2012 International Conference on Computer Technology and Science (ICCTS 2012)
IPCSIT vol. 47 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V47.1

Atom Based Network Architecture for Cloud Governance

Alex Antony Arokiaraj [1]

[1] Ericsson Inc

Abstract. The current world is moving towards cloud offering. Service providers are attracted towards the policy “Own Less Use More”. Services are offered in various modes like infrastructure, memory and even software. Works on tools and technologies for cloud computing, managing bandwidth, access methodologies like creating access points for cloud are progressing, however there are limited works concerning the security in cloud environments. In fact, “Governance” is a term which is rarely talked about along with security, though it plays a key role between two parties having distrust. `Governance` is the term used to refer to the balance between the power and flexibility. There are some common interpretations to this definition. Power and Flexibility are considered to be almost mutually exclusive terms, the consideration of which poses a severe threat to the future of cloud. Let’s say a service provider leases an infrastructure from a technology provider. In a real world scenario, this leasing period will extend for many years. During this period, the service provider assumes the role of a virtual owner, though the infrastructure is maintained by the technology provider. During this period, there are several concerns regarding fidelity, security and privacy. These concerns should be properly addressed to carry the technology for the future. The ultimate solution to these concerns is proper governance. The Atom based Network Architecture fills the distrust between two parties by providing a solution for proper governance in networks. I present the Atom based Network Architecture, and demonstrate the non-trivial policies. A system that is a predecessor of the expected architecture is developed and key architectural enhancements are identified. I conclude by considering the advantages of the architecture.

Keywords: Governance, Security, Networks, Distributed Systems, Cloud Offering, Computer Architecture

1. Requirements and Goals

There are a few demands in the cloud business that leverage this architecture. They are **decentralization of power**- the entire power should not reside with the administrator of the system, if the system is a part of a network. This is very important for Cloud Networks, **governance**- the architecture must allow proper governance between the service provider and the technology provider, **flexibility**- there should be a flexible approach towards enforcing security policies, **uniformity**- there should be uniformity of policies in a network. Policies should not be modified because of the fact that the system is rigid, **efficiency**- the enforcement of policies should not in any way affect the throughput of the system.

The atom based network architecture addresses the key impediments by focussing on two major things. **Monitoring the actions of the administrator and logs to be handled by the service provider**- the service provider will have no control over the contracted policies that the administrator is allowed to perform. The technology provider cannot avoid the transparency of operations to the service provider. **Sequestration of the secondary complexities**- A system inside the network that is meant for a specific purpose does not need to carry the burden of maintaining security policies. This is achieved by removing the processes that are meant for security from the operating system. The inbuilt security mechanisms will be replaced by a security resource group in a network. The security resource group will be floating across the network. Communications to and from the system will be handled by the “floating security resource group” thereby discounting the possibility of vulnerable attacks when the inbuilt mechanisms are removed from the system.

[+] Alex Antony Arokiaraj. Tel.: +(91-9582894572);
*E-mail:*alexantony1988@gmail.com

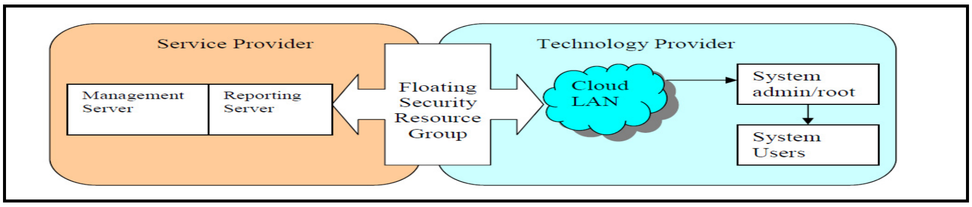

Figure 1: Floating Security Resource Group Model (FSRG)

2. Evolution of the Architecture

In early 2010, Bharti Airtel (World's largest Prepaid Network) faced the tragedy of most of the calls not being billed during the Common- Wealth Games. The managed services were maintained by Ericsson and the technology provider (Ericsson) had to bear the losses. Root Cause Analysis reports showed the faulty handling of the system, but what exactly was done and who did it went obscure. The Security measures adopted and the security policies to be adopted in the future remained dubious. A Solution had to be designed that would properly govern more than 3500 network elements securely, that would maintain the trust between the two organizations.

2.1. Implications of Level Based Access Control

The first approach towards the solution was to have a non-discretionary security policy, *Level Based Access Control*. In a Level Based Access Control, privileges are allotted on the basis of level of work to be carried out. Deeper discussions with the system administrators and further analysis led to the following observations. In a larger enterprise network environment, level based access control is less efficient, since it is obligatory that the highest power of access, commonly called as root access in UNIX based operating systems, had to be distributed among a large number of people. The enforcement of security had to compromise power. This strategy will therefore reduce the operational efficiency. For example, in a larger network operations team, the administrator is not always a single person. The role of administrator is shared between individuals on time basis. In such scenarios, restricting rights would not allow another administrator to maintain the system effectively or troubleshoot in event of failures. If there are no restrictions, then all the administrators have equal rights to accuse each other. The solution has to therefore consider the balance between the exercise of power and the enforcement of security in addition to considering the flexibility of the currently available operating systems.

2.2. Middleware® as a Possible Solution

The solution can be as simple as a middleware or complex as the floating security resource group model as proposed in the Atom based Network Architecture. All the intended users (including the system administrators, system operators and other users) who want to login to a node will have a limited access on the middleware and the middleware will connect them to the respective network elements (irrespective of the OS they run on Linux®, Solaris®, HP-UX®, IBM AIX®) with the highest power of access. The user who wishes to login to any node uses the graphical user interface of the middleware and logs in to his/her account. Each account is associated with a profile that is presented to the user. The user then culls the specified node and type of access needed. As the user selects the root access, handshake happens between the gateway and the desired node on behalf of the user. The handshake is based on public key authentication mechanisms. For operating Systems that have inbuilt public key exchange mechanisms, the property was used directly (Ex. Solaris®).

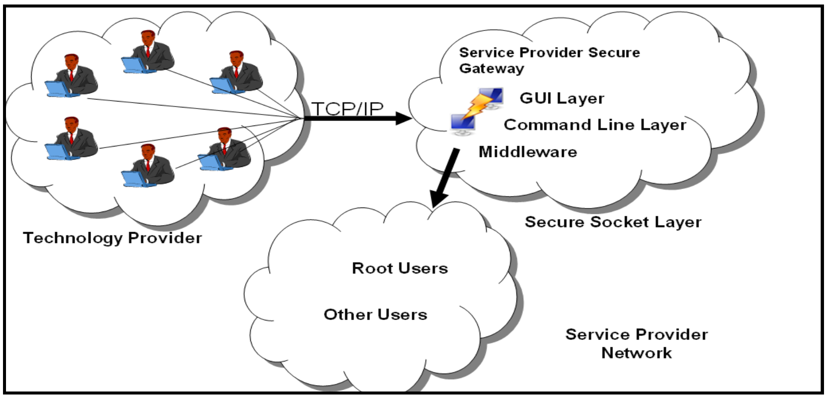

Figure 2: Representation diagram of the implemented middleware

The user is redirected to the node with the root access. The user then uses the terminal, which is given by the gateway, to perform the necessary operations on the node. The logs from the gateway are directed to the reporting server.

The middleware also had two terminal layers for the users in the technology provider's network. The users can choose to use the graphical user interface from which they can login to the nodes in the service provider's network. They also have the option of the using the third party terminals available (like *putty*®). In this case the gateway® processes runs as a daemon providing the same functions as the middleware terminals will provide.

The middleware system that we developed had the following advantages.

- It allowed the distribution of highest access to many administrators involved in maintaining the network, thereby allowing the operational efficiency but also brought in the transparency between the two organizations doing business together.
- The logging of commands executed by all the users in the system was embedded in all the operating systems, but the root users had the authority to delete it as root was considered to be supreme. In our case root users are a part of the technology provider network. The distrust between two parties who owns the nodes and who manages the nodes was eradicated by the implementation of the middleware system.
- The middleware also abducted the processes that were used to create and manage users from the node. It also provided password-less access to nodes with highest access but still maintaining the security level by providing a jailed access to the middleware.

3. Looking Beyond the Middleware®

The deployment of the middleware system satisfied the desideratum to keep the trust between the two parties, but the system had its own disadvantages. The availability of the middleware had to be at its ideal level, because the availability of the middleware determined the availability of the entire network (only from the perspective of Network Operations). The only solution to increase availability is to deploy more nodes in a cluster thereby forming a distributed system. This solution had the following threats. First, the expansion of the network would demand the expansion of the middleware and also the gateways. Second, the expansion of the gateways directly means an increased cost involved for the service providers. Increased investment in security is not carried out by most of the service providers although they excogitate on high security.

4. The Atom Based Network Architecture (ABNarch)

I propose the atom based network architecture that will introduce a model called as the Floating Security Resource Group Model (FSRG). The design of this architecture is impelled by a range of requirements as mentioned in the beginning of this paper and also to profligate the disadvantages posed by the middleware systems acting as gateways.

The users and the system administrators of the technology provider network form the integral part of the 'nucleus'. 'Orbits' epitomize the functionality that the individual system can offer in a network. A 'revolving electron' instantiates a physical system, running on various platforms. I will be using the terms 'electrons' in place of systems and 'orbits' in place of 'functionality'.

The higher the orbit goes, the higher the intended core functionality of the system. For example, the intended core functionality of a router is routing packets. Functionalities like security mechanisms are presumed to be subordinate functionalities. The atom based architecture completely relies on isolating the subordinate functionalities of the system and moving them to the lower orbits of the architecture close to the nucleus. The security related processes and mechanisms are insulated from the operating system and form the FSRG.

Only the members of nucleus can access the electrons of the orbit and in any case the orbits should not be circumvented. Access to the higher orbits should pass through the lower orbits. The higher orbit is owned by the service provider during the leasing period, but administered by the technology provider. The lower orbits are always governed by the service provider. All the systems in the same orbit versify a distributed system. These orbit clusters provides an easy way of capacity expansion for the lower orbits. Since the FSRG is a dissociated unit from the actual system, eventually the memory and processor requirements remain the same. Moreover, a *floating* SRG will reduce the memory and processor requirements, as the FSRG will be shared by the higher orbit elements on time basis.

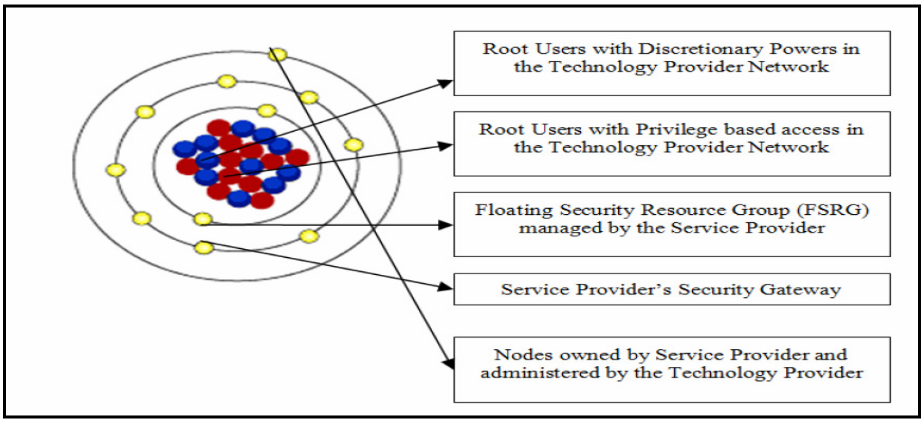
Root Users with Discretionary Powers in the Technology Provider Network
Root Users with Privilege based access in the Technology Provider Network
Floating Security Resource Group (FSRG) managed by the Service Provider
Service Provider's Security Gateway
Nodes owned by Service Provider and administered by the Technology Provider

Figure 3: Orbit Model of the Atom Architecture

4.1. Rule-Set of the Atom Architecture

There are some rule-sets that the architecture must conform to. Assimilating to the rule-sets is a means to ensure that the architecture meets the ultimate goal of maintaining the trust between the two parties engaging in providing and vending cloud services. They also provide a proper framework, so that the basic design of the architecture does not hinder the flexibility of the system.

- Each orbit can have as many systems as possible. But all the systems in a same orbit must form a distributed system.
- FSRG always occupies the orbit that is next to the nucleus. In precise terms, FSRG is always the first orbit (Fig 3). There is only one exception to this. The FSRG may be made virtual in order to reduce the costs. In case the FSRG is made virtual, physically it will be present in the highest orbit. This means that if the service provider is not willing to build a hardware orbit, the orbit can be implemented in the kernel software. This will be among the immutable properties of an operating system. In a generic scenario, where there is only winnowing of security resources, this virtual layer will be in the third orbit. The virtual location of FSRG however, is still assumed to be in first orbit.
- The Gateway Orbit (second orbit as in Fig 3) is considered to have no connection with the semantics of the architecture. The architecture uses the second orbit as a transparent medium.
- The orbits next to the gateway orbit can be subdivided into many orbits. These orbits except the highest orbit will be owned, administered and maintained by the service provider during the leased period. The highest orbit will be owned by the service provider and will be administered by the technology provider. The Sub division of orbits is based on the functionalities of the system.

Segmentation based on other factors is not allowed. (Detailed investigation on the subdivision of orbits will be a part of my future works)

- The higher orbits always contain the intended core functionality. Communication between the nodes in the higher orbits will not be mediated by FSRGs but it will be facilitated by them. This means that system calls, system processes and other processes such as cron jobs will not be mediated by FSRG.
- All the nodes in the higher orbit will respond only to the packet that has passed through lower orbits, especially the orbit in which the FSRG resides.

4.2. Conceptual Changes

The architecture tries to accomplish a simple factor. It rephrases the definition that a network is a group of systems and system in turn is a group of distinct functionalities into the following that a network is a group of distinct functionalities and functionalities in turn is a group of distinct systems. This change is facilitated by the communication protocol suite that follows in the next section.

5. FSRG Communication Protocol Suite

Each FSRG is considered to be floating inside the orbit. This means that there is no permanent binding between a node in the higher orbit and the FSRG. As soon as there is a request for a terminal session, the request reaches the FSRG Engine. The FSRG Engine performs the administration of the FSRGs and hence takes care of allocating a particular FSRG to a terminal session request. The FSRG Engine creates a Free Resource Request (FRR) to check the FPV of a particular FSRG. Each FSRG has a floating point value associated with it. The FPV determines wheather the FSRG is currently floating or it is fixated by a terminal session request. The FSRG once fixated, consults the PEE (Policy Enforcement Engine) for the policy consultation for the specific user who is bounded. The Policy Enforcement Engine is the location where the security policies are implemented by mutual agreement between the two parties. In case of a software implementation of the orbit, the main components of the suite such as FSRG Engine, FSRGs and PEE exists on the systems in the higher orbit and policy synchronization is done between the customer system and virtual FSRG suite on the system itself. Although on a virtual layer implementation, the FSRGs on a system are allowed to handle the requests meant for other nodes in the same orbit, since the same orbit nodes form a distributed system. The Policy Consultation request is followed by a FSRG bind request and the user performs the necessary administration on the node. The FPV of the FSRG is restored once the terminal session is left free.

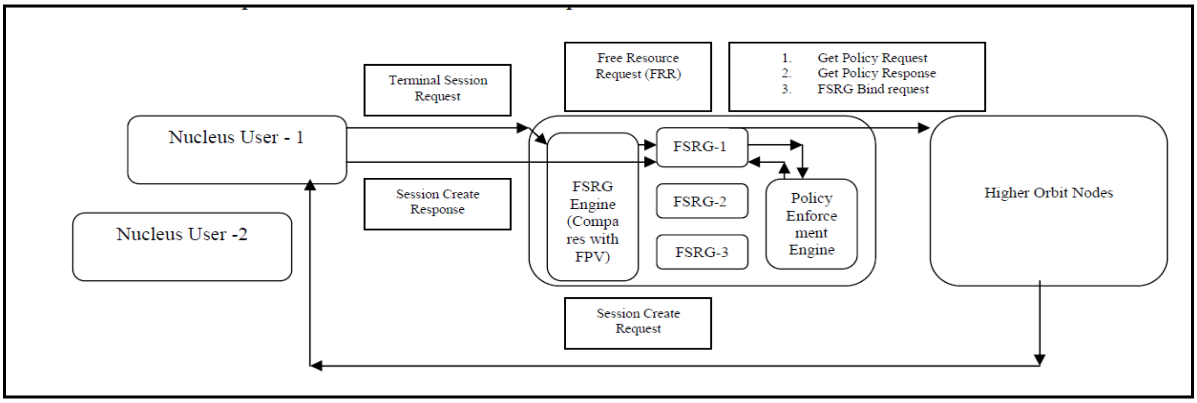

Figure 4: FSRG Communication Protocol Suite

6. Conclusion

I want to quote one of key observations made by Dr. Lorrie Cranor, in one of her publications that “we find that some commonly desired policies cannot be fully enforced with the access control mechanisms that are used to implement them, leading to some cumbersome workarounds”[1] (however we are dealing only with the policies of systems only in a networked environment unlike the publication that deals with both the physical and file-system security). In an enterprise network environment, where large numbers of systems interoperate, coercing a common security policy is onerous and sometimes impossible. Even security policies in multiple versions of a same platform are not alike. The policy enforcement engine and the FSRG communication protocol suite in the atom architecture, enables the enforcement of a common security policy, without affecting the processing throughput of the system, thereby making managed services easier. The architecture itself is particularly useful in cloud environments and thereby provides a proper governance solution for two parties having distrust on each other.

7. Acknowledgements

Abhishek Kumar U, Senior Solution Integrator, Operational Support Systems, Ericsson

8. References

[1] L. Bauer, L. Cranor, R.W. Reeder, M.K. Reiter, and K. Vaniea. Real life challenges in access-control management. In CHI 2009: Conference on Human Factors in Computing Systems, pages 899-908, April 2009.