SETECS
Secure Transactions and Electronic Commerce Systems

Tel: (240) 535–2095 ◆ **http:**//security.setecs.com ◆ **E-mail:** info@setecs.com

Security Architecture for Cloud Computing Environments

White Paper– February 1, 2011

Executive Summary

This document describes the concept, components, and architecture of the SETECS® OneCLOUD™ system. OneCLOUD™ is security system especially designed and structured to provide security services for cloud computing environments.

Cloud computing is becoming very popular computing paradigm for network applications in open distributed environments. In essence, the idea is to host various application servers in a virtual network environment ("cloud") and offer their use through the concept of (Web) and other services. Contrary to classical network applications approach in the form of client–server model, in a cloud environment users do not access individual application servers, do not establish direct connections with them, do not send request messages directly to those servers, and do not receive direct replies from them. Instead, clients access those application servers through cloud access proxies, special servers that perform publishing and exporting various (usually Web) services available in a cloud.

In such environments, security has much more important role than in classical network, client–server, environments. Not only that the same, standard, security services are needed (authentication, authorization, confidentially, integrity, authorization, etc.), but their provision must be offered to clients transparently and in an environment comprising distributed components and delegated authorities. Cloud computing makes security not only much more important, but also much more difficult to organize and manage, due to the transparent nature of cloud resources, components, and services.

This document describes conceptual model of the security architecture for cloud computing environments, components of that architecture, specialized security servers, security clients and security protocols. Implementation of the described architecture and examples of cloud secure applications and protocols are described in another, accompanying white paper.

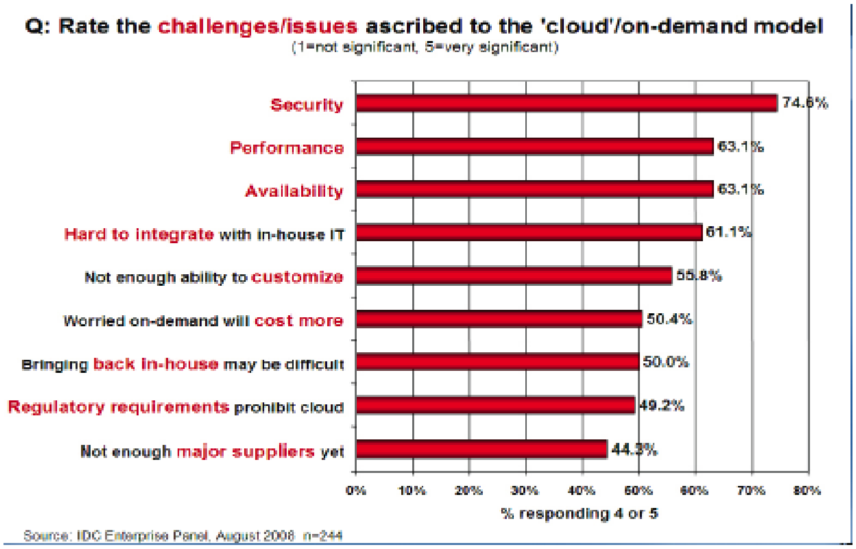
1. Background and Motives

Cloud computing is becoming very popular computing paradigm for network applications. In essence, the idea is to host various application servers in a virtual network environment ("cloud") and offer their use through the concept of (Web) and other services. There are many research papers, development results, commercial products, and even large–scale commercial operational environments offering computing services based on the concept of cloud computing. Commercial companies are offering cloud computing services ([1, 2]), new standards are being developed [3], and even one of the largest users of IT services in the World, US Federal Government, has declared transition from client/server-based to cloud computing services [4].

In this document, we quote one of many definitions and descriptions of such an environment [5]:

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This *cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.*

There are still many open and interesting issues regarding cloud computing paradigm and standards are still evolving. But, it is a general opinion that security is indeed one of the most important issues [6]. In the recent IDC report over 74% of users think that security is dominant issue for widespread use of cloud computing services:



Source: IDC Enterprise Panel, August 2008 n=244

Figure 1: Importance of Security for Cloud Computing Environments

2. Cloud Computing – Functional Architecture

In order to include the components of our security architecture in a concept of a cloud computing environment, in this section we briefly describe functional architecture and basic components of a cloud computing environment, based on three cloud service models from [6]: (a) cloud software as the service (**SaaS**), (b) cloud platform as a service (**PaaS**), and (c) cloud infrastructure as a service (**IaaS**). For the functional description of a cloud computing environment, we start with SaaS, where software is represented by various Application Servers. There may be multiple servers in a cloud, even some of them offering the same types of services. One of the main characteristics of a cloud is transparent access to its services. This means that servers are not accessed by their network address / location (in a client/server model usually their IP number and port or URL address), but by the services they provide. More precisely, clients do not access servers directly and they do not trigger server applications directly through network connections. Users in fact access certain cloud components (request brokers) and those cloud components distribute requests to individual servers, as appropriate.

This approach indicates that, in addition to various application servers, one distinctive service of a cloud is "Service Dispatcher", in this document called **Applications Access Point (AAP) Server.** AAP is service–level dispatcher, i.e. it distributes service requests into a cloud to individual application servers, based on types of requests and other processing parameters. Good analogy is a switch for bankcard payment transactions, distributing payment transactions to various banks in the background.

In order to discover application services available in the cloud, those services must be published and must be discoverable. This is another cloud service in a cloud, in this document called **Services Publishing and Dispatching (SPD) Server**. This server is usually based on the standard concept of the UDDI Server, as specified by OASIS [7], i.e. it is the server used for publishing and discovery of cloud application services. AAP Server queries SPD Server to discover application services, conditions and rules of their access and invocation, and parameters that must be provided in service requests. Access to SDP and AAP servers is usually performed through Web service APIs.

Clients may access cloud services through a variety of communication protocols, as shown in Figure 2. Usually, it used to be Internet and HTTP (Web access) protocol. But, with the recent advances of mobile and wireless technologies and networks the scope of communication protocols is much wider. Clients today may access a cloud using SMS messages, GPRS data channels, Wi–Fi, Bluetooth, RFID and even some proprietary communication protocols. Therefore, in order to be able to accept requests coming through different communication protocols, a cloud needs in front of it and facing various communication networks another service provider. This is communication services provider, in this document called **Communication Access Point (CAP).**

So, simplified functional architecture of a cloud computing environment is shown in Figure 2. It includes front-end CAP, handling alternative communication protocols, after it is AAP, handling different application service requests, and distributing them to appropriate Application Servers located in a cloud and providing various application services. To simplify the Figure, SDP Server is not shown, as it is connected/related to all Application Server and also SPD Server.

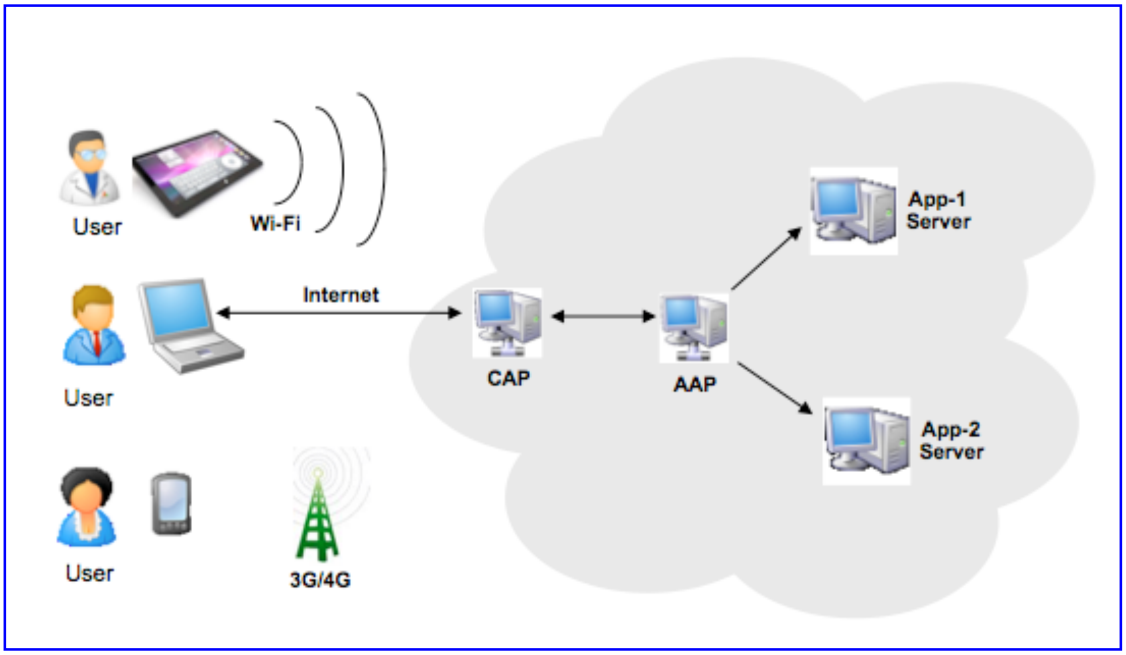


Figure 2: Simplified Functional Architecture for Cloud Computing Environments

3. Cloud Computing – Security Architecture

We designed and structured our security architecture for cloud computing based on the described functional architecture. The approach was to enhance the components of a functional architecture with additional components providing various security services. This is an extension of the SaaS concept, which is suitable for functional description of a cloud, to the new approach where cloud infrastructure is treated as a service – IaaS approach. The idea is to have several security components, which are common to all application servers and their services. Therefore, clients are provided security services by an infrastructure itself, not by individual servers.

As all other components and services in a cloud, security components and services must also be transparent and generic. Transparent means that they are automatically applied, without too much of user intervention, and generic means that they are adjustable to individual users, requirements, applications, and required services. We already have all the results for such approach to security in a distributed environment as the result of our on–going research [8, 9, 10, 11, 12].

Based on those results, we have extended functional components and architecture of a could computing environment, shown in Figure 2, with the following additional security components and services:

3.1 Security Access Point

The first component that is needed as an extension of the functional architecture is **Security Access Point (SAP).** That is cloud server providing front-end security services. The first service, which is important before any access to a cloud is allowed, is *authentication of users*. Authentication must be based on open standards (for interoperability) and without any pre–arrangements (to be applicable in an open environment). We used challenge-response authentication protocol based on the FIPS 196 standard [13]. The standard requires the use of public–key cryptography, therefore for the first, identification message in the FIPS 196 protocol, we use client's certificate. Public–key cryptographic operations on a client side are performed using PIV–compliant smart card, so client certificate is in fact stored in the card. The first component of the SAP is therefore Strong Authentication Server providing strong authentication service.

Certificate is issued to a client and to the SAP server by Certification Authority (CA) server that provides certification services in a cloud. Certificates are stored in an Identity Management System (IDMS) X.500 compliant directory – another server that provides registration and identification services in a cloud. CA and IDMS Servers and their services are described in section 3.2.

In addition to authentication between a client and the SAP server, we also use *single sign–on protocol*. The standard to be followed for this service is Secure Assertion Markup Language (SAML) [14]. Therefore, SAML Server and single sign–on authentication is the next component and single sign–on is the next service provided by the SAP server. SAML ticket is also stored in client's smart card, described in section 3.3.

Finally, after a client has been authenticated and SAML ticket has been issued, the final service before allowing him/her to access any Application Servers is *authorization*. SAP must verify that a client and his/her request are authorized to access internal cloud resources. Authorization is based on the XACML standard [15]. The standard specifies two core components for enforcement of authorization policies: Policy Decision Point (PDP) and Policy Enforcement Point (PEP). Therefore, PEP Server and authorization enforcement is the final security service provided by the SAP server.

3.2 Security Infrastructure Servers

Three security servers are security infrastructure servers already mentioned in the previous section: IDMS Server, PDP Server, and CA Server. IDMS is X-500 compliant server that stores registration data for all local resources of a cloud. Data for users registered in the cloud may also be stored in that server. Alternatively, user registration data may be stored in IDMS servers located in users' home environments. In that case, those servers and security servers in the cloud must be federated. Federation may be accomplished as binding of X.500 directories, as Microsoft forest of domain servers (Active Directories), as distributed database, or using federation protocol for Web services.

Since cloud should also support open access, even by users being registered in other clouds, binding between IDMS Servers located in cooperating clouds must be performed as a prerequisite for federated secure cloud architecture. Besides binding of IDMS servers, in case of multiple clouds, federation must also be established between authorization policies. In a single cloud PDP is the server maintaining XACML authorization policy and performing verification of access requests on behalf of a SAP server that acts as a PEP Server. So, in a federated environment, authorization policies maintained by individual PDP servers must be synchronized. Synchronization is performed using federation protocol and it covers both aspects – syntactic (dictionaries) and semantic (rules) synchronization.

Finally, CA is standard Certificate Authority Server. For the purpose of scaling, CA Server must be linked into a large–scale Public–Key Infrastructure (PKI).

Therefore, security components and architecture for cloud computing environments are shown in the following Figure 3:

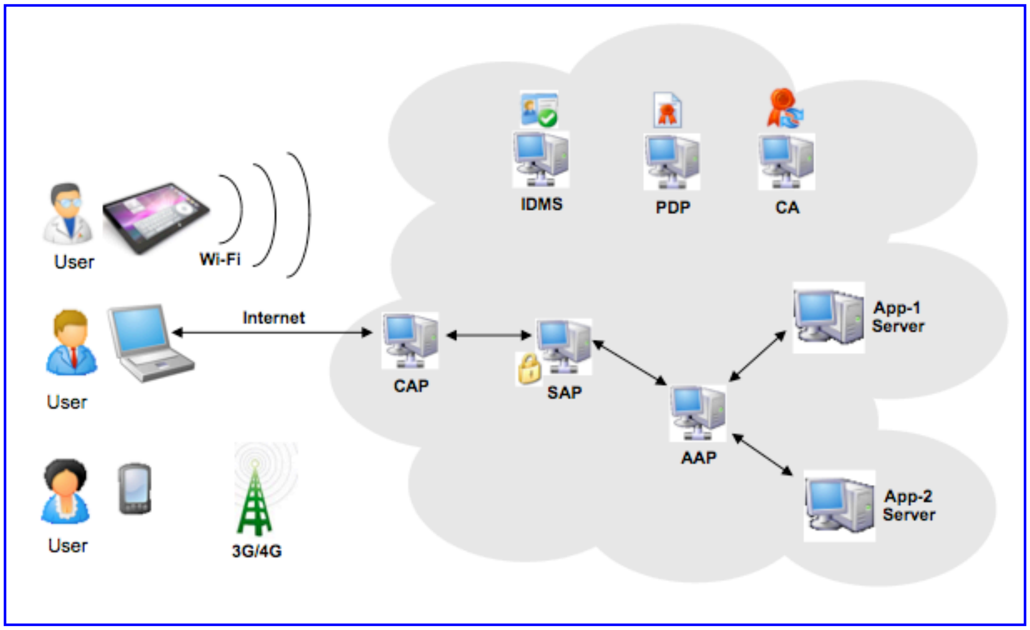


Figure 3: Security Components and Architecture for Cloud Computing Environments

3.3 Secure Client for Cloud Computing

In order to use effectively all security services, as specified in sections 3.1 and 3.2, standard client stations (PCs with a browser) must also be extended with some security components. The most important of them is ability to use smart cards. PIV cards are used for strong authentication to the cloud. Based on the scope of the FIPS 201 standard, PIV–compliant cards can only be used for local and strong remote authentication. PIV cards can not support single–sign on protocol and also can not support authorization services, since SAML token cannot be stored in a PIV applet. Since PIV applet is strictly standardized, it cannot be extended. Therefore, in order to store SAML ticket and some other security parameters, we use an additional – Security applet in our PIV card. Our cards are fully compliant to the FIPS 201 standard, since they contain standard PIV applet and support all standard PIV security services.

Our extended PIV card can be used by two additional security components at a client's workstation – authentication module, supporting singe sign–on authentication protocol and authorization module, supporting authentication protocol, based on SAML and XACML standards.

Client performs strong authentication protocol with the SAP server by sending PIV authentication certificate to the server. That certificate is verified and if valid, further used to verify the identity of a user by consulting IDMS server. Certificate validity is verified either by using verification of signatures and expiration dates for all certificates in a certificate chain or by using On–Line Certificate Status Protocol (OCSP), depending on the configuration of the SAP server. After successful validation of the user's certificate and user's identification data, SAP performs strong authentication (challenge/response) protocol. In our system, all cryptographic responses by the client are performed by user's PIV card.

When strong authentication protocol is successfully completed, SAP server sends SAML authentication request to the PDP server. If user is authorized, PDP server will return SAML authentication response, i.e. SAML ticket. That ticket will be returned to the client. If user uses our extended PIV card, SAML ticket will be stored in the Security applet in the card. With this feature we provide mobility to users, as they can move from one station to another, accessing repetitively the cloud without performing repetitive authentication and authorization.

When a user sends some application service request to the cloud, that request will first reach SAP server. That server will trigger an action by the client located at user's workstation, which will read SAML ticket from the extended PIV card and send it to the SAP server. That server will forward it to the PDP Server for the authentication and authorization decision. If both approved, user's application request will be passed to the appropriate application server, where it will be served, and the response will be returned to the user. All described security actions: request for the SAML ticket, sending it to the SAP server, then forwarding it to the PDP server, receiving reply back to the SAP server and, finally access to the application server providing the requested service, are performed instantaneously and transparently to the user. Therefore, the user is not aware of any of these actions, except if some unauthorized action is attempted.

4. Conclusions and Distinguished Features

The described security system for cloud computing environments is very advanced and extremely important today. It combines all popular security technologies and provides all standardized security services. It very efficiently utilizes standard and extended PIV cards for several security services. The system is very easy to install, configure and activate. Its administrative actions are well documented, easy to understand and perform.

For developers, the system provides rich set of Web service APIs and other programing interfaces, so it is easy to enhance various applications in a cloud with its security services. The system is fully compliant to all Internet security standards, so its certification and validation is simplified. Its components are modular, so each of them can be replaced with equivalent functional component, based on the same standard.

One of the most important features of all components and products comprising the described system is that all software modules, components, libraries and local files are encrypted. This means that the system has internal protection and therefore, it is not vulnerable to attacks by hackers, viruses, worms, malware or to any other software problems.

Finally, SETECS® owns the complete source code of all the described components and products, so each of them can be modified and customized, if required by customers and users.

5. References

[1] Amazon.com, http://aws.amazon.com/

[2] IBM, http://www.ibm.com/ibm/cloud/

[3] NIST, http://www.nist.gov/itl/cloud/index.cfm

[4] US Federal Government, http://www.informationweek.com/

[5] Mell, P., Grance, T., "*The NIST Definition of Cloud Computing*", Version 15, 10-7-09, http://csrc.nist.gov/groups/SNS/cloud-computing/

[6] Mell, P., Grance, T., "*Effectively and Securely Using Cloud Computing Paradigm*", Workshop presentation, Oct 7, 2009, http://csrc.nist.gov/groups/SNS/cloud-computing/

[7] Bellwood, T., et al.: "*UDDI Version 3.0, UDDI Spec Technical Committee Specification*" [Specification], July 2002, web: http://uddi.org/pubs/uddi-v3.00-published-20020719.htm

[8] Abdul Ghafoor, Sead Muftic, and Gernot Schmölzer, "*CryptoNET: A Model of Generic Security Provider*"', published in the International Journal of Internet Technology and Secured Transactions, Vol. 2, Nos. 3/4, pp.321–335, 2010

[9] Abdul Ghafoor, Sead Muftic "*CryptoNET: Software Protection and Secure Execution Environment*", published in the International Journal of Computer Science and Network Security (IJCSNS), pp. 19-26, Vol 10, February, 2010

[10] Abdul Ghafoor, Sead Muftic, "*CryptoNET: Security Management Protocols*", published in the Proceeding of The 9th WSEAS International Conference on Data Networks, Communication, Computers (DNCOCO-2010), Faro, Portugal, pp. 15-20, November, 2010

[11] Muftic, S., Zhang, F., DeZoysa, K., "*SAFE System: Secure Applications for Financial Environments using Mobile Phones*", Proceedings of the IADIS International Conference e–Society, Barcelona, Spain, 2009

[12] Zhang, F., Muftic, S., Schmölzer, G. "*Secure Service-Oriented Architecture for Mobile Transactions*",

Proceedings of the World Congress on Internet Security (WorldCIS-2011), February 21-23, 2011, London, UK

[13] NIST, “*Entity Authentication Using Public Key Cryptography*”**,** http://csrc.nist.gov/publications/PubsFIPS.html

[14] OASIS, “*Security Assertion Markup Language (SAML*)”, http://saml.xml.org/about-saml

[15] OASIS, “eXtensible Access Control Markup Language (XACML)”, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml