CLOUD COMPUTING DATA ARCHITECTURE

Harry Katzan, Jr.
Savannah State University

ABSTRACT

Cloud computing is an application architecture for accessing and hosting computing service over the Internet. Access is achieved with a web browser and service is supplied by software running on a cloud platform. Client benefits include lower up-front infrastructure cost and reduced time to deployment. Provider benefits are provided through the monetization of client tenancy, achieved through prudent service provisioning and operational efficiency. The paper addresses the securitization of client data in a cloud computing domain.

CLASSIC CLOUD COMPUTING MODEL

The essence of *cloud computing* is software deployed as a hosted service and accessed over the Internet. The two kinds of software in this category are business software and consumer software. Business software provides business services and emphasizes business solutions, such as CRM, SCM, ERP, and human resources. Consumer software is characterized by publicly oriented personal solutions, such as office and social applications. Software application, known as *Software as a Service* (SaaS), normally operate on a cloud platform.

A *cloud platform* resides in a cloud data center and exists as a powerful computing facility, a storage system, an advanced operating system, support software, and the necessary fabric to sustain a server farm and scale up to support millions of Internet clients. A cloud platform is as much about operating in the cloud, as it is about developing applications for the cloud. A cloud platform is usually employed to do the following: develop and run client applications, develop SaaS applications, and run SaaS applications.

The objective of cloud software vendors is to increase revenue by giving the client more money to spend on cloud software service by reducing operational costs and then to use economy of scale to equitably provide requisite computing services. If the software is built to scale well, the operating cost for each client will be less as additional clients are added.

Accordingly, a successful SaaS vendor will use multi-tenancy theory to develop effective software.

BUSINESS AND CONSUMER SERVICES

With business services, the most important consideration is whether the process is executed in-house or as a cloud service. When the process is handled in-house, total control over the operation is obtained along with limited opportunity for achieving economy-of-scale. As processes are distributed outward on the cloud, control is decreased but opportunities for achieving economy-of-scale are increased. The considerations are different with consumer services. Pure service, as with office applications, provides practically no control over the application to the client and a reasonably high-level of economy-of-scale to the provider. In many cases, consumer services are advertising supported and are complimentary to the client through advertising. Business applications that reside “on premises” are governed by the traditional considerations of application acquisition and deployment. If an application resides on and is deployed from the cloud, then a client gets a customized version achieved with a separate code base, or its

operational equivalent achieve through configuration options. The subject of business services is covered under "multi-tenancy."

The primary advantage of a cloud consumer service is that it is typically free to the client, as well as being accessible from any location via the Internet, and it yields advertising-supported revenue for the provider. Consumer services have a near-zero marginal cost of distribution to clients, because of the long tail, and require only a fraction of the number of clients to respond to advertising. This is the well-known *Freemium Business Model* [And04], characterized as follows: In the free sample product model, you give away 1% of your product to sell the additional 99%, whereas in the freemium model, you give away 99% to sell 1%. Because of the scale of the Internet with millions of users, you can reach a large market, so that the 1% is a huge amount. Consumer services would normally employ multi-tenancy.

CLOUD SERVICE ARCHITECTURE

A comprehensive SaaS application structure includes a continuum of architectural levels, based on the capability of handling multiple clients and software configurability. Four levels are identified. The number of levels in any specific operational environment is based on the cloud platform and its characteristics.

> **Level One.** At the first level, the users within a client domain address a single instance of an application running on a server. Each client/instance is totally independent of other client/instances running on the same server. This is the traditional hosted service operating in the cloud. Each software instance is individually customized for each client. This is the *single-tenant model*.
>
> **Level Two.** At the second architectural level, the vendor runs a sole instance that is shared by multiple clients. The feature set for each client is determined by configurable metadata, and authorization/security policies insure the separation of user data. This is the *multi-tenant model*.

The choice among architectural levels is determined by the provider/client's business, architectural, and operational models.

MULTI-TENANT ARCHITECTURE

From an architectural point of view, there are three features that identify an effective multi-tenant application: scalability, efficiency, and configurability. With a level one architecture, it is not possible to produce a distinctly customized version of the application software for each client, and at the same time, yield the economy of scale needed for successful monetization. This might be possible with single-client on-premises software, but with hundreds or millions of clients, producing and maintaining custom code is not feasible. Accordingly, even when a single-tenant environment is involved, metadata driven software is required.

With multi-tenancy, a single code instance services all clients with authorization and security policies in place to isolate client data. Instance options provide unique applicability, as required. This requires a trade-off between features and tenancy.

Business Model

The business model for cloud computing reflects how service providers can increase revenue and how clients can reduce operational costs of services over on-premises facilities. There are two areas that can

be addressed: the application architecture and the operational structure. From a monetization viewpoint, there are two options for application architecture: common features and unique features. For operational structure, the options are single-tenant and multi-tenant. The various options are regarded as service drivers and are summarized in Figure 1.

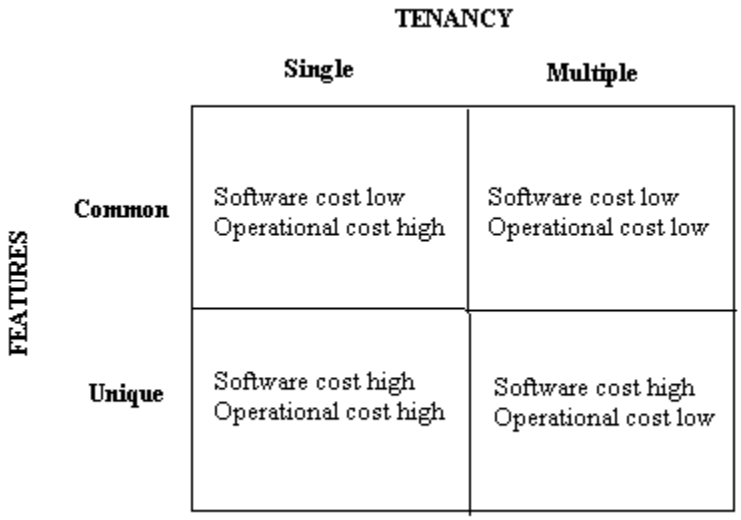


Figure 1. Service Drivers for Cloud Service Monetization.

The salient features of the cloud computing business model are summarized as follows:

- The ownership of the software is transferred to the cloud service provider.
- The responsibility for hardware, application software, storage facilities, and professional services resides with the provider.
- Systems software is available from a trusted vendor for supporting cloud services.
- Data centers are available for sustaining the operational structure and supporting the requisite fabric needed to utilize service farms.

Accordingly, the business model provides the economy of scale needed to target the long tail by providers and reducing up-front and operational costs for the client.

The SaaS provider with cloud computing will characteristically experience high up-front costs for infrastructure and software development. The SaaS client will have to give up a certain level of control to benefit from the economy-of-scale supplied by the provider. There are lingering questions over "who owns the software," "who owns the data," and information security.

Business Perspective

An element of software is essentially a collection of features – functions that perform a computational task. The set of features in a product or service support the selling aspects of that item. The cost of providing software services in a cloud environment are substantial and involve design, implementation, testing, marketing, and support activities. In short, complex software has a higher cost over simple software. When cloud software is architected for multi-tenancy to achieve sharing and economy-of-scale for the client, the costs are dramatically higher. Cloud software provisioning is a trade-off between

features and tenancy. Each of the items, i.e., features and tenancy, is actually a continuum, even though it is customarily treated as "lower" and "higher" to simplify the analysis and understanding. The scenario is inherent in the graphs in Figure 2.

Cost per tenant is the cost of delivering an element of software to one client and covers any requisite expenses needed for the associated service. Multi-tenant architecture maximizes sharing between users and increases the total revenue and economy of scale for the client; however, it increases development costs.

Cost per feature is the cost of implementing a specific functionality. Simpler features cost less to develop and maintain. Adding multi-tenancy increases the cost of a feature. The tenancy/feature conundrum is summarized as follows: "… multi-tenancy incurs a higher cost per feature, but lower cost per tenant while isolation has lower cost per feature but higher cost per tenant." [Car08b, p.1] Figure 5 summarizes in graphical form the advantages of each approach over time. As the tenant base grows (t3 in the diagram), multi-tenancy has good monetization for the provider.

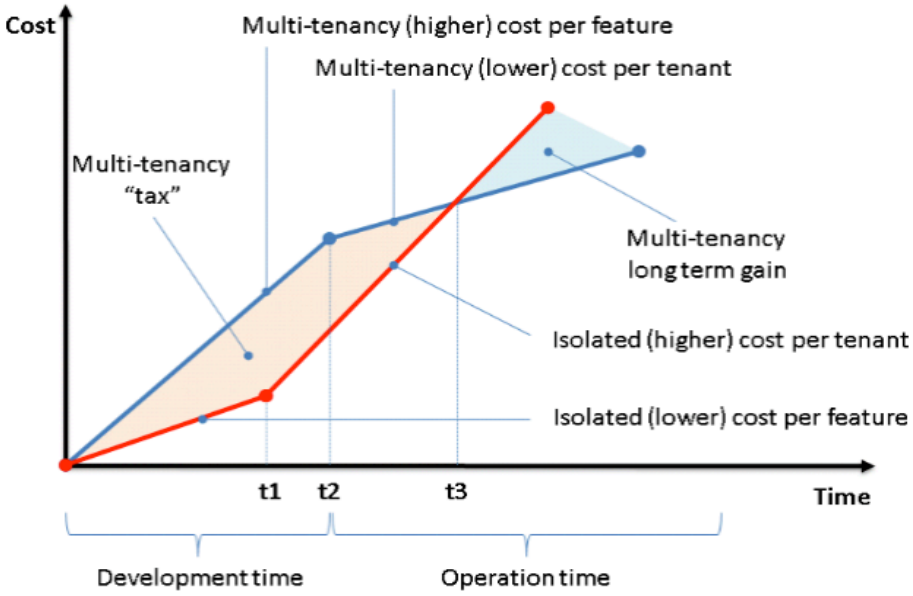


Figure 2. Cost per Feature vs. Cost per Tenant. [Car08b, p.2]

A promising approach to achieving economy-of-scale is to use virtualization at the cloud platform operating system level instead of designing an application for multi-tenancy.

MULTI-TENANT DATA ARCHITECTURE

Clearly, multi-tenant data architecture requires a correspondingly complex data model, ranging from data isolation to shared schemas as features range from common to unique and tenancy ranges from single to multiple, respectively. A continuum from isolated to shared is conceptualized, and three levels are identified: separate database per tenant, shared database and separate schema per tenant, and shared database and shared schemas per tenant.

Separate Database per Tenant

This option is suggested by Figure 3. In this case, computing resources and application code are shared among tenants, and each tenant has its own unique database. This instance supports single- and multi-tenant architecture. While this option more easily satisfies client needs, database maintenance cost are high and database server requirements are high, thereby limiting the number of tenants that can be supported by a specific computing system. This is a unique solution for clients with special security requirements, and are willing to pay for them.

Database Scalability

With cloud business applications, the game changes form mission criticality to Internet scaling to support millions of users. Two approaches can be envisioned: scaling the application, scaling the database, or both. In this paper, we are going to discuss data scaling.

Two approaches are: replication and partitioning. *Replication* refers to copying parts of the data to other tables or other databases and keeping the parts in synch. The basic idea is that the master copy is written to and replicated automatically to secondary databases for reference efficiency. *Partitioning* refers to moving "chunks" of data on a horizontal (row-based) basis to outlying tables or databases to increase efficiency, security, and manageability.

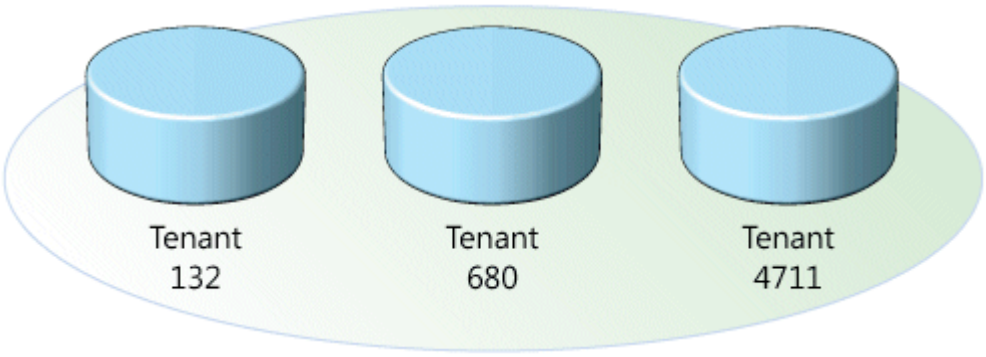


Figure 3. Separate Database per Tenant. [Cho06b, p. 2]

Shared Database, Separate Schemas

The option stores the data from multiple tenants in the same database with each client having its own set of tables and schema, as suggested by Figure 4. This approach requires a more complex software development, but provided added economy of scale for the client and the capability of servicing more clients.

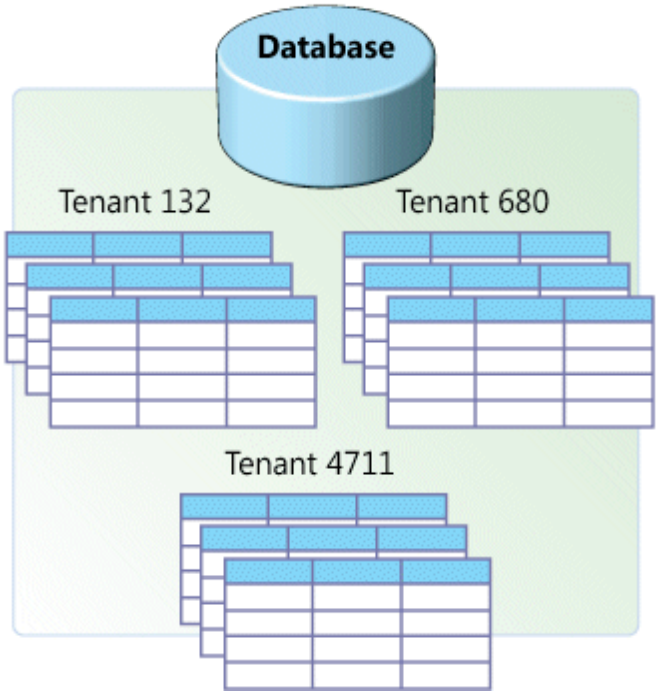
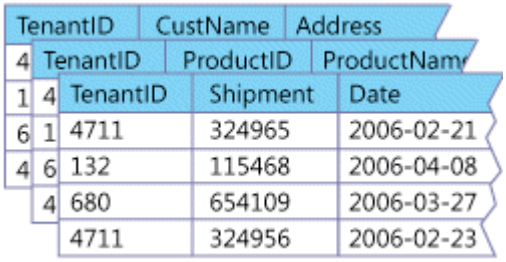


Figure 4. **Shared Database, Separate Schemas. [Cho06b, p. 3]**

Shared Database, Shared Schema

With this option, as suggested by Figure 5, tenant share the database and the tables are "logically" partitioned by tenant ID. This approach requires the most complex software but has the lowest hardware and software infrastructure costs. The number of database servers is lowest since tables are shared and database maintenance is reduced over the other options.



Firgure 5. Shared Database, Shared Schema. [Cho06b, p. 4]

SUMMARY AND FURTHER RESEARCH

Cloud computing has evolved into a huge research topic with each major player in the IT service business supplying its own version of exactly what the subject matter should incorporate. The above materials are

an attempt at finding a middle ground and providing a basis for further research in the area of small systems. Two very important areas have not been covered: security and data architecture. Both are significant for exchanging data between applications within the cloud.

REFERENCES

[And04] Anderson, C., "The Long Tail," *Wired Blog Network*, 2004.

[Car08a] Carraro, G., "I don't believe we are still talking about whether SaaS == multi-tenancy … ," *Microsoft Corporation*, blogs.msdn.com/gianpaolo/archive, (2008).

[Car08b] Carraro, G., "Monetization: the next frontier of SaaS/S+S architecture," *Microsoft Corporation*, blogs.msdn.com/gianpaolo/archive, (2008).

[Cha08] Chappell, D., "A Short Introduction to Cloud Platforms, " *Microsoft Corporation*, August 2008.

[Cho06a] Chong, F. and G. Carraro, "Architecture Strategies for Catching the Long Tail," *Microsoft Corporation*, April 2006.

[Cho06b] Chong, F., Carraro, G. and R. Wolter "Multi-Tenant Data Architecture" *Microsoft Corporation*, June 2006.

[Cho08] Chong, F., "Application Marketplaces and the Money Trail," *Microsoft Corporation,* March 2008.

[Kat08a] Katzan, H., *Service Science: Concepts, Technology, Management,* New York: iUniverse, Inc., 2008.

[Kat08b] Katzan, H., "Cloud Computing, I-Service, and IT Service Provisioning," *Journal of Service Science*, Volume 1, Number 2, (2008).

[Mil08] Miller, M., *Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online,* Indianapolis: Que Publishing, 2008.

[Rap04] Rappa, M., "The utility business model and the future of computing services," *IBM Systems Journal,* Volume 43, Number 1, (2004).