

Enforcing Multitenancy for Cloud Computing Environments

Jinan Fiaidhi, *Lakehead University*

Irena Bojanova, *University of Maryland University College*

Jia Zhang, *Northern Illinois University*

Liang-Jie Zhang, *Kingdee International Software Group*

Multitenancy is a defining characteristic of cloud applications. The shared infrastructure changes the underlying economics of enterprise applications, letting the vendor maintain a single application instance for thousands of customers (see www.appirio.com/ecosystem/appirio.swf). In a multitenancy cloud environment, multiple vendors, using the same infrastructure, can access and use an application. The application design thus must distinguish between customers to ensure they can't share or see each other's data.

Achieving Multitenancy

There are three different methods for achieving multitenancy: using a database, using virtualization, or through physical separation.

According to Lori MacVittie, a frequent cloud computing DevCentral blog writer, "In the case of software as a service (SaaS), multitenancy is almost always achieved via a database and configuration, with isolation provided at the application layer."¹ So, at the application layer, service providers should design and implement a specific class, and then create an object of the class in

a manner that serves the needs of multiple users in an effective way. Designing SaaS applications in this way will solve many issues related to multitenancy, such as the need for data security, data separation, and customized applications (to minimize the hard binding of runtime computing resources).

Many researchers consider virtualization to be an alternative to multitenancy.² However, virtualization is simply another technology for achieving multitenancy, especially for infrastructure as a service (IaaS). You can achieve virtualization using virtual machine technology that provides a hardware emulation layer over the real hardware. This technology can run multiple copies of server operating systems within one physical machine, and it can share physical hardware, such as network cards and disk storage, between the virtual operating system instances. Such multitenancy virtualization might reduce the costs and expenses of the provided services, but compared to multitenancy using databases technology, virtualization is more costly.

According to other researchers,³ you can achieve multitenancy via another dedicated technology

that provides resources to tenants individually. This technology is generally known as multitenancy via physical separation, but it's rarely used, because it relies on giving each tenant his or her own dedicated hardware resources.

Managing Multitenant Data

There are three approaches to managing multitenant data in the cloud:³

1. storing tenant data in separate databases, which is the simplest approach to data isolation;
2. housing multiple tenants in the same database, with each tenant having its own set of tables grouped into a schema created specifically for the tenant; or
3. using the same database and same set of tables to host multiple tenants' data.

Figure 1 illustrates the general architecture for representing multitenancy for effective cloud environments. This architecture employs customer integration on three layers:⁴

- the application layer,
- the infrastructure layer, and
- the data-center layer.

The data-center-layer multitenancy is well known and provides the highest level of security, if implemented correctly.

Infrastructure- and application-layer customer integration for multitenancy are new additions to the cloud computing topology design. The infrastructure layer dedicates one stack of software to a specific customer, deploying stacks for each customer account. The hardware requirements depend on actual service use.

For application-layer multitenancy, architectural implementations concern both the software and infrastructure layers. This type of multitenancy can compromise security, because application methods and database queries can access and store data from different user accounts. However, if implemented correctly, it can offer significant cost savings.

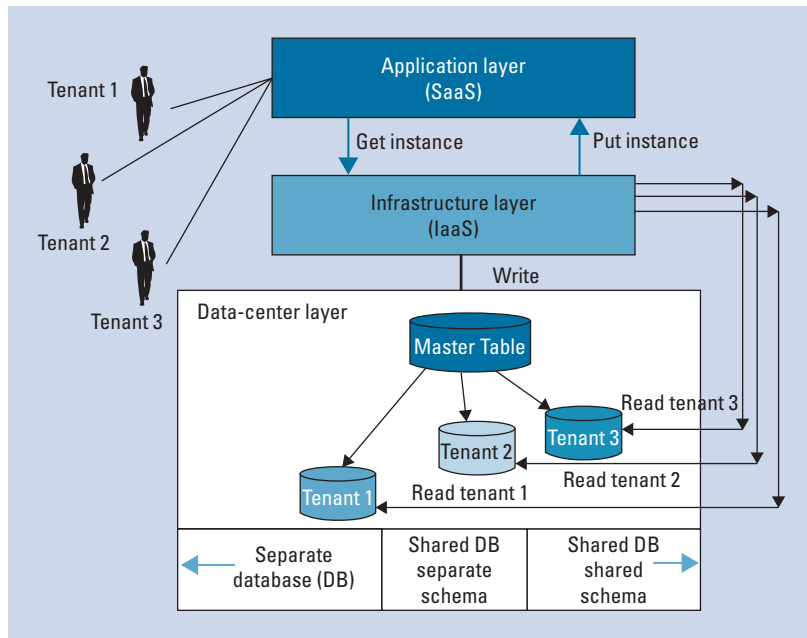


Figure 1. An overview of a general multitenancy cloud architecture.

While multitenancy on cloud environments provides seemingly limitless scalability and an alternative to the expensive data-center infrastructure, it raises new security and privacy issues, because it hands the processing and storage tasks over to third parties.

This requires building adequate security into every aspect of the SaaS application, as well as for every IaaS virtual service, using *filtering* (which provides an intermediary layer between a tenant and data source), *permissions* (which use access control lists), *encryption* (which obscures every tenant's critical data), or some combination of these techniques.

In This Issue


The articles in this special issue analyze many of the important issues in a multitenancy cloud environment, including the need for logical data model architectures, the difficulty of merging heterogeneous resources offered by different domains, and the need to secure services on the cloud.

The first article, "Logical Data Models for Cloud Computing Architectures," by Augustine Samba, examines existing cloud computing architectures and describes generic logical data models, which are independent of implementations. Samba analyzes logical models for two cloud architectures: the Distributed Management Task Force (DMTF) and the National Institute of Standard Technology (NIST). The data

models specify the logical interactions between the cloud entities, provide a framework for developing a common set of requirements for cloud architectures, and facilitate traceability between evolving business requirements and cloud architecture implementations.

The second article, "Cloud@Home: Toward a Volunteer Cloud" by Salvatore Distefano and Antonio Puliafito, describes a scenario for incorporating some elements of volunteer computing into the cloud computing paradigm. The result is a volunteer cloud that obtains its infrastructure by merging heterogeneous resources offered by different domains or providers (such as other clouds, grid farms, clusters, and data centers) or single desktops. This new "Cloud@Home" paradigm aims to merge the benefits of cloud computing (such as service-oriented interfaces, dynamic service provisioning, and guaranteed QoS) with those of volunteer computing (such as the use of idle resources and reduced operational costs). The mechanisms for aggregating, enrolling, and managing the resources, which take into account service-level-agreement and QoS requirements, are a mix of both worlds.

In the last article, "Threat as a Service? Virtualization's Impact on Cloud Security," by Hsin-Yi Tsai, Melanie Siebenhaar, André Miede, Yu-Lun Huang, and Ralf Steinmetz, describes the importance of security issues in cloud environments. The authors investigate cloud security by analyzing each cloud service model and how the impact of or potential for an existing attack in each model differs from that of conventional IT environments. The authors scrutinize the differences and implications in each service model, analyzing well-known security issues from the perspective of virtualization—a key driver of cloud computing.

Currently, individuals who want to access their desktop files remotely must accomplish certain setup requirements and keep their computers running. This isn't a very secure approach, so users who require remote access are starting to store files and applications in the cloud instead of on different physical devices (such as on laptops and in smartphones). This trend will only continue, as multitenancy with different access controls lets us share this wide variety of information securely. 

References

1. L. MacVittie, "Architectural Multi-tenancy," blog, 18 May 2010; <http://devcentral.f5.com/weblogs/macvittie/archive/2010/05/18/architectural-multi-tenancy.aspx>.
2. F. Chong, "Multi-tenancy and Virtualization," blog, 23 Oct. 2006; http://blogs.msdn.com/b/fred_chong/archive/2006/10/23/multi-tenancy-and-virtualization.aspx.
3. F. Chong, G. Carraro, and Roger Wolter, "Multi-Tenant Data Architecture," Microsoft, June 2006; <http://msdn.microsoft.com/en-us/library/aa479086.aspx>.
4. J. Meiers, "Best Practices for Cloud Computing Multi-tenancy: Building a Scalable, Available Cloud Computing Multi-tenancy Architecture," IBM, 5 July 2011; www.ibm.com/developerworks/cloud/library/cl-multitenantcloud.

Jinan Fiaidhi is a professor and graduate coordinator in the Department of Computer Science at Lakehead University, Canada. She's also an adjunct research professor at the University of Western Ontario, Canada. Her research interests include mobile and collaborative learning, ubiquitous computing, cloud computing, and Web-oriented architectures. Fiaidhi received her PhD in computer science from Brunel University, UK. She's a Professional Engineer of Ontario and a senior member of IEEE. Contact her at jfiaidhi@lakeheadu.ca.

Irena Bojanova is a professor and program director in the Information Technology Systems Department in the Graduate School of the University of Maryland University College. Her research interests include cloud computing, cybersecurity, Web-based systems, and educational innovations. Bojanova received her PhD in computer science and mathematics from the Bulgarian Academy of Sciences. She's a member of IEEE, the National Professional Science Master's Association, the US Distance Learning Association, and Sloan-C. Contact her at ibojanova@umuc.edu.

Jia Zhang is an associate professor in the Department of Computer Science at Northern Illinois University. Her technical interests center around services computing. Zhang has a PhD in computer science from the University of Illinois, Chicago. She's a member of IEEE. Contact her at jiazhang@cs.niu.edu.

Liang-Jie (LJ) Zhang is a senior vice president, chief scientist, and director of research at Kingdee International Software Group. He's also director of The Open Group. He's the founding EIC of IEEE Transactions on Services Computing. He's an IEEE Fellow and an ACM Distinguished Scientist. Contact him at zhanglj@ieee.org.