be hosted by a third party database service provider’s platform. These factors are presented in this paper and examine the degree of impact and subsequently a framework is developed for informed decision making by tenants to guide their decisions about the concept.

The structure of this paper is as follows: Section II provides the details about the different factors that influence the choice of MTD and their re-grouping. The explanation of each factor is also presented. Section III introduces a framework developed based on a qualitative literature review of the subject area. Section IV presents the results from a survey carried out. Section V shows details of findings and discussions based on the results from data analysis. Section VI presents the conclusions and ideas for future work.

II. Factors Influencing the Choice of Multi-Tenant Database Approaches

There are factors that help in determining the most suitable and appropriate approach of Multi-tenants database. The use of the system should be one of the influencing factors towards the decision. Elmore et al [9] emphasise that the tenant application and usage requirements should be the primary consideration in deciding the right model of multi-tenant database. Sometimes users (tenants) are not equipped with the necessary information about this before taking a decision on what approach to adopt. Their decision is sometimes influenced by what vendors tell them. There is a need to examine all of these basic factors before approaching a service provider in order to make the right decision on this.

Some of these factors are itemized by Keemti [10] as follows.

1. Size of tenant database.
2. Number of tenants.
3. Number of users per tenant.
4. Growth rate of tenants.
5. Growth rate of tenant database.
6. Security.
7. Cost.
8. Flexibility – ability to create multiple tables by tenants.
9. Regulatory considerations (UK/EU countries)

According to Khan et al [11] while implementing Multi-tenancy, some factors must be analyzed to successfully implement the Multi-tenant applications which are:

1. System Cost
2. System Complexity
3. System Performance
4. System Security
5. System Scalability
6. System Flexibility

Some of these factors were mentioned by different scholars depending on the context of their research. Matthew et al [12] has now regrouped these into four major headings as follows:

1. Cost consideration.
2. Growth consideration.
3. Security consideration.
4. Regulations consideration.

A. Cost Consideration.

Cost is vital when considering the appropriate approach to be adopted in the implementation of database multi-tenancy. This cost is referred to as Total Cost of Ownership (TCO) which is broken down into three major types. These are infrastructural cost, management cost and application development cost [13]. Infrastructural cost includes the cost of hardware, software and utilization costs. Management costs are costs related to the operational activities and processes like lifecycle management, monitoring data backup and restoration. Application development costs are costs related to meeting each customer additional unique requirement [13]. Considering all these costs involved in acquiring a dedicated database system, it is obvious that huge amount of resources are involved. Therefore the intending tenant might want to consider an approach that will require less cost of development than a large-scale development approach.

B. Growth Consideration.

A decision was made to group all points 1-5 of the above mentioned factors as part of growth consideration factor by Matthew et al [12]. Jacobs and Aulbach [14] carried out an experiment on memory (storage) and disk usage of the five different databases. This experiment focused on the shared process approach which is a model type of multi-tenant implementation. This shows that the size of tenant database in terms of storage capacity is a factor needed to be considered during the decision about choice of approach.

Schiller et al [4] give an illustration that the shared table approach has a promising benefit for a service provider that has a target for long tail because it offers the lowest overhead per tenant and thus is suitable for a large number of small tenants. An example of 1000 tenants with each uses less than 50MB of data and at most 5 concurrent users. This is an indication that the number of tenants on the database and the number of users per tenant are factors which must be thoroughly examined and also contribute to the performance of the database system which invariably is also a factor to be considered when taking the decision on what approach to be used. Myer in Schaffner [15] reveal that in October 2007, the SaaS CRM vendor RightNow had 3,000 tenants which were distributed across 200 MySQL database instances with 1-100 tenants per instance. This means that the CRM can evolve overtime. It has been designed to accommodate more tenants. The growth rate of tenants on it is never limited to a small number.

C. Security Consideration

In Multi-tenant Database Systems, one of the major concerns is the risk of data being exposed to third parties. Based on the fact that every database is designed to store sensitive data, the prospective tenant will have very high security expectations. Every service provider will always want to operate at the highest security standard but sometimes this might not be to a hundred per cent. Therefore the Service Level Agreement (SLAs) will have to provide very strong data security guarantees.

Some of the security issues related to Multi-tenancy Database include data isolation among the tenants. Gao et al [6] mention some of the challenges facing the ISVs (Independent Service Vendor/Provider) in delivering their service these include the data security isolation among tenant, the different tenants having different SLA demand, customization requirements and effective database scale out mechanism as the number of tenant increases. Hui et al [16] itemises the problems faced in providing database as a service which includes security, contention for shared resources and extensibility. Hui et al [16] also mentioned scalability as one of the problems which was defined as the ability to serve an increasing number of tenants without too much query performance degradation. In spite of the increase in the tenancy and query requests, the system should still be able maintain its performance level. This tenancy growth should not impact the service availability to other tenants.

Another major security issue is the flexibility of the system. A MTD must be able to serve hundreds and thousands of tenants in one instance. Aulbach et al [17] expand this by saying that a Multi-tenant Database must be flexible by being able to extend the base schema to support multiple specialised versions of application and to dynamically evolve the base schema and its extension while the database is on line.

It was also pointed out by Vashistha and Ahmed [18] that isolation should be carefully considered in almost all parts of architecture design from both the functional and non-functional level such as security, performance, availability, administration and also support tenant customization of their own services in runtime without impacting on others.

D. Regulatory Consideration.

There are laws and regulations put in place by different governments that serve as protection to databases of different entities that operate in that geographical location. Companies, organisation and governments are often subject to regulatory laws that affect their security and even record storage needs. The knowledge of these different laws and regulations are considered when making decision about multi-tenant database. Chong et al [19] argue that the investigation of regulatory environments that your prospective customers occupy in the market in which you expect to operate is important. This shows that you have to be conversant with the laws that operate in that area or country. It is important to find out whether there are any aspect of the law that present any condition that will influence your decision toward given your database service to a third party.

There are challenges when tenants are from different regulatory authority. Most of the countries have different laws which sometimes are not having same conditions and to now manage these conflicts will pose a great technical and economical challenge to the ISVs. Effort must be made to look into the regulatory provisions for prospective tenants. The regulatory laws of all the tenants on a particular multi-tenanted database must be the same in order to have harmony in the laws that govern the system.

All these are major consideration that must be considered to make a good decision about the adoption of a multi-tenant database model depending on the tenant individual system requirements.

III. Framework Foundation

The foundation of the framework is based on the findings from different scholars' view of the concept MTD. Software as a service (SaaS) as the bedrock of the concept, facilitates the acquiring of huge number of small tenants by providing low service fees. Therefore it is essential to reduce cost per tenant to achieve per tenant low service cost.

A. Cost Factor and its Postulate

The goal of every organisation is to find a costminimal solution for the considered application. This is one of the advantages pointed out by Walraven et al [2] that multi-tenancy achieved a high operational cost efficiency by sharing the same resources among multiple customer organizations. In discussion of multi-tenancy in relational databases, Narasayya et al [20] also attest to this point giving the example of Microsoft SQL Azure where resources of a single database server can be shared among multiple tenants to achieve cost reduction. Multi-tenant is crucial for this purpose since dedicating a machine for each tenant makes the service prohibitively expensive. Yaish and Goyal [21] also allude to the fact that multi-tenant database helps reduce the Total Cost Ownership (TCP) from the tenants' perspective, by sparing them spending money and effort on writing SQL queries and backend data management codes. According to Khan et al [11] the primary factor which is highly influential on Multi-tenant applications in a positive way is Cost. Instead of dedicating separate infrastructure for each individual tenant the infrastructures are shared by virtualisation. Khan et al [11] said this results in significant decrease in system cost. This obviously will drive users towards the adoption of MTD. This has proved that cost is minimised in the adoption of a MTD which means that subscribers will want to embrace the concept for the purpose of saving cost. Therefore, based on all the position of different scholar it shows that tenants will choose MTD from any reliable vendor as a means of storing and protecting their database.

COST FACTOR → MTD ADOPTION

Postulate 1

B. *Security Factor and its Postulate*

Gao et al [6] explains security in terms of isolation among tenants. Gao et al argues that access to other tenants' data is sometimes granted by the system admin but if not controlled can be a source of data leakage. There should be a more flexible, fine-granular access control mechanism which will prevent SQL injection among tenants. Furthermore Gao et al highlights that MTD has issues related to customisation of applications to the business needs of every tenants, considering the fact that each tenant might want different SLA (Service Level Agreement).

Khan et al [11] allude to the fact that security is a major concern when it comes to multi-tenancy. There are two approaches of security which are overall system security and individual security level of a tenant [11]. Once one of the security approaches is affected. It will also affect other and the system as a whole. The system architecture should be designed such that it should be scalable considering the future requirement. If this is not done it will affect the security of the system. Remember that security is a key credential when it comes to a standard system, but security is compromised by sharing of resources to achieve multi-tenancy [11]. So according to Khan et al resources sharing makes application more scalable and flexible but does not give one hundred percent secured system.

Pippal and Kushwaha's [22] proposed approach for multi-tenant database is the shared database shared schema technique which has the advantage of suitability for large number of tenants. Cost minimised and leveraged benefit of using same hardware, software, database, schema and table for all tenants. Pippal and Kushwaha proves that this guarantee good isolation and security. While the second approach shared database separate schema will be limited to the number of instances of the database supported by the DB server. This now means that adding more tenants will cost more.

The use of shared database shared schema technique however requires the implementation of extension table. Extension tables contain a lot of information for Meta data i.e. for a single row of table of a tenant that consists of four columns, the Record_ID and Extn_ID are repeated four times, and this information introduces a kind of redundancy [22]. Li et al [23] argues also that to guarantee high level of performance in terms of performance and data integrity of results from query operations in a large multi-tenant database, adequate pivot table for tenant data are set up to speed up the query process. This will lead to having several stored tables which might not be in use after the query operations. This is data redundancy. Data redundancy is one of the major issues that affect organisation data management and security. When queries are issued depending on the number of queries the processing time is increased because of this factor.

Luo et al [24] argues that a lot of memory is consumed on maintaining a huge number of tables for a large number of inactive tenants. This will unavoidably limit the number of tenants that can be consolidated into the same DBMS instance. This implies that as time goes on so many tables would have been created and while few of these will be used or queried. These idle tables would occupy a lot of system resources [24]. This will invariably lead to degrade system efficiency and scalability. As number of tenants increase the number of table also increases and the system becomes very slow and inefficient. The performance of the system increasingly becomes very slow, not meeting up with the demands of tenants. This is a major issue that lead to lack of security. This is also a reason for the slow pace towards the multi-tenant database.

Luo et al [24] identifies the issue of scalability which refers to the ability to serve an increasing number of tenants in getting an acceptable level of performance which seems to be a problem of multi-tenant database. A previous study by Aulbach et al [25] also observed very high significant performance degradation on a database server when the number of table exceeds 50000.

Based on all these above mentioned reasons, tenants might decide to shift away from the adoption of a MTD.

SECURITY FACTOR → MTD REJECTION

Postulate 2

C. Growth Factor and its Postulate

Growth of Multi-tenant database is an aspect that DaaS providers always want to handle before it hampers the smooth running of their platform. The growth is in different aspects which include the increase in the number of tenant, the increase in the storage capacity of each tenant and also the increase in the number of users per tenant. This seems to be a concern to both the tenants and service providers.

According to Narasayya et al [20], there is a serious problem with varying and increasing tenant's workload contending with other tenants for key resources such as CPU, I/O and Memory. Tenants of a relational database-as-a-service platform can execute arbitrary SQL queries that can be complex and whose resource requirements can be substantial and widely varied. This tends to affect the performance of the MTD system. This is because the performance of individual tenant's workload can vary significantly depending on the workload of other tenants contesting for the shared database resources. Narasayya et al further illustrate that tenant's data size, distribution and access pattern can change over time. According to Narasayya et al these factors contribute to even greater variability in query throughput and latency. When the throughput (i.e. queries/sec) is affected then performance is also affected and hence leads to a very slow system. This suggest that there must be a mechanism in place by MTD providers to reduce to barest minimum the variability in the performance which occurs as a result of contention with other tenant for very important shared database resources.

Lang et al [26] refers to growth as the degree of multi-tenancy which is explain to mean the increase in the number of tenant and the increase in the storage space occupied by each tenant. Lang et al alludes to the fact that the degree of multi-tenancy is an additional factor that impacts performance both for the overall system and the performance experienced by each individual tenant. The increase in the degree of multi-tenancy decreases per-tenant performance. There is however a high reduction in the overall operating cost for the MTD providers. The MTD providers always adopt scheduling method of provisioning the resources to accommodate the tenants for each resource. This will help improve the performance level and thereby still enjoy the reduced overall operational cost for the MTD provider which also implies that the tenants' cost will also be reduced. This provisioning of resources and other measures to minimized the variability in performance will encourage more tenants subscribe to the technology MTD.

Gao et al [6] also explains along this line that the maximum number of tenants supported by multi-tenant system should increase in direct proportion to the increase of resources. This will ensure that the performance level of each tenant will remain in an acceptable level. This provides a more cost-effective and smooth scalability that will enhance system performance. Therefore growth as a factor will encourages the adoption of MTD concept.

GROWTH FACTOR → MTD ADOPTION

Postulate 3

D. Regulation Factor and its Postulate

There are laws and regulations put in place by different governments that serve as protection to databases of different entities that operate in that geographical location. Companies, organisations and governments are often subject to regulatory laws that affect their security and even record storage needs. The knowledge of these different laws and regulations are considered when making decision about multi-tenant database. The entire responsibility of database management, i.e., database backup, administration, restoration, database reorganization to reclaim space or to restore preferable arrangement of data, migration from one database version to the next without impacting availability which befall organization is now transferred to database as a service provider [27]. This aspect of administration, restoration and database reorganization will require protecting laws or regulations that will safe guard the interest of both the tenant and the service provider. There are federal, state, and local laws and regulations. Weis and Alves-Foss [28] also shows how each of these applies depends on the situation, the service level agreement (SLA) between the customer and provider, and data location. The latter brings a new concern because users do not control and often do not know where their data is physically located. This introduces a slew of legal problems, especially if their data crosses international boundaries. This shows that the service level agreement will take care of all the security flows that may occur as a result of regulations.

The SLAs for SaaS application differ per tenant. According to Gey et al [29], the SLA violation and service disruption caused by updating and upgrading the entire SaaS application for the sake of maintenance and evolution is unacceptable. This cause service to be disrupted at the tenant's point during the run time and eventually results in multiple co-existing versions of individual components and such brings about what we might refer to as management complexity. Gey et al however proposes a powerful run-time adaption mechanism that will allow the provider to update and upgrade the platform in a gradual process on a fine-grained, per-tenant basis. To maintain a good quality and also comply with SLAs throughout an upgrade, the multi-tenant SaaS application must ensure on-going tenant request and also keep accepting and processing new request. These are covered by the SLA agreement and understanding that exist between the service provider and the tenants.

Considering the complex regulatory issues surrounding data protection across various jurisdictions, the inability to know where one's data is located, or if and when the data may be moved to another state or country, implies a good deal of potential legal risk [30]. There are numerous state and federal privacy and data security laws and regulations mentioned earlier on requirements that are triggered as result of storing your data on a third party platform. As mentioned earlier, the hosting of data owned by another tenant belonging to a different data protection authority will adequately be taken care when the host provider operates what is called Safe Habour Program e.g U.S Harbour Program [30]. This means that the receiving country or state or authority where the provider resides is protected.

In conclusion, this shows that irrespective of the state or authority under which your service provider occupies, there will be a protective regulation to safeguard the tenant's data. In the case where a safe habour program is absent there will be a discouragement to MTD, and prospective tenants would not want to take the risk of given their data to a third party host who cannot guarantee the safety of it. Therefore regulation should either encourage adoption or rejection of Multi-tenant database.

REGULATION → MTD ADOPTION

REGULATION → MTD REJECTION

Postulate 4

All these formed postulates were put together to form the framework of Multi-tenant database adoption shown fig 1 below

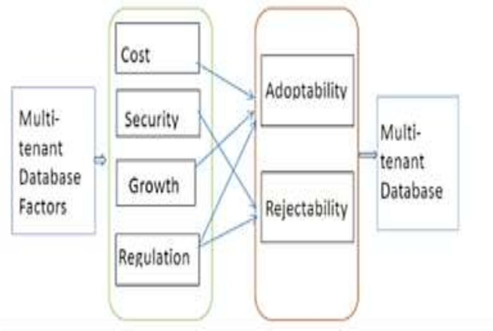

Fig.1. Framework of Multi-tenant Database Adoption

IV. Survey Results

A. Data and Methods

A survey was carried out using quantitative approach where questionnaire were used as the survey instrument. One major reason for using a survey is that surveys are viewed as the most appropriate method of studying participants' behaviour and job perceptions [31; 32]. According to Burns [33] and Creswell [34] a questionnaire is one of the best means of obtaining survey data. The questionnaire was designed carefully to make sure that it demonstrates and draws out useful responses. It was made easy to read and understand, short and have potency of being completed quickly. There are simple instructions given to the participants at the beginning of every section for easy, personal and independent inputs. The data extracted from the questionnaire were carefully analysed with the use of a simple quantitative statistical method called weighted score method. The UK Oracle user group (UKOUG) members were the participants in this survey. This is an association formed since 1983 with the mission to serve the Oracle community. This is based in United Kingdom and made up of over 8,500 people working for a variety of Oracle customers, Oracle partners and Oracle corporations. They come together every year with the purpose of education, innovation and information. The survey was conducted in the December, 2013 conference held in Manchester. For ethical reasons, all respondents' personal details were made anonymous. Thirty responses were collected and carefully analysed to obtain the outcome of this paper.

This group of people were selected because of their high level of involvement with databases and also because of the recent product of Oracle called 12C which is based on the multi-tenant concept.

B. Analysis

The graphical representation of the responses based on each factor is shown below. This graph shows the lines of the rankings in relation with each factor.

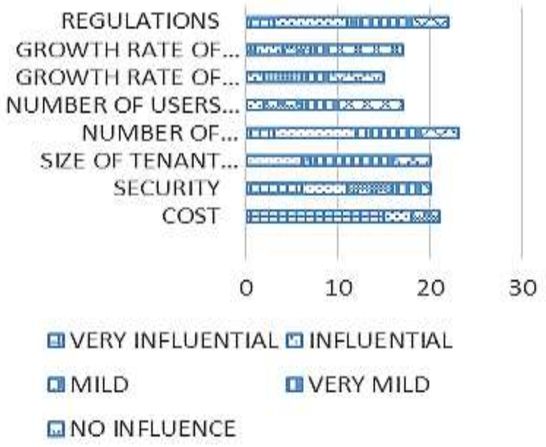

Fig.2. Graphical Summary of Respondents' Data

The graph above shows that cost has the highest value for very influential with 15 respondents agreed for cost as the most influential factor. While none of the respondents indicate that cost has no influence. The growth rate of number of tenants has the highest value for no influence with 8 respondent agreed that it is of no influence to the adoption of MTD. All the factors (1-5) above that makes the growth factor has values for no influence and when put together it shows that growth might has the least effect on the decision to adopt MTD.

In analysing this data further, a simple quantitative statistical method was adopted known as weighted score method also known as numerical indicator [35] as shown in Equation (1) below.

$$\frac{N_1 * R_1 + N_2 * R_2 + N_3 * R_3 \ldots\ldots + N_n * R_n}{N_1 + N_2 + N_3 \ldots\ldots\ldots\ldots\ldots\ldots N_n} \quad (1)$$

Where N = the frequency of each ranking.

R = the ranking value.

Very influential – 5, influential – 4, Mild – 3, very mild – 2, No influence 1

The above (1) is applied for Cost as shown below.

$$\frac{(15*5) + (3*4) + (3*3) + (0*2) + (0*1)}{15 + 3 + 3 + 0 + 0} = 4.57$$

The results from the method are represented in the table 1 below.

Table 1 - Table of Results

Factor	Result
Cost	4.57
Security	3.6
Size of Tenant Database	2.45
Number of Tenants	3.04
Number of Users per Tenant	2.05
Growth rate of Number of Tenants	2.13
Growth rate of Tenant's Database	2.24
Regulations	3.0

V. Findings and Discussion

The respondents' data shows that some gave ranking to some of the factors while some factors were not ranked. This might be that some of these factors that were not ranked don't have impact or influence on the decision towards MTD. The result of the analysis shows that cost has the highest indicator value of 4.57 which means it has the greatest impact based on the perspective of the respondents. This means that cost should be first factor to be considered. This is the total cost of ownership which includes some other cost as explained above. Since every organisation is set up for business purpose that means that any form of saving cost and increasing revenue will inform the decision to be taking.

Security is another factor that has second indicator value of 3.6 and should be consider after cost has been taking into consideration. Data safety remains one major aspect any organisation would like to maintain. The risk of data being exposed to another party must be guided against in MTD so that prospective tenant will be confident with the service provided. There should be a good service level agreement to protect your data policies.

Regulations have a value of 3.0 which means that this is consider even before the growth factor. The regulations that govern the environment of your tenants as a service provider must be considered. Tenants should be conversant with the laws and regulations that protect your database before even considering the MTD service provider to engage. These regulations differs based on regions, continent, nations or state where tenants are located.

Growth which includes the following factors; the size of tenant database, number of tenants, number of users per tenant, growth rate of number of tenants and growth rate of tenant's database has an average value of 2.38 shows that is the least factor that should be considered. This might suggest that it will have the least degree of impact on the decision about MTD.

VI. Conclusion

This paper introduced a framework for the adoption of MTD where it has identified and structured all the factors showing the direction it will go putting these militating factors into consideration. Thus, out of these four factors, three encourages MTD adoption while one discourages it. One of these also shows a two-way direction. There appears to be a further study required in the aspect to ascertain this model.

This research has shown that MTD will enable service providers to transform fixed services to variable services of their technology footprint to enhance business agility, optimize their operations and lower their operational cost thus driving business. Also, the study has been able to identify the factors and proved that these identified factors have real impact on the adoption of MTD. The set of responses obtained from the survey's participants were able to prove the impact of these factors through their contribution in this research. The outcome however has not shown whether it is positive or negative impact this has not been proved.

Finally, the work has shown that MTD helps reduce the TCO involved in acquiring a dedicated database system. The growth of MTDs includes the number of tenants, the number of users per tenant, the size of each tenant's database, growth rate of tenants and the growth rate of tenant's database should all be a consideration in adopting MTD. Technical issues like data isolation, scalability, flexibility and customization are all incorporated in the security consideration. Finally, the regulations that govern all tenants in a MTD must be made harmonised for adherence to law.

References

[1] Banville, R. and Holzel, R. (2012) Openedge multi-tenancy overview: Progress software [online]. Bedford: [Accessed 23 January 2013]. Available at < http://www.progress.com>

[2] Walraven, S., Van Landuyt, D., Truyen, E., Handekyn, K. and Joosen, W. (2014) Efficient customization of multi-tenant Software-as-a-Service applications with service lines. Journal of Systems and Software [online] (0), Available at:<http://www.sciencedirect.com/science>

[3] Al-Aqrabi, H., Liu, L., Hill, R. and Antonopoulos, N. (2015) Cloud BI: Future of business intelligence in the Cloud. Journal of Computer and System Sciences [online], 81(1), pp. 85-96 Available at:<http://www.sciencedirect.com/science>.

[4] Schiller, O., Schiller, B., Brodt, A. and Mitschang, B. (2011) Native support of multi-tenancy in RDBMS for software as a service Proceedings of the 14th International

Conference on Extending Database Technology. [online]. Uppsala, Sweden New York, NY, USA: ACM, pp.117-128.

[5] Bezemer, C., Zaidman, A., Platzbeecker, B., Hurkmans, T. and Hart, A. (2010) Enabling multi-tenancy: An industrial experience report Software Maintenance (ICSM), 2010 IEEE International Conference on. [online]. pp.1-8.

[6] Gao, B., An, W., Sun, X., Wang, W., Fan, L., Guo, C et al. (2011) A Non-intrusive Multi-tenant Database Software for Large Scale SaaS Application e-Business Engineering (ICEBE), 2011 IEEE 8th International Conference on. [online]. pp.324-328.

[7] Ni, J., Li, G., Zhang, J., Li, L. and Feng, J. (2012) Adapt: adaptive database schema design for multi-tenant applications Proceedings of the 21st ACM international conference on Information and knowledge management. [online]. Maui, Hawaii, USA New York, NY, USA: ACM, pp.2199-2203.

[8] Ying, H., Wang, Q., Wang, Z., and Wang, N. (2011) DB2MMT: A Massive Multi-tenant Database Platform for Cloud Computing e-Business Engineering (ICEBE), 2011 IEEE 8th International Conference on. [online]. pp.335-340.

[9] Elmore, A. J., Das, S., Agrawal, D. and Abbadi, A. (2011) Towards an elastic and autonomic multitenant database Proc. of NetDB Workshop. [online].

[10] Keemti, P. (2010) Multi-tenant Database Architecture [online]. [Accessed 23 August 2013]. Available at <http://www.msdn.microsoft.com/eus/library/aa479086.aspx#mlttntda_topic1>

[11] Khan, M.F. and Ullah, M.A. (2012) an Approach Towards Customized Multi-Tenancy. International Journal of Modern Education and Computer Science (IJMECS) [online], 4(9), pp. 39.

[12] Matthew, O., Dudley, C. and Moreton, R. (2014) A Review Of Multi-Tenant Database And Factors That Influence Its Adoption. UKAIS 2014 Conference [online]. Oxford, UK.

[13] Wang, Z., Guo, C., Gao, B., Sun, W., Zhang, Z and An, W. (2008) A Study and Performance Evaluation of the Multi-Tenant Data Tier Design Patterns for Service Oriented Computing e-Business Engineering, 2008. ICEBE '08. IEEE International Conference on. [online]. pp.94-101.

[14] Jacobs, D. and Aulbach, S. (2007) Ruminations on multi-tenant databases. BTW Proceedings [online], 103pp. 514-521.

[15] Schaffner, J. (2013), Multi tenancy for cloud-based in-memory column databases: workload management and data placement, Springer, Wien

[16] Hui, M., Jiang, D., Li, G. and Zhou, Y. (2009) Supporting Database Applications as a Service Data Engineering, 2009. ICDE '09. IEEE 25th International Conference on. [online]. pp.832-843

[17] Aulbach, S., Jacobs, D., Kemper, A. and Seibold, M. (2009) A comparison of flexible schemas for software as a service Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. [online]. Providence, Rhode Island, USA New York, NY, USA: ACM, pp.881-888.

[18] Vashistha, A and Ahmed, P. (2012)"SaaS Multi-Tenancy Isolation Testing- Challenges and Issues", International Journal of Soft Computing and Engineering,[online] 2(5), pp. 49-50 pp. 49-50 Available at:http://wlv.summon.serialssolutions.com

[19] Chong, R., Carraro, G., Wolter, R. (2006) Multi-tenant Data Architecture [online]. [Accessed 23 August 2013]. Available at :<http://www.msdn.microsoft.com/en-us/library/aa479086.aspx#mlttntda_topic3>

[20] Narasayya,V. R., Das,S., Syamala,M., Chandramouli,B. and Chaudhuri,S. (2013) SQLVM: Performance Isolation in Multi-Tenant Relational Database-as-a-Service.CIDR. [online].

[21] Yaish,H. and Goyal,M. (2013) A multi-tenant database architecture design for software applications Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on. [online]. IEEE, pp.933-940.

[22] Pippal, S.K. and Kushwaha, D.S. (2013) A simple, adaptable and efficient heterogeneous multi-tenant database architecture for ad hoc cloud. Journal of Cloud Computing [online], 2(1), pp. 1-14.

[23] Ni, J., Li, G., Wang, L., Feng, J., Zhang, J. and Li, L. (2014) Adaptive Database Schema Design for Multi-Tenant Data Management. [online].

[24] Luo, Y., Zhou, S. and Guan, J. (2015) LAYER: A cost-efficient mechanism to support multi-tenant database as a service in cloud. Journal of Systems and Software [online], 101(0), pp. 86-96 Available at:<http://www.sciencedirect.com/science>.

[25] Aulbach, S., Grust, T., Jacobs, D., Kemper, A. and Rittinger, J. (2008) Multi-tenant databases for software as a service: schema-mapping techniques Proceedings of the 2008 ACM SIGMOD international conference on Management of data. [online]. Vancouver, Canada New York, NY, USA: ACM, pp.1195-1206

[26] Lang, W., Shankar, S., Patel, J. M. and Kalhan, A. (2012) Towards Multi-tenant Performance SLOs Data Engineering (ICDE), 2012 IEEE 28th International Conference on. [online]. pp.702-713.

[27] Hacigümüs,H., Iyer,B. and Mehrotra,S. (2002) Providing database as a service Data Engineering, 2002. Proceedings. 18th International Conference on. [online]. IEEE, pp.29-38.

[28] Weis, J. and Alves-Foss, J. (2011) Securing database as a service: Issues and compromises. IEEE Security & Privacy [online](6), pp. 49-55 .

[29] Gey, F., Van Landuyt, D., Joosen, W. and Jonckers, V. (2015) Continuous evolution of multi-tenant SaaS applications: A customizable dynamic adaptation approach. status: accepted [online].

[30] Sotto, L.J., Treacy, B.C. and McLellan, M.L. (2010) Privacy and Data Security Risks in Cloud Computing. World Communications Regulation Report [online], 5(2), pp. 38.

[31] Mintzberg, H. (1973) The Nature of Managerial Work, Harper and Row, New York.

[32] Rea L. M. and Parker, P. A. (1997). Designing and Conducting Survey Research, 2nd Ed., Jossey-Bass Publishers, San Francisco, USA.

[33] Burns, R.B (2000) Introduction to research methods (4th Edition). SAGE Publications, London.

[34] Creswell, J. (2003) Research Design: Qualitative, Quantitative, and Mixed Methods Approaches (2nd Ed), Sage, Thousand Oaks, CA.

[35] Abeysekera, S. (2001) Analysis approaches in participatory work involving ranks or scores. DFID Theme Paper (revised).UK: Statistical Services Centre, University of Reading [online].

Authors' Profiles

Olumuyiwa Matthew received B.Tech and MSc degrees in Computer Science and Information Technology respectively. He is currently Management working towards the PhD degree with the School of Mathematic & Computer Science University of Wolverhampton, Uk. His current research interest is issues relating to the adoption of Multi-tenant database. He is an associate member of BCS the Chartered Institute for IT and a member of Computer Professionals of Nigeria (CPN)

Dr Mary Garvey is a senior lecturer at the University of Wolverhampton, specialising in teaching databases at all levels and web programming. She has several academic publications to her credit. She is a Member of the BCS CITP and is the Chair of the Wolverhampton Branch of the BCS.

Dr Kevan Buckley is the Head of School of Mathematics and Computer Science at the University of Wolverhampton so has diverse interests across the subjects. In particularly he is interested in the application of Machine Learning and Optimisation techniques to data from a range of disciplines from Social Media to Forensic Science. Techniques for data capture, processing and storage underpin this research

How to cite this paper: Olumuyiwa Matthew, Kevan Buckley, Mary Garvey,"A Framework for Multi-Tenant Database Adoption based on the Influencing Factors", International Journal of Information Technology and Computer Science(IJITCS), Vol.8, No.3, pp.1-9, 2016. DOI: 10.5815/ijitcs.2016.03.01