Chapter 16
Co-Existance of High Assurance and Cloud Based Computing

William R. Simpson and Coimbatore Chandersekaran

Abstract Cloud computing is emerging as an attractive, cost effective computing paradigm. However, many of the applications require high assurance, attribution and formal access control processes including defense, banking, credit, content distribution, etc. Current implementations of cloud services do not meet high assurance requirements. The high assurance requirement presents many challenges to normal computing and some rather precise requirements that have developed from high assurance issues for web service applications. The challenges of high assurance associated with cloud computing are primarily in five areas. The first is virtualization and the loss of attribution that accompanies a highly virtualized environment. The second is the loss of ability to perform end-to-end communications. The third is the extent to which encryption is needed and the need for a comprehensive key management process for public key infrastructure, as well as session and other cryptologic keys. The fourth is in monitoring and logging for attribution, compliance and data forensics. The fifth is in cloud content storage. We explore each of these challenges and discuss how they may be able to be overcome. Our view of high assurance and the issues associated with web services is shaped by our work with DoD and the Air Force, but applies to a broader range of applications, including content delivery and rights management.

Keywords Attribution · Authentication · Cloud computing · Content management · High assurance · Hypervisor · IT security · Virtualization

W. R. Simpson (✉) · C. Chandersekaran
Institute for Defense Analyses, 4850 Mark Center Drive,
Alexandria, VA 22311, USA
e-mail: rsimpson@ida.org

C. Chandersekaran
e-mail: cchander@ida.org

H. K. Kim et al. (eds.), *IAENG Transactions on Engineering Technologies*,
Lecture Notes in Electrical Engineering 170, DOI: 10.1007/978-94-007-4786-9_16,

16.1 Introduction

This paper is based in part on a paper published in WCECS [1]. Cloud computing has come to mean many different things. To some, it is simply putting one's data on a remote server. However, in this paper, we utilize the definition provided by NIST [2]. They define five essential characteristics of any cloud computing environment:

1. On demand self-service,
2. Broad network access,
3. Resource pooling,
4. Rapid elasticity, and
5. Measured service.

It is important to note that multi-tenancy and virtualization are not essential characteristics for cloud computing. For our discussion we will assume no multi-tenancy, which adds the largest element of security complication.

Arguments below do not require either. Cloud computing is, at its core, a *service*. There are three primary models of this service. In the lowest level Infrastructure as a Service (IaaS), storage, computation, and networking are provided by the cloud provider to the cloud consumer. In the next level up of Platform as a Service (PaaS), all of the trappings of IaaS plus an operating system and perhaps some application programming interfaces (APIs) are provided and managed by the cloud provider. The highest service model is Software as a Service (SaaS), in which the cloud provider provides an end-user service such as webmail. The higher the service model, the more control the cloud provider has as compared to the cloud consumer.

There are four different models for deploying cloud services. Primarily, they are public or private clouds. In a public cloud, the infrastructure—although generally not the data on it—may be used by anyone willing to agree to its terms of use. Public clouds exist off the premises of the cloud consumer. Private cloud infrastructure is used only by one organization. It may exist either on or off the organization's premises. There are two twists to these infrastructures. In a community cloud, a group of organizations with similar interests or needs share a cloud infrastructure. That infrastructure is not open to the general public. The community may adopt a single security approach and the same security mechanisms or it may not. Community clouds are best formed in this manner. In this form the shared cloud is similar to an enterprise with restricted sharing. In the latter there is a restricted form of multi-tenancy which may lead to security issues unless low assurance satisfies the basic requirement. In a hybrid cloud, two or more cloud deployment models are connected in a way that allows data or services to move between them. An example of this would be an organization's private cloud that makes use of a community cloud during loads of high utilization.

16.2 Cloud Computing

Cloud computing benefits emerge from economies of scale [2]. Large cloud environments with multiple users are better able to balance heavy loads, since it is unlikely that a large proportion of cloud consumers will have simultaneously high utilization needs. The cloud environment can therefore run at a higher overall utilization, resulting in better cost effectiveness. In a large cloud computing environment, rather than having a number of information technology generalists, the staff has the ability to specialize and become the masters of their own domains. In many cloud environments this balancing is done by virtualization and the use of a hypervisor. With regard to information security, the staff can become even more specialized and spend more time hardening platforms to secure them from attacks. In the homogeneous cloud environment, patches can be rolled out quickly to the nearly identical hosts.

16.2.1 Drawbacks of the Cloud

Cloud computing is not without its drawbacks. In cases where services are outsourced, there is a degree of loss of control. This can affect compliance with laws, regulations, and organizational policies. Cloud systems have additional levels of complexity to handle intra-cloud communications, scalability, data abstraction, and more. To be available to cloud consumers, cloud providers may need to make their services available via the Internet. And critically, many clouds use multi-tenancy, in which multiple organizations simultaneously utilize a single host and virtualization. If one tenant organization is compromised or malicious, it may be able to compromise the data or applications of the other organizations on the same host. The load balancing may use a single identity for all instances of a service whether it is virtual or real.

16.2.2 Differences from Traditional Data Centers

Cloud computing relies on much of the same technical infrastructure (e.g., routers, switches, operating systems, databases, web servers) as traditional data centers and as a result, many of the security issues are similar in the two environments. The notable exception in some cases is the addition of a hypervisor for managing virtual machines. The Cloud Security Alliance's security guidance states "Security controls in cloud computing are, for the most part, no different than security controls in any IT environment. Cloud computing is about gracefully losing control while maintaining accountability even if the operational responsibility falls upon one or more third parties." While many of the controls are similar, there are two factors at work that make cloud computing different: perimeter removal and trust. With cloud computing, the concept of a network or information perimeter changes radically. Data and applications flow from cloud to cloud via gateways along the cloud perimeters. However, since the data may be stored in clouds outside the organization's premises or control, perimeter controls become less useful. In exchange for the lack of a single perimeter around one's data and applications, cloud consumers must be able to trust their cloud providers. A lack of trust in a cloud provider does not necessarily imply a lack of security in the provider's service. A cloud provider may be acceptably secure, but the novelty of cloud computing means that many providers have not had the opportunity to satisfactorily demonstrate their security in a way that earns the trust of cloud consumers. Trust must be managed through detailed Service Level Agreements (SLAs), with clear metrics and monitoring mechanisms, and clear delineation of security mechanisms [3].

Cloud computing benefits emerge from economies of scale [4]. Large cloud environments with multiple users are better able to balance heavy loads, since it is unlikely that a large proportion of cloud consumers will have simultaneously high utilization needs. The cloud environment can therefore run at a higher overall utilization, resulting in better cost effectiveness. In a large cloud computing environment, rather than having a number of information technology generalists, the staff has the ability to specialize and become the masters of their own domains. In many cloud environments this balancing is done by virtualization and the use of a hypervisor. With regard to information security, the staff can become even more specialized and spend more time hardening platforms to secure them from attacks. In the homogeneous cloud environment, patches can be rolled out quickly to the nearly identical hosts.

16.2.3 Some Changes in the Threat Scenario

There are clear differences in many of the threat scenarios as detailed below [2]:

1. Loss of governance (or visibility and/or control of the governance process),
2. Lock-in (threats may be present and locked into the cloud environment),
3. Isolation failure (e.g., hypervisor attack, lack of accountability),
4. Compliance risks (if provider cannot provide compliance evidence or will not permit audit by customer, lack of accountability),
5. Management interface compromise (and or inheritance of threats and/or malicious code from other users of the cloud),
6. Data protection (how does customer verify protection, lack of accountability),
7. Insecure or incomplete data deletion,
8. Malicious insider (often the cloud insider is not vetted as well as the organizational insider, and insiders from other customers could bring in contagious viruses—see 5 above).

16.3 High Assurance Computing

While the current implementations of Cloud Computing provide efficient and operationally friendly solutions to data computing and content distribution, they are not up to the challenge of high assurance. In certain enterprises, the network is continually under attack. Examples might be; banking industry enterprise such as a clearing house for electronic transactions; defense industry applications; credit card consolidation processes that handle sensitive data; both fiscal and personal, medical with concerns for privacy and statutory requirements; content distributor's worried about rights in data, or theft of content.

The attacks have been pervasive and continue to the point that nefarious code may be present, even when regular monitoring and system sweeps clean up readily apparent malware. Despite this attack environment, the web interface is the best way to provide access to many of its users. One way to continue operating in this environment is to not only know and vet your users, but also your software and devices. Today we regularly construct seamless encrypted communications between machines through SSL or other TLS. These do not cover the "last mile" between the machine and the user (or service) on one end, and the machine and the service on the other end. This last mile is particularly important when we assume that malware may exist on either machine, opening the transactions to exploits for eaves dropping, ex-filtration, session high-jacking, data corruption, man-in-the-middle, masquerade, blocking or termination of service, and other nefarious behavior. Before we examine the challenges of Cloud Computing systems, let us first examine what high assurance architecture might look like.

16.3.1 Architectural Features

In order to build an architecture that conforms to these tenets, there must be elements that insure that they are built into the systems. In the architecture we espouse, the basic formulation is based on web services and uses Organization for the Advancement of Structured Information Standards (OASIS) standards of security [5].

16.3.1.1 Naming and Identity

Identity will be established by the requesting agency. To avoid collision with the names, the identity used by all federated exchanges shall be the name as it appears on the primary credential provided by the certificate authority. The name must be unique over time and space which means that retired names are not reused and ambiguities are eliminated. Naming must be applied to all active entities (persons, machines, and software).

16.3.1.2 Credentials

Credentials are an integral part of the federation schema. Each identity (all active entities) requiring access shall be credentialed by a trusted credentialing authority.

16.3.1.3 Bi-Lateral End-to-End Authentication

The requestor will not only authenticate to the service (not the server), but the service will authenticate to the requestor. This two way authentication avoids a number of threat vulnerabilities.

16.4 Challenges in Bringing the Cloud and High Assurance Together

Despite the obvious advantages of cloud computing, the large amount of virtualization and redirection poses a number of problems for high assurance. In order to understand this, let's examine a security flow in a high assurance system (Fig. 16.1).

The application system consists of a web application (for communication with the user), one or more aggregation services that invoke one or more exposure services and combines their information for return to the web application and the user, The exposure services retrieve information from one or more Authoritative Data Sources (ADSs).

Once the authentication is completed, an SSL connection is established between the requestor and the service provider, within which a WS-Security package will be sent to the service. The WS-Security [5] package contains a SAML token generated by the Security Token Server (STS) in the requestor domain. The primary method of authentication will be through the use of public keys in the X.509 certificate, which can then be used to set up encrypted communications (either by X.509 keys or a generated session key). Session keys and certificate keys need to be robust and sufficiently protected to prevent malware exploitation. The preferred method of communication is secure messaging using WS Security, contained in SOAP envelopes. The encryption key used is the public key of the target (or a mutually derived session key), ensuring only the target can interpret the communication.

The problem of scale-up and performance is the issue that makes cloud environments and virtualization so attractive. The cloud will bring on assets as needed and retire them as needed. Let us first examine scale-up in the unclouded secure environment. We will show only the web application, although the same rules apply to all of the communication links between any active elements shown in the Fig. 16.2. The simplest form of dividing the load is to stand up multiple

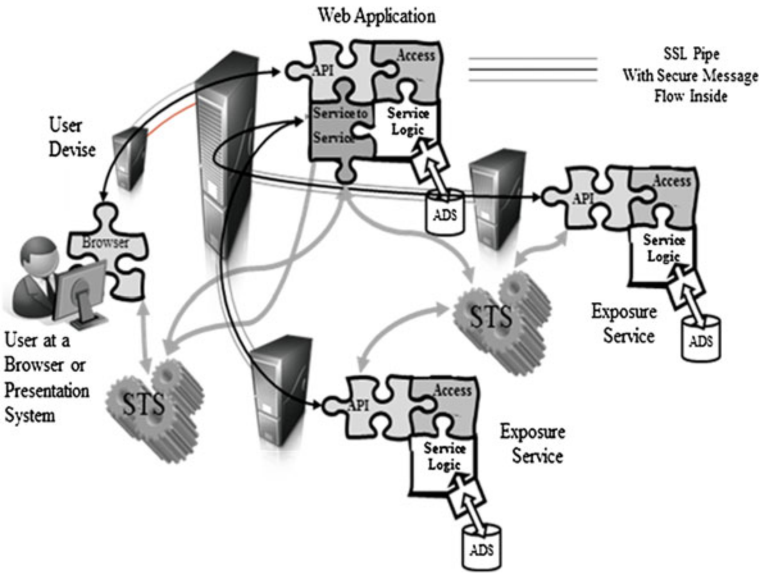

Fig. 16.1 High assurance security flows

independent instances and divide users into groups who will use the various instances. Dependent instances that extend the thread capabilities of the server are considered single independent instances. Remember, all independent instances are uniquely named and credentialed and provisioned in the attribute stores. A representation that is closer to the cloud environment is shown in Fig. 16.3.

A traffic cop (load balancer) monitors activity and posts a connection to an available instance. In this case all works out since the new instance has a unique name, end-point, and credentials with which to proceed. All of this, of course needs to be logged in a standard form and parameters passed to make it easy to reconstruct for forensics. We have shown a couple of threats that need mitigation where one eavesdrops on the communication and may actually try to insert himself into the conversation (man-in-the-middle). This highlights the importance of bi-lateral authentication and encrypted communications. The second is present on instance 4 and highlights the need to protect caches and memory spaces.

When a cloud environment runs out of resources for computing, it builds additional instances, some of these may be thread extension schemas, and some may be independent instances. The traffic cop here is often called a hypervisor and it keeps track of the instances and connections. Figure 16.3 shows notionally how this operation works. When thread capacity is saturated at the server, the hypervisor would nominally redirect the request to an independent virtual or real instance of the web application. If none exists, it will build one from elements in the resource pool as depicted in instance 4 on the chart. If the last user signs off of an independent virtual or real instance (instance 3 in the Fig. 16.3), the hypervisor

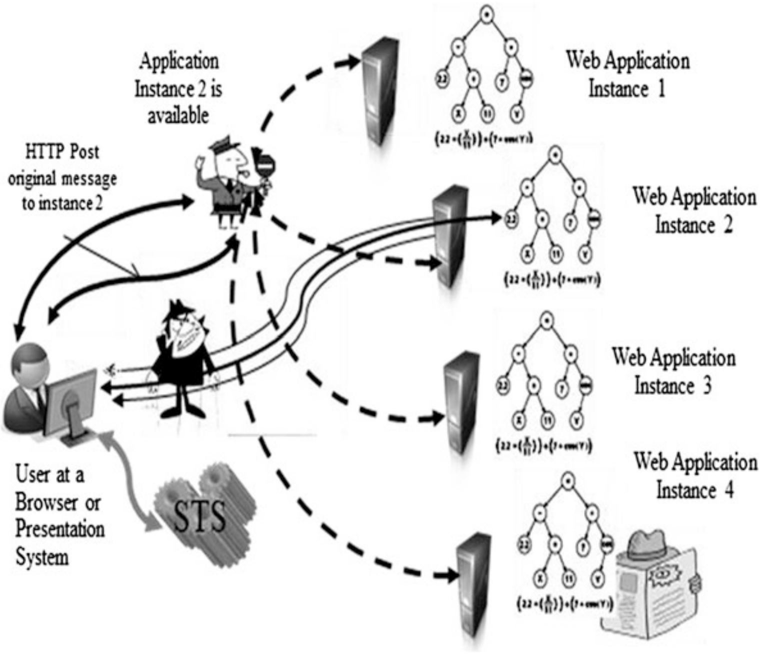
Application Instance 2 is available
HTTP Post original message to instance 2
Web Application Instance 1
Web Application Instance 2
Web Application Instance 3
Web Application Instance 4
User at a Browser or Presentation System
STS

Fig. 16.2 High assurance load balancing

tears down the instance and places the resources back into the resource pool. This provides an efficient re-allocation of resources.

There are several steps that must be taken to preserve the security, if we are interested in a high assurance computing environment. The number of independent instances must be anticipated. Names, credentials and end points must be assigned for their use. The attribute stores and HSMs must be provisioned with properties and key to be used. The simple re-direct must be changed to a re-post loop as in Fig. 16.3. The user will then have a credentialed application to authenticate with bi-laterally and an end point for end-to-end message encryption. Key management is complex and essential. When a new independent instance is required it must be built, and activated (credentials and properties in the attribute store, as well as end point assignment). All of these activities must be logged in a standard format with reference values that make it easy to reassemble the chain of events for forensics. When a current independent instance is retired, it must be disassembled, and de-activated (credentials and properties in the attribute store, as well as end point assignment).

All of these activities must be logged in a standard format with reference values that make it easy to reassemble the chain of events for forensics. The same threats exist, and the same safeguards must be taken. In fact, in Fig. 16.3 nefarious code is built right into the virtual or real instance 4, which underscores the need for trusted and verified software to do the virtualization, and protection of the resources while they are in the resource pool.

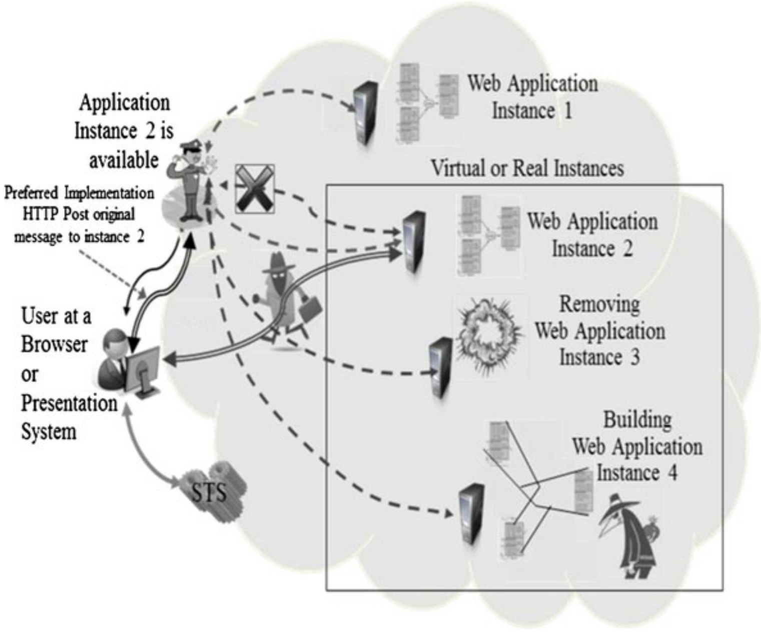

Fig. 16.3 High assurance virtualized hypervisor activity

A recap of these challenges is listed below:

1. Shared Identities and credentials break the accountability paradigm.

- Each independent instance of a virtual or real machine or virtual or real service must be uniquely named [6, 7] and provided a PKI Certificate for authentication. The Certificate must be activated while the virtual machine is in being, and de-activated when it is not, preventing hijacking of the certificate by nefarious activities. Each instance of an independent virtual or real machine or virtual or real service must have a unique end point. This means that simple re-direct will not work. Extensions of the thread mechanism by assigning resources to the operating system may preserve this functionality.

2. Multi-tenancy (multiple tenants using a single host) must be prohibited.
3. No virtualization across machines (each virtual machine must reside in a single real machine).
4. Each potential independent instance of a service must have an account provisioned with appropriate elements in an attribute store. This is required for SAML token issuance.
5. A cloud based Security Token Service (STS) needs to be installed and implemented and it must meet all of the requirements listed her for uniqueness of names and end points as well as instantiated certificates and cryptographic capability.

6. The importance of cryptography cannot be overstated, and all internal communications as well as external communications should be encrypted to the end point of the communication. Memory and storage should also be encrypted to prevent theft of cached data and security parameters.
7. Private keys must reside in Hardware Storage Modules (HSMs).
   a. Stand-up of an independent virtual or real machine or virtual or real service must link keys in HSM, and activate credentials pre-assigned to the virtual service.
   b. Stand-down of an independent virtual or real machine or virtual or real service must de-link keys in HSM, and de-activate credentials pre-assigned to the virtual service.
   c. Key Management in the virtual environment is a particular concern and a complete management schema including destruction of session keys must be developed.
8. Proxies and re-directs break the end-to-end paradigm. When end points must change, a re-posting of communication is the preferred method.
9. Resource pools must be protected from persistent malicious code.
10. All activities must be logged in a standard format with reference values that make it easy to reassemble the chain of events for forensics.

The aforementioned challenges are daunting, but provisions must be made if high assurance computing environments are take advantage of the cloud computing environment.

16.5 Content in the Cloud

In the high assurance enterprise, content stored in general cloud areas can only be protected in one of two ways. The first is total isolation and restrictive gateways for access, which is contrary to the cloud computing paradigm. The second is in encrypted form when unauthorized access is an issue. The latter implies a rights management system of some type.

16.5.1 The Rights Management Function

The Rights management is a collective concept that includes the automated and manual processes to accomplish the steps below:

a. The information asset must be labeled for access and distribution; this is done by the author. Defaults may be assigned absent author input.
b. The information asset must be signed by the author for content integrity (additional signatures may be affixed for authority).

c. Generation of associated metadata for search and discovery.
d. Assignment of an identity (name)—defaulted by the system but can be changed by the user.
e. Author assignment of the actual storage location on the network and filing of the cross-reference between the location and the identity of the asset. The location may be physical or logical.
f. Presentation of a rights information request page (defaulted to read/write/delete rights to the creator and read/delete rights to all others and signature. If additional rights are required (e.g., interest group, special-access group), these are specified at this time.
g. Examination of the access control labels and where an information asset is restricted and not available to all (internal/external), encryption of the information asset and the attachment of an appliqué to the information asset which is used to communicate to the Rights Manager for access control. If the information asset is not access control labeled and is available to all internal and external, the information asset is not encrypted. As a consequence, both encrypted and unencrypted assets may be further distributed without consequence.

To access an information asset, the appliqué attached to the content program for the information asset examines the information asset and if it is encrypted communicates to the Rights Manager via a secure web session to verify claims.

16.5.2 The Components of a Stored Information Asset

The components of a stored information asset are provided in Fig. 16.4 and must be created in steps as described below:

Formatted Document Section a. Information Labeled

Provided by the rights management software with defaults based upon user interest group memberships or by user from approved list, also includes "draft", "final", or Approved as previously described.

Formatted Document Section b. Information Asset Signature(s)

The author's signature (and others) are added and further changes to the information asset at this point are prohibited.

External Information c. MDE Metacard

The Meta Data Environment (MDE) Metacard is prepared. This involves a number of items described below: It should be noted that most information assets are not directly retrievable and it must be retrieved by the content retrieval service for checking of ACLs, MAC issues and restricted authorities. The exception is unclassified, unlimited distribution.

- ***Access Control Labels***

These are taken directly from the trusted labeling of the information asset.

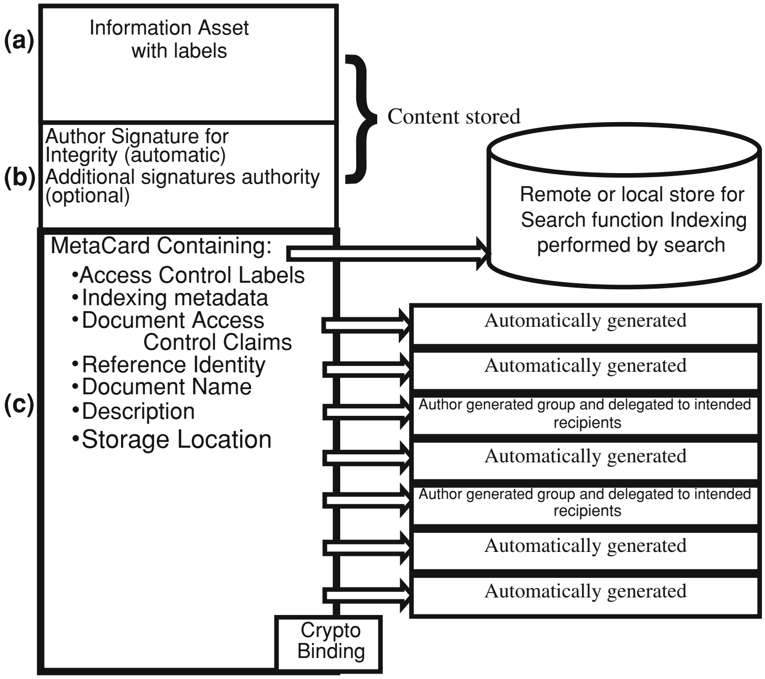
(a)
Information Asset with labels
(b)
Author Signature for Integrity (automatic) Additional signatures authority (optional)
Content stored
Remote or local store for Search function Indexing performed by search
(c)
MetaCard Containing:
•Access Control Labels
•Indexing metadata
•Document Access Control Claims
•Reference Identity
•Document Name
•Description
•Storage Location
Automatically generated
Automatically generated
Author generated group and delegated to intended recipients
Automatically generated
Author generated group and delegated to intended recipients
Automatically generated
Automatically generated
Crypto Binding

Fig. 16.4 Authoritative content information asset format

- ***Key Word MetaData***

The key words are developed from full information asset text scan and/or can be manually entered.

- ***ACL Lists and Associated Data***

The primary ACL is provided by the author (example, “MyGroup”). Once the label is chosen the rights manager inserts this as a delegatable claim of the author.

- ***Reference Identity and Information asset Description***

This is the mechanism for retrieval. The rights manager software defines the identity to prevent duplication, ambiguity or confusion in the information asset file keeping system.

- ***Information asset Name***

The rights management software will provide a default name. It may be modified by the author.

- ***Information asset Description***

The rights management software will suggest a description based upon a title or lead heading. It may be modified by the author.

- ***Storage Location(s)***

This is the actual storage location of the information asset in network asset store. Each time an unmodified copy of the information asset is stored in a different location, the appliqué provides that location and the unique id of the information asset to the rights manager for updating the metacard. The metacard may contain any number of storage locations. This latter allows cleanup when archiving old content.

16.6 Summary

We have reviewed the basic approaches to clouds and their potentials for savings in computing environments. We have also discussed at least one high assurance architecture and its' requirements which provide direct challenges to the way cloud computing environments are organized. Notably the extensive use of virtualization and re-direction is severe enough that many customers who need high assurance have moved away from the concept of cloud computing [8, 9]. Content storage in high assurance cloud environments is also a concern requiring some tools and processes. We believe, however, that a precise statement of the high assurance requirements will lend themselves to solutions in the cloud computing environment, and expand the potentials use of this technology. These concepts are part of a more comprehensive enterprise architecture for high assurance that is web-service based and driven by commercial standards. Portions of this architecture are described in references [10–14].

References

1. Simpson WR, Chandersekaran C (2011) High assurance challenges for cloud computing. In: Proceedings of the world congress on engineering and computer science 2011, Lecture notes in engineering and computer science, vol I. San Francisco, Oct 2011, pp 61–66
2. Jansen W, Grance T (2011) NIST SP 800-144 Draft: guidelines on security and privacy in public cloud computing, security division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8930, Jan 2011. http://csrc.nist.gov/publications/drafts/800-144/Draft-SP-800-144_cloud-computing.pdf
3. Mell P, Grance T (2011) NIST SP 800-145 Draft: cloud computing, computer security division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8930, Jan 2011. http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf
4. Cloud Security Alliance (2009) Security guidance for critical areas of focus in cloud computing V2.1, Dec 2009, https://cloudsecurityalliance.org/csaguide.pdf
5. OASIS Identity Federation (2011) Liberty alliance project, Available at http://projectliberty.org/resources/specifications.php. Accessed 19 Feb 2011

6. OASIS profiles for the OASIS security assertion markup language (SAML) V2.0. Available at http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security. Accessed 19 Feb 2011
7. Standard for Naming Active Entities on DoD IT Networks, Version 3.5, Sept 23, 2010
8. Remarks-Debra Chrapaty, Corporate Vice President, Global Foundation Services, Microsoft Mgt Summit, Las Vegas, May 2008. http://www.microsoft.com/Presspass/exec/debrac/mms2008.mspx. Accessed 19 Feb 2011
9. Plesser A (2008) Executive producer, Beet.tv, cloud computing is hyped and overblown, Forrester's Frank Gillett.Big Tech Companies have "Cloud Envy". http://www.beet.tv/2008/09/cloud-computing.html, Sept 26, 2008. Accessed 19 Feb 2011
10. Catteddu D, Hogben G, European Network Information Security Agency (ENISA) (2009) Cloud computing risk assessment, Nov 2009. http://www.enisa.europa.eu/act/rm/files/deliverables/cloud-computing-risk-assessment
11. Simpson WR, Chandersekaran C, Trice A (2008) A persona-based framework for flexible delegation and least privilege. In: Electronic digest of the 2008 system and software technology conference, Las Vegas, Nevada, May 2008
12. Simpson WR, Chandersekaran C, Trice A (2008) Cross-domain solutions in an era of information sharing. In: The 1st international multi-conference on engineering and technological innovation (IMET 2008), vol I. Orlando, FL, pp 313–318
13. Simpson WR, Chandersekaran C (2009) Information sharing and federation. In: The 2nd international multi-Conference on engineering and technological innovation (IMETI 2009), vol I. Orlando, FL, pp 300–305
14. Chandersekaran C, Simpson WR (2010) A SAML framework for delegation, attribution and least privilege. In: The 3rd international multi-Conference on engineering and technological innovation (IMETI 2010), vol 2. Orlando, FL, pp 303–308