Service Oriented Cloud Computing Architectures

Asher J. Vitek

ABSTRACT

This paper discusses cloud computing and the eventual goal to make a uniform cloud computing architecture that will allow people to move from one cloud provider to another with ease. The paper talks about service oriented architecture and why it is so important in the eventual goal of a unified architecture. The paper discusses two proposed architectures and describes a small scale implementation of one of the architectures.

General Terms

Cloud Computing

Keywords

Service Oriented Architecture (SOA), Cloud Computing Open Architecture (CCOA), Service Oriented Cloud Computing Architecture (SOCCA), Multitenancy, Single tenancy

1. INTRODUCTION

In the recent years, as technology advances and more and more people have their own personal computers, cloud computing has become more popular than ever. Products like Windows 8 using cloud computing and many companies like Amazon and Google making use of cloud computing. The concept of a uniform architecture for all cloud providers has risen up. Concepts such as service oriented cloud computing have sprung up and have set goals to achieve a uniform architecture that all cloud providers can use so that all people can interact with all cloud providers in a uniform manner. There are many proposed architectures that put forward the ideas on how to make a cloud architecture that will do all this. This paper will take a look at Cloud Computing Open Architecture (CCOA) [8] and Service Oriented Cloud Computing Architecture (SOCCA) [5]. Both of these architectures aspire to improve upon current cloud architectures and make a unified architecture that all clouds should use. Both of these architectures use Service Oriented approach [6] [3].

UMM CSci Senior Seminar Conference Morris, MN.

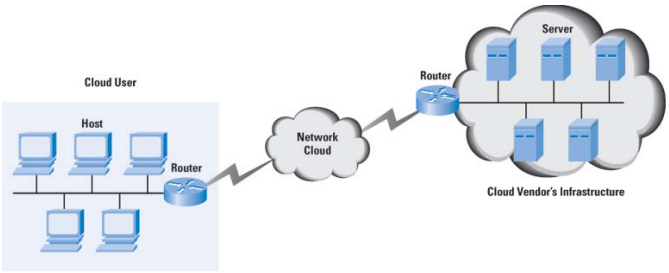

Figure 1: cloud computing [2]

2. CLOUD COMPUTING

Cloud computing is the use of shared computing infrastructure to provide IT services in the form of a large pool of systems that are linked together. In essence cloud computing is a large group of computers working together to provide a service. In Figure 1 an example cloud computing is shown with a large group of servers providing a service to users through a network.

Cloud computing has many different characteristics that define it. Some of the most prominent characteristics are elasticity and scalability, pay-per use, on demand, resiliency, multitenancy, and workload movement. Elasticity and scalability refer to being able to expand and improve the cloud as time goes on. Pay-per use and on demand mean that users can choose when they want to use a service and only pay for it when they use it. Resiliency means the cloud will be stable when any sort of failure occurs. Multitenancy will be discussed in section 3. Workload movement refers to being able to move and manage data through different servers. These characteristics make cloud computing good at adapting to change and different demands. If a cloud provider sees an influx of people using their product and needs more resources they can simply buy more servers or computing power and remedy the problem. Clouds like Amazons EC2 are pay per use and on demand.

There are three major models of cloud computing [2]. The first is software as a service (SaaS). SaaS is providing a service through the use of the internet instead of having an individual license that a company hands out to users of that software. Examples of SaaS are yahoo mail, Google mail and Quicken Online. Some of these services users pay for, like Quicken Online and others are free of charge, like Gmail. All of these services are handled by the provider of the service. All software updates are handled by the provider as well. All the user has to do in SaaS model of cloud computing is

pay and then use the application.

The second model of cloud computing is platform as a service (PaaS). PaaS is when a provider provides a software platform that the users can use to make their own applications. Example of PaaS are online development tools, such as Windows Azure, Google AppEngine, and Force.com. All of these products are tools that are used to develop programs online and using their infrastructure and are intended to make developing applications easier.

The third main cloud computing model is infrastructure as a service (IaaS). This is when a provider provides computing power and storage for their users. It is used for letting customers install their own operating systems on their portion of the cloud that they have purchased. An example of PaaS is Amazons Elastic Cloud computing cloud (EC2). Amazon rents users disk space, CPU power and memory to run operating systems and applications on the space users have rented.

3. SERVICE ORIENTED ARCHITECTURE

There are many things that are required to be a service oriented Architecture (SOA) and most of these qualities are features that we do not have in our modern architecture [3] [6]. This presents many problems for switching to SOA, but it must be done to improve current architectures and move forward to more efficient systems. The requirements for SOA are:

- The user must be able to switch between different clouds as long as they are compatible. An example would be if a user is running an OS on an IaaS cloud. They should be able to transfer their information to the new cloud provider they want to switch to.

- There must be a want to create a federation of resources. This means that even though there are many providers competing against each other; those providers must want to create a conglomeration of resources and want to work together at providing those resources. An example would be, two cloud providers work together at providing their combined resources through the same source.

These two requirements define a truly service oriented architecture. With this knowledge in hand, how would we achieve a service oriented architecture? What are the problems that have to be dealt with to achieve a service oriented architecture?

There are some problems with SOA itself as well. The main problem being that if all cloud providers can be switched between without consequence; then there is no reason for a user to stay with a company. Most companies want their customers to stay with them and if a user can leave a company without worry of loss of data or hassle there is no incentive for users to stay with a company. This implies that some companies are holding onto customers because the customer risks loosing their project. An answer to this is that better providers of the same services will thrive and others will dwindle and this is incentive for confident cloud providers to use SOA.

Though SOA is the most ideal architecture that a cloud computing world could have, the concept of SOA is currently not fully implemented and is not likely to be fully implemented. So why is SOA included at all? Why talk about SOA if it wont be fully implemented? The concept of the perfect architecture is something that companies and programmers should strive for. Though as a whole SOA may not and probably will not ever be accomplished, parts of SOA will at some point be accomplished. Being able to switch between providers with relative ease is something that could be possible and is now. Users can move information around from one provider to another right now, the user just has to do it themselves. For example users can import emails from one email to another. The problem is that it is not easy as it could be. The point of SOA is to make something that is best for the customer. Though it may not all be fully achieved, there is still a goal.

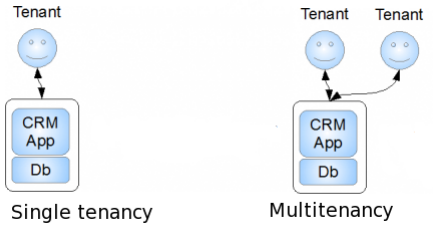

Figure 2: Simple examples of single and multitenancy. CRM app means CRM application or just an instance of an application and db means Database. Tenants are users. [1]

3.1 Multitenancy

Multitenancy is another property that a service oriented architecture must have. This is what a lot of current cloud computing architectures do not have. The reason multitenancy is important is for efficiency.

Terms that need to be known are single tenancy and mutlitenancy [1]. Single tenancy is when a provider has an application running and only one user is using it at a time. An example of a single tenancy program would be a text editor. A user has an application on their computer and only they can run that application. Figure 2 is an example of single tenancy and multitenancy. Multitenancy is when a provider has one instance of a program running on a server and many people connect and use the application instance at the same time. An example of a multitenancy program would be gmail or hotmail.

There are positives and negatives to both of these options. Most of the advantages to single tenancy programs are when the program is being used on a person's personal machine and not when running on the cloud. The major downside to single tenancy programs on the cloud is that they are less efficient because for each person that uses that application there has to be a new instance and this uses a large amount of resources.

The positives of multitenancy is everything is handled for the user by the provider. This includes security of user data, backup of user data, and updates to software and hardware. The downsides are that you have to trust that provider is doing their job. There is never any guarantee that the company providing the service will not go out of business or make a mistake managing the users data.

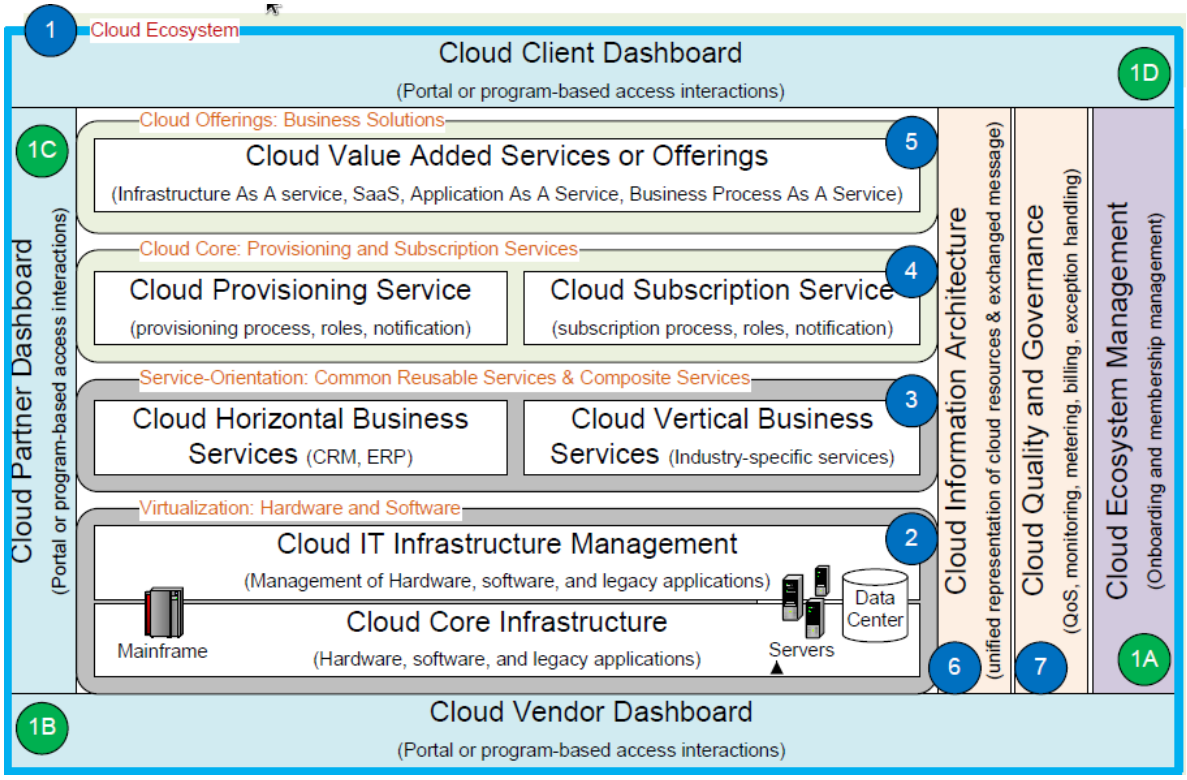

Figure 3: The CCOA architecture. Modules are labeled 1-7 and the name of each section is above each piece of the architecture [8]

4. CLOUD COMPUTING OPEN ARCHITECTURE

With a service oriented model of cloud computing in mind there needs to be some type of architecture behind it. Cloud computing open architecture (CCOA) [8] is an architecture that incorporates many of the concepts from SOA and attempts to make the SOA a reality. In this section CCOA will be discussed at a conceptual level and then how it was implemented in a case study. The case study was a small scale implementation of most of the ideas introduced in this section. The case study was an in house study that was done to show that CCOA can be implemented in small scale and there will be only one provider. Each piece of the case study will be talked about after each principle. CCOA has three goals.

The first goal is to make an architecture that is reusable and scales into the future. This architecture should be able to be upgraded and change with time without having to be replaced. This is the goal that links tightly together with SOA.

The second goal is to propose a unified platform to build the cloud on. CCOA designers hope it will serve as the building blocks for all cloud services providers. This will let the user switch from cloud to cloud without having to start all over or having to scrap a large part of the users project.

The third is to hopefully make businesses run at optimum efficiency. The goal of all businesses is to make money and when businesses run efficiently they make more money. This goal stems from the first two goals being met.

For these three goals to be achieved there are seven principles that CCOA will follow. These seven principles are the basis for the architecture of CCOA. Figure 3 is an illustration of the 7 principles.

4.1 Principle 1

The first principle is Integrated Ecosystem Management (IEM). The IEM is the user interface of CCOA. This part provides an interface with the clients, partners and other vendors of cloud computing resources. The cloud ecosystem has four components. The first is *cloud ecosystem management* (1A). In Figure 3 this is module 1A. Module 1A manages modules 1B, 1C, ad 1D in Figure 3. The second part is the *cloud vendor dashboard* (1B). This is an interface that will allow the provider to see the important information that they need to know, such as front end and back end interaction.

The third part is the *cloud partner dashboard* (1C). It is needed because many vendors and providers of services do not work alone. There are many collaborations of services that combine to make up many different services. The cloud partner dashboard is needed for the vendor to observe and collaborate with their partners.

The fourth and final part is the *cloud client dashboard* (1D). This final part is the part the clients will be using to access the cloud providers and hopefully pick one that they would like to use and switch between cloud providers. From this dashboard they will see things such as providers, services that they provide, pricing and rating of cloud providers. Other values can be added to this list.

When all three dashboards are put together we get the cloud Ecosystem, module 1A. These three pieces combined have everything that a cloud provider needs.This will be all or most of the front end that people will see and interact with.

In the case study preformed this module was handled by web portals that were handled by WebSphere. A web portal or links page is a dynamic collection of links related to a specific topic. An example of a web portal is MSN, Yahoo or AOL. Webshpere is software from IBM that gives you a set of tools that lets you interact and manage web portals. This is used to link the module 1A to the other modules. Module 1A provided user account management. 1C is not used in the case study because there are no partners in the study and thus no need for 1C. If 1C were to be used it would handle the login process and then redirection when logged in. 1B and 1D handle login and authentication of name and passwords and then use Websphere portals to redirect the customer or vendor once logged in. See Figure 4 for an example of the cloud client dashboard.

From the dashboard the users had a choice of Windows or Linux. Then there was multiple different server types that were used. The server types were Xen-VM, xSeries, Xen Cluster, VMware, DynamicP6 and DynamicP. These were all the choice that users could pick from.

4.2 Principle 2

Virtualization is to create a virtual version of an operating system, or storage device on a hardware platform [7]. An example of virtualization is a Virtual Machine (VM). A VM is an instance of an operating system running on a server. There can be multiple different VMs running at the same time on the same server.

The second principle is virtualization for cloud infrastructure. This principle deals with using the proper amount of resources that are needed and is an extension of concepts in SOA. This means that if a company is expecting an increase or decrease of customers they will adjust their resources for the number of people that the company thinks that it will have to deal with. Companies will also have to deal with an increase in resource use. If multiple users start to use more resources, the amount of resources will need to be adjusted. There are two ways of dealing with this concept.

The first way is hardware optimization, adjusting the physical amount of resources that a provider has depending on the amount of physical resources needed at the time. For example if a provider is expecting an increase in users and does not have enough CPU power and storage capacity they would have to adjust by installing more servers to handle this. The goal here is to have a provider optimize their resources.

Another approach to this problems is software virtualization. This relates to the example of virtualization[1] above. The servers in the cloud will have one or more VMs running on each server. How many VMs run on each server depends on the amount of resources each server has and how much computational power each VM is using. For example, suppose server one has 100 percent of its resources open and the provider puts a VM on that server that uses 50 percent of those resources. The provider can then put another VM on that server that uses 50 percent or less total resources of the server.

In Module 2 this is where all the virtualization takes place. The infrastructure of this module is composed of a large bank of servers. These are the physical servers that the cloud runs on. The software on the server consists of Websphere Application server (WAS). WAS is a component of Websphere the software by IBM. WAS is software that allows a provider to manage server-side Java. The other component is the DB2 database. DB2 is a relational model database server that runs on Unix developed by IBM. DB2 handles all database interaction. LDAP is used to handle user information and access. LDAP or Lightweight Directory Access Protocol is an internet protocol that is used to encrypt and store user names and passwords.

4.3 Principle 3

The third principle in CCOA is service orientation for common reusable services. The goal here is to use services that are reusable to make the services more efficient. There are two ways to do this: cloud horizontal and vertical business services. Horizontal is meant to hide the technical side of the cloud. An example of horizontal would be email notification services. The vertical side is all the payment services. Anything that lets you pay the company like paypal or credit card services are included here. This is how users will pay the providers for the services that they are using.

In the case study there are multiple services used for the horizontal business services. These services consist of an order management service, email notification service, resource management service, and work flow management service. All these services are implemented as a type of web service. The vertical business service is not used because there is no use of shipping or payment services.

4.4 Principle 4

The fourth principle is extensible provisioning and subscription for cloud. The cloud provisioning service separates paying customers from non-paying customers. The subscription service handles the user subscriptions.

In the case study these services are handled by a few different services. This section handle paying vs non-paying customers. If the user is a non-paying customer and then they start paying the cloud provisioning service will move their status into the paying customer side.

4.5 Principle 5

The fifth principle is configurable enablement for cloud offerings. The cloud offering is the final product that the cloud provides. This brings us back to the idea of infrastructure as a service (IaaS), software as a service (SaaS), and platform as a service (PaaS). This is the main service that the cloud provider gives to its customers. These services are provided though different mediums. The most common is through an internet browser. As discussed in the cloud computing section, depending on what you are delivering as a product, the way the customer gets their product will vary.

In the case study this module contained server provisioning. This mean that IaaS was provided. In Figure 4 the user chooses what OS they want and how much computational power they need. PaaS and SaaS can be added to this without changing the architecture. They are not implemented in this study.

[1]The authors of the paper [8] calls software virtualization what some people call hardware virtualization.

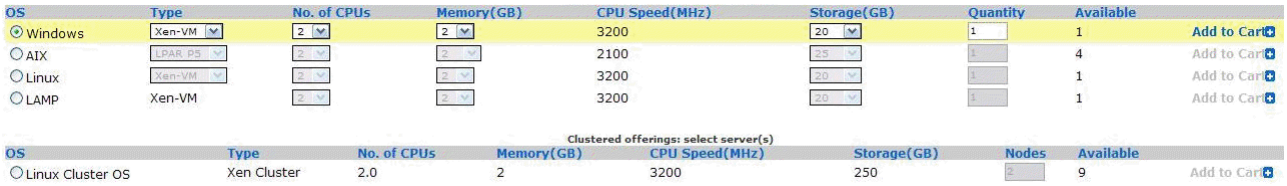

Figure 4: Cloud client dashboard [8]

4.6 Principle 6

The sixth principle is unified information representation and exchange framework. This section is also linked to one of the goals of SOA; the goal of having a unified information framework. All users should be able to interact with all the cloud providers at the same level. There are a few ways to implement this framework. Resource description framework (RDF) is a method of conceptual modelling of information in web pages, web services resource framework (WSRF) is a framework for modelling and accessing stateful resources using the web, and Extensible Markup Language (XML) is a markup language fo encoding documents with structured information. All of these tools can be used to model data and define a base framework that all provider would have to use. If each provider is forced to use the same framework CCOA can achieve a unified information representation and exchange framework.

In the case study the researchers define database schema for orders, contracts, project information, and business scenarios. The data is stored in module 2 in the DB2 database. Then there are schema for users and groups. This is also stored in the DB2 database. A technology called Enterprise Services Bus (ESB) is used to handle all information routing and transformation in the cloud information architecture module. ESB is designed by IBM to interact with WebSphere and provides integration for SOA applications. It is a software architecture model for communication between interacting software.

4.7 Principle 7

The seventh and final principle of CCOA is cloud quality and governance. This is the module that monitors all performance of the cloud and the servers and ensures that the cloud servers are running efficiently. This is the section that takes care of quality and will determine what needs to be changed in the design, deployment, operation, and management of the cloud offerings. This module measures quality of the product the vendor provides and decides if something needs to be changed in the product.

In the case study service-level agreements, contracts, and resource statistics are saved and stored. This module monitors the use of resources being used at the moment and determines if the vendor needs more or less resources. Tivoli Enterprise Portal Server is used to watch over service operations. Then Tivoli Provisioning Manager is used to manage and schedule all provisioning tasks. Provisioning tasks include allocating space on a server for new VMs and making sure there is room on that server. Tivoli Provisioning Manager [4] collects data and then determines the best course of action on how to provision the servers. It builds and formats the portal workspaces that display real-time data collected by monitoring tools. With the data collected by Tivoli people can see if the servers are operating at maximum efficiency and adjust resources, hardware and anything else where needed. Tivoli can provision space by measuring the amount of resources needed. Tivoli then provisions the space on a server that has the required amount of resources and moves that application to the new server.

Along with these provisioning tasks the cloud needs to deal with updates and downtime of servers. All software needs updates and all hardware can fail. How does a cloud provider dealing with these issues in real time, keep the all services running? This can be done with proper scheduling tasks [6]. First the place that the users information is stored and is being accessed from must be changed to a different server. The server that is going down will notify the user that there will be temporary downtime and then transfer the user and their processes to a new server. This is when either software or hardware updates can take place. Ideally these updates can take place when no users are using the services but this is almost never the case.

4.8 CCOA Conclusion

CCOA is a an architecture that claims to a service oriented architecture and does have some of the qualities of one. In the end they do have an architecture that could become a service oriented architecture but their case study is lacking in details to provide proof that what they implemented was a fully implemented service oriented architecture. The authors do have the cloud client dashboard but they do not implement the ability to switch between providers and transfer data between them. CCOA does not support multitenancy and if it does it is not talked about by the researchers within the paper. Multitenancy is one of the most important pieces of a service oriented architecture. Over all CCOA could be a service oriented architecture but because of lack of details in the paper it can not be concluded to be one.

5. SERVICE ORIENTED CLOUD COMPUTING ARCHITECTURE

Service Oriented Cloud Computing Architecture (SOCCA) [5] is the other architecture proposed by researchers. This is a theoretical architecture that is not implemented but discussed in the paper. This architecture is similar in many ways to CCOA and different in others. This is another architecture that attempts to take advantage of the goals and views of SOA. SOCCA has four layers that make up its architecture.

5.1 Cloud Provider Layer

This is the layer at which each individual cloud provider has their own hardware and software. Just like in CCOA there are multiple providers providing services, but unlike

CCOA each provider manages their own software and hardware. In this layer all the cloud providers take care of their own resources and virtualization. Each provider has to figure out how much resources they will need and if they need to cut down on the amount that they are using or increase the amount they have.

5.2 Cloud Ontology Mapping Layer

The cloud ontology mapping layer is the layer of SOCCA that hopes to mask the differences between separate clouds. This layer is to help the transfer of data from one cloud to another which fulfils one of the key goals of SOA. To achieve this goal SOCCA has three ontology systems. An ontology is generic knowledge that can be reused by different applications.

- Storage Ontology: This ontology deals with data manipulation on the cloud. This includes data update, date insert, data delete, and data select and so on.
- Computing Ontology: "It defines the concepts and terms related to distributed computing on the clouds."
- Communication Ontology: "It defines the concepts and terms related Communication Schema among the clouds, such as data encoding schema, message routing."

An example of how these three otologies are used would be the storage ontology would handle data on the cloud. This would be much like the sixth principle in CCOA. The otologies are so that all providers have a framework to build on, but instead of each cloud provider building there cloud on that framework, like in CCOA, each cloud provider shares information in the same way.

5.3 Cloud Broker Layer

This is the layer that deals with information for each cloud provider. Information such as pricing, hardware, software, and services provided. This is the equivalent to the cloud client dashboard of CCOA. The core components of this layer is cloud provider information which is the information above. Ranking is how well each provider is rated by their customers. This involves comparing prices and reliability and reviews from people who have used that cloud.

5.4 SOA Layer

This layer deals with the ideas of SOA and implementing them into the SOCCA architecture. One of the key and most important ideas of SOA is multitenancy. SOCCA hopes to utilize multitenancy to its fullest. There are two types of applications in this architecture. The first is Multiple Application instance; these are single tenancy applications. Examples of this would be, applications that like VMs that are individual operating systems that are single tenancy. The second is single application instance to multiple users. This would be multitenancy applications. Examples these applications would be Gamil.

6. CONCLUSION

This paper discussed these theoretical architectures CCOA and SOCCA. Both of these architectures propose an ideal architecture that hopes to make cloud computing better. CCOA, as talked about above, implemented an architecture, but the paper does not have enough information or detail in the case study to say that this is a service oriented architecture. In the end the architecture proposed could be a service oriented architecture, but there is not enough evidence in the case study to say they succeeded. SOCCA is a promising idea and could possibly be made into a service oriented architecture. The concept of a cloud computing architecture that has all the best properties would be an amazing feat, but is still far away and needs more effort invested into it before a service oriented architecture can become a realization. These two architectures could be a good place to start.

7. REFERENCES

[1] S. Bobrowski. All about multi-tenancy. `http://thecloudview.com/all-about-multi-tenancy-part-1/`, January 2010.

[2] A. Cardenas. Cloud computing overview. `http://www.docstoc.com/docs/85286348/Cloud-Computing-Overview`, July 2011.

[3] W. Cellary and S. Strykowski. e-governament based on cloud computing and service oriented architecture. *ICEGOV, 2009*, pages 5–10, 2009.

[4] IBM. Tivoli provisioning manager. `http://www-01.ibm.com/software/tivoli/products/prov-mgr/`.

[5] W.-T. Tsai, X. Sun, and J. Balasooriya. Service oriented cloud computing architecture. *Inforamtion Technology, New Generations (ITNG), 2010 Seventh International Conference*, pages 684–689, 2010.

[6] W.-T. Tsai, X. Sun, Q. Shao, and J. Elston. Real-time service oriented cloud computing. *2010 6th World Congress*, pages 473–478, 2010.

[7] Wikipedia. Virtualization. `http://en.wikipedia.org/wiki/Virtualization`, 11 2011.

[8] L.-J. Zhang and Q. Zhou. Ccoa: Cloud computing open architecture. *Web Services 2009, ICWS 2009, IEEE International Conference*, pages 607–616, 2009.