

Cloud Architecture Reference Models

Cloud Computing is a new term for a long-held dream of computing as a utility, which focuses on the delivery of scalable IT resources over the Internet, as opposed to hosting and operating those resources locally. The NIST definition of cloud computing includes five essential characteristics, three **service models** (IaaS/PaaS/SaaS), and four **deployment models** (public cloud/private cloud/community cloud/hybrid cloud). In this research, we focus on the **cloud architecture reference models**.

A **Cloud Architecture** [11][12] is the structure of a cloud solution which uses Internet-accessible on-demand services. The cloud architecture comprises on-premise and cloud resources, services, middleware, and software components, geo-location, the externally visible properties of those, and the relationships between them. The term also defines the structure of cloud services and cloud components, management, security, operation and monitoring. The documentation is also necessary for a system's cloud computing architecture. Documenting facilitates communication between stakeholders, documents early decisions about high-level design, and allows reuse of design components and patterns between projects.

A **Cloud Architecture Reference Model** [13] is an abstraction of cloud computing concepts and relationships that can be used to educate organizations, and create standards and guidelines to help organizations apply these concepts. Groups such as Distributed Management Task Force, Cloud Security Alliance and Open Security Architecture are developing cloud reference models to help organizations make informed decisions regarding if and how to adopt these cloud technologies. Cloud providers (e.g. IBM, Cisco, etc.) and federal agencies (e.g. GSA, etc.) are also working on their own cloud reference models which meet their specific application requirements.

Cloud service providers, end users and corporate IT departments are all expecting a cloud architecture reference model. The reference model is of great interest to service providers because it likely represents the next great wave of innovation sweeping across the Internet and presents tremendous business opportunities for those who can successfully define and implement the new paradigm. Cloud end users are interested because services are reasonably priced and can be accessed from any browser giving access to the computing environment from any location and making collaboration much easier. Cloud service developers are interested because the model reduces capital investments, removes constraints on power and space, may deliver much faster development and implementation times, and promises to simplify the management of complex environments.

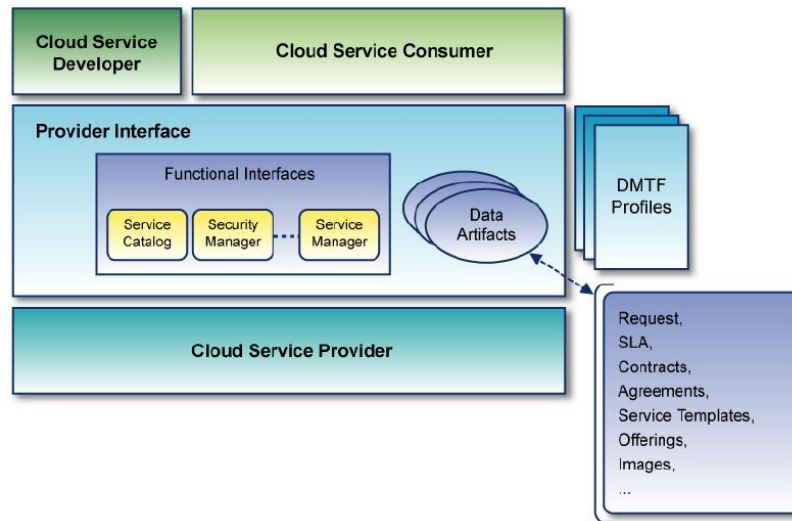
In this report, we survey the cloud reference architecture models proposed by known cloud organization, cloud providers and federal agencies, which include:

- General reference models:
 - o Cloud Computing Use Case Discussion Group
 - o Distributed Management Task Force (DMTF)
 - o Cloud Security Alliance
 - o IBM Cloud Reference Architecture
 - o GSA: FCCI (Federal Cloud Computing Initiative)
 - o Cisco Cloud Reference Architecture Framework
- Reference models focusing on specific application requirements:

- Open Security Architecture: Secure Architecture Models
- SNIA standard: Cloud Data Management Interface
- Elastra: A Cloud Technology Reference Model for Enterprise Clouds

Finally, we conclude the report with a brief comparison among the presented models.

1. Distributed Management Task Force (DMTF): Cloud Service Reference Architecture [2]



DMTF's cloud Service Reference Architecture describes key components — such as actors, interfaces, data artifacts, and profiles — and the interrelationships among these components.

1.1 Actors

The architecture has three primary actors: Cloud Service Provider, Cloud Service Consumer, and Cloud Service Developer. An organization may simultaneously play the roles of any combination of these actors.

- The Cloud Service Provider makes services available to Cloud Service Consumers at agreed service levels and costs. The services may be of any type or complexity. The provider manages the technical infrastructure required for providing the services and provides billing and other reports to consumers.
- The Cloud Service Consumer represents an organization or individuals who contract for services with Cloud Service Providers and then use those services. The Cloud Service Consumer could be another cloud who is a provider to other consumers. The consumer is responsible for selecting the appropriate services, arranging payment for the services, and performing the administration necessary to use those services, such as managing user identities.
- The Cloud Service Developer designs and implements the components of a service. The developer describes the service in a service template. The developer interacts with the Cloud Service Provider to deploy the service components, based on the description in the templates, which the provider may customize before making them available as service offerings.

1.2 Interfaces and Data Artifacts

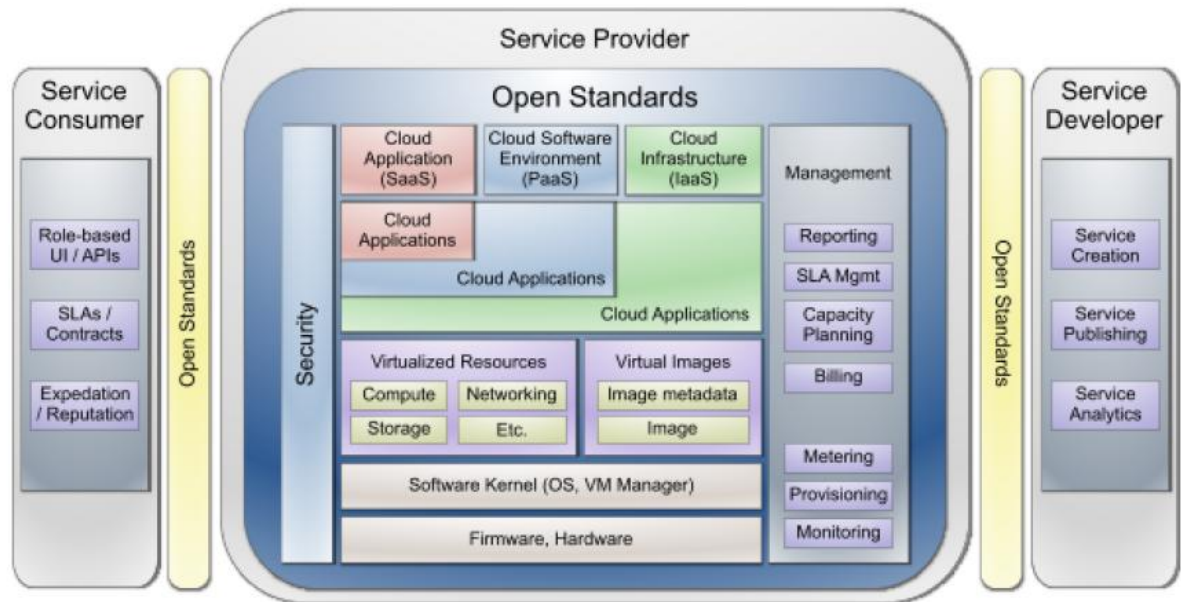
A provider interface defines how the developer and consumer interact with the provider. This architecture differentiates between service endpoints that accept (and respond to) messages over a protocol based on some message exchange pattern (functional interfaces) and the data elements and operations that an interface can support (data artifacts). The interface comprises both functional interfaces and data artifacts.

- Functional interfaces, such as Service Catalogs and Service Managers, are programming interfaces (for example, APIs). Through these interfaces, developers and consumers interact with providers to request, deploy, administer, and use services. Examples of likely functional interfaces are:
 - A Service Catalog, through which service offerings are offered, requested, and managed
 - A Security Manager, through which the security-related aspects of a cloud are managed
 - A Service Manager, through which instances of deployed services are managed and modified
- Data artifacts are exchanged over the functional interfaces. In this context, a data artifact definition describes the semantic content and the specific format (for example, the XML schema definition that describes the XML payload). Examples of data artifact types include service requests, service level-agreements (SLAs) and other contracts, service templates, service offerings, and images that contain applications. For example, a customizable contract template that includes the customer request, SLA, and security requirements is needed to support the service catalog interface. SLA, security requirements, and resource specifications are used to build offerings.

1.3 DMTF Profiles

DMTF profiles are normative specializations or extensions of the interfaces and artifacts, or combinations of them, which are useful in addressing certain contexts, such as those of interest to a security manager or a contract billing administrator. Profiles may be used to simplify the interactions and the potentially complex definitions and negotiation needed to request, manage, and use services. A profile may also specify the use of particular standards that are useful in the profile's target environment and use cases. A profile represents a view into the provider interface.

2. Cloud Computing Use Case Discussion Group: a taxonomy for cloud computing [1]



In this diagram, Service Consumers use the services provided through the cloud, Service Providers manage the cloud infrastructure and Service Developers create the services themselves.

2.1 Service Consumer

The service consumer is the end user or enterprise that actually uses the service, whether it is Software, Platform or Infrastructure as a Service.

Depending on the type of service and their role, the consumer works with different user interfaces and programming interfaces. Some user interfaces look like any other application; the consumer does not need to know about cloud computing as they use the application. Other user interfaces provide administrative functions such as starting and stopping virtual machines or managing cloud storage. Consumers writing application code use different programming interfaces depending on the application they are writing.

Consumers work with SLAs and contracts as well. Typically these are negotiated via human intervention between the consumer and the provider. The expectations of the consumer and the reputation of the provider are a key part of those negotiations.

2.2 Service Provider

The service provider delivers the service to the consumer. The actual task of the provider varies depending on the type of service:

- For Software as a Service, the provider installs, manages and maintains the software. The provider does not necessarily own the physical infrastructure in which the software is

running. Regardless, the consumer does not have access to the infrastructure; they can access only the application.

- For Platform as a Service, the provider manages the cloud infrastructure for the platform, typically a framework for a particular type of application. The consumer's application cannot access the infrastructure underneath the platform.
- For Infrastructure as a Service, the provider maintains the storage, database, message queue or other middleware, or the hosting environment for virtual machines. The consumer uses that service as if it were a disk drive, database, message queue, or machine, but they cannot access the infrastructure that hosts it.

In the service provider diagram, the lowest layer of the stack is the firmware and hardware on which everything else is based. Above that is the software kernel, either the operating system or virtual machine manager that hosts the infrastructure beneath the cloud. The virtualized resources and images include the basic cloud computing services such as processing power, storage and middleware. The virtual images controlled by the VM manager include both the images themselves and the metadata required to manage them.

Crucial to the service provider's operations is the management layer. At a low level, management requires metering to determine who uses the services and to what extent, provisioning to determine how resources are allocated to consumers, and monitoring to track the status of the system and its resources.

At a higher level, management involves billing to recover costs, capacity planning to ensure that consumer demands will be met, SLA management to ensure that the terms of service agreed to by the provider and consumer are adhered to, and reporting for administrators.

Security applies to all aspects of the service provider's operations. (The many levels of security requirements are beyond the scope of this paper.) Open standards apply to the provider's operations as well. A well-rounded set of standards simplify operations within the provider and interoperability with other providers.

2.3 Service Developer

The service developer creates, publishes and monitors the cloud service. These are typically "line-of-business" applications that are delivered directly to end users via the SaaS model. Applications written at the IaaS and PaaS levels will subsequently be used by SaaS developers and cloud providers.

Development environments for service creation vary. If developers are creating a SaaS application, they are most likely writing code for an environment hosted by a cloud provider. In this case, publishing the service is deploying it to the cloud provider's infrastructure.

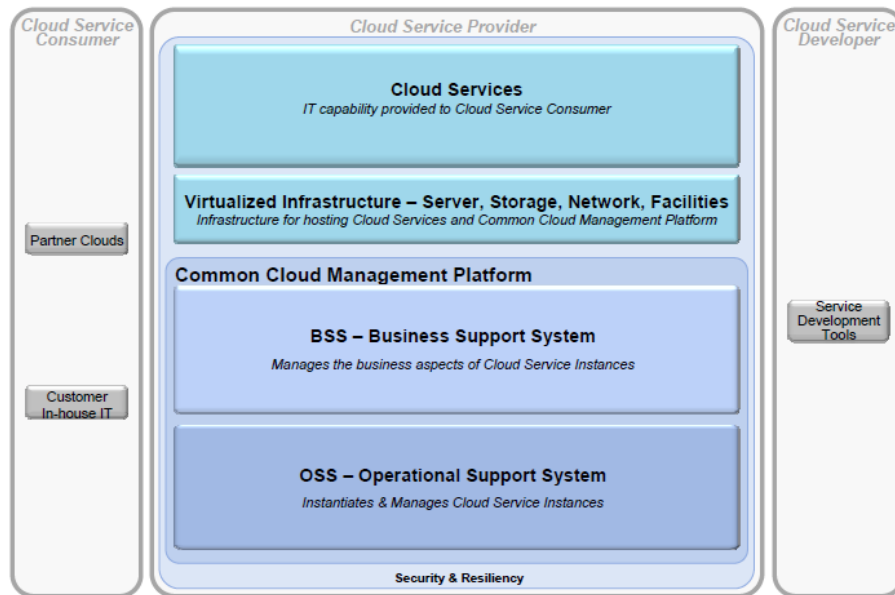
During service creation, analytics involve remote debugging to test the service before it is published to consumers. Once the service is published, analytics allow developers to monitor the performance of their service and make changes as necessary.

3. IBM Cloud Reference Architecture [14]~[20]

3.1 Overview

The next picture shows a typical cloud management architecture overview. It consists of three major building blocks based on the high-level roles that are involved in cloud computing: Cloud Service Consumer, Cloud Service Provider, and Cloud Service Developer.

IBM Cloud Reference Architecture



3.1.1 Cloud service consumer

Cloud Service Consumer includes every user and system that uses or consumes resources of the cloud.

The functions and services offered by cloud computing start with the needs of the consumer, who makes a request for services and resources through a Self-Service portal. Cloud applications then search for resources to match the request using a portfolio of cloud services. Access is provided back to the consumer through the portal.

3.1.2 Cloud service provider

The Cloud Service Provider offers cloud, supplies Infrastructure and a Common Cloud Management Platform. The Common Cloud Management Platform (CCMP) consists of two major building blocks:

- The Business Support Services (BSS) consists of the business-relevant platform services.
- The Operational Support Services (OSS) represents the more infrastructure and operational aspects of the Common Cloud Management Platform.

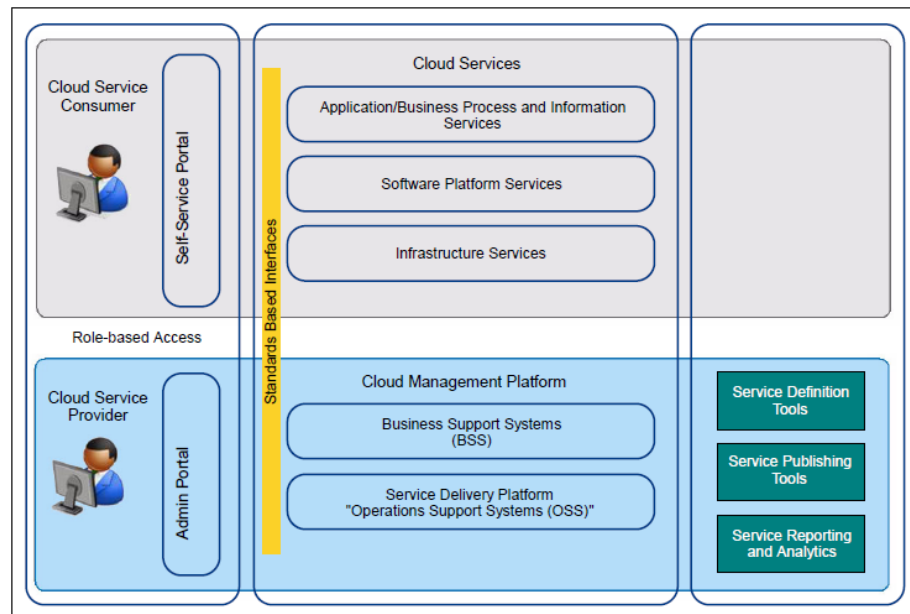
The Cloud service provider role includes the following:

- Service operations managers: Manage technical infrastructure required for providing cloud services.
- Service business managers: Offer all types of services created by the service developer.
- Service transition managers: Responsible for enabling a client to use the cloud service, including on-boarding, integration, and process adoption.

The Cloud service provider performs the following tasks:

- Build services by (optionally) consuming services provided by other service providers.
- Offer services based on a management infrastructure.
- Host services created by other service creators (on top of their own services).

Cloud services are made available to the consumer by the administrator. The administrator uses the Admin Portal to publish the services that are offered as part of the cloud, as well as to prepare resources for use within the cloud. This portal is the interface to a number of tools and reports as well, as shown in next figure.



3.1.2.1 Admin portal

The Admin portal provides administrators with the capability to add, manage, support, and administer private test cloud services and resources in order to fulfill consumer requests. The functions of the Admin portal are to:

- Manage growth, outages, changes, and other life cycle aspects of the cloud.
- Automate actions in the cloud based on monitoring metrics and threshold measurements.
- Control the process workflow to be documented, record approvals, and measure key performance indicators for SLA adherence.

3.1.3 Cloud service developers

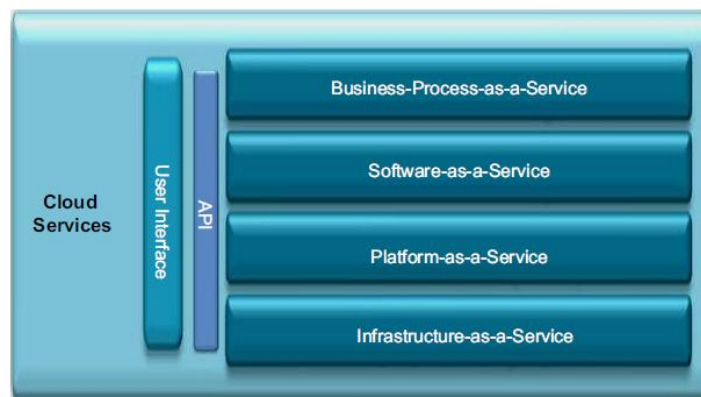
A Cloud Service Developer uses Service Development Tools to develop new cloud services, which includes both the development of runtime artifacts (for example, database persistence, transactional handling, and so on) and management-related aspects (for example, monitoring, metering, provisioning, and so on). In this context, the service development tools support the cloud service developer in creating a service template and a service offering, whereas the service template defines how the Common Cloud Management Platform (CCMP) OSS functionality is used in the context of the respective cloud service and the service offering specifies how the CCMP BSS functionality is used in the context of the respective cloud service.

In the context of a particular infrastructure or platform as-a-service offering, there might also be tooling to develop artifacts that are specific to the particular cloud service.

So in summary, there are two categories of service development tooling: tooling to develop a cloud service by itself and tooling to develop artifacts that are specific to the respective cloud service.

3.2 Cloud services

Cloud services represent any type of (IT) capability that is provided by the Cloud Service Provider to Cloud Service Consumers. Typical categories of cloud services are infrastructure, platform, software, or Business Process Services. In contrast to traditional (IT) services, cloud services have attributes associated with cloud computing, such as a pay-per-use model, self-service usage, flexible scaling, and shared underlying IT resources.



3.2.1 Business process as a service

Business process services are focused on providing existing business processes through a cloud. If there is an existing process with steps that are known, it can be provided as a service within the catalog. This allows the service provider to automate any steps within the process while leaving the changes transparent to the customer.

3.2.2 Software as a service

Software services allow a consumer to select a specific software instance that they want created, without the need to be aware of where and how it will be hosted. For example, a developer can request a new database instance without having to be aware of what OS or hardware the database

will run on. This allows the consumer to focus on the characteristics of the application and gives the provider the freedom to fulfill the request with any resources that will meet the need.

3.2.3 Platform as a service

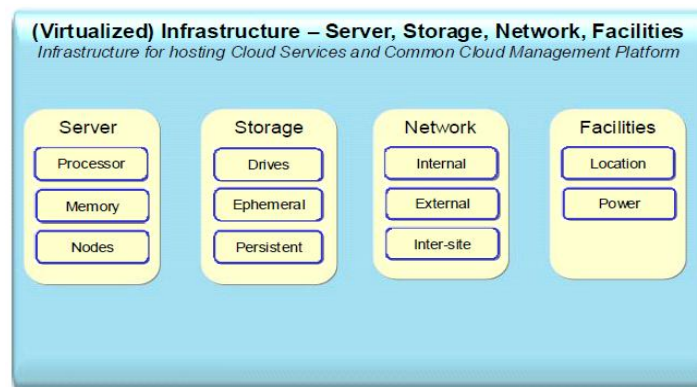
Platform service offerings include workflow facilities for design, development, testing, deployment, and hosting, as well as services that enable team collaboration, Web service integration and marshalling, database integration, security, scalability, storage, persistence, state management, application versioning, application instrumentation, and developer community facilitation. These services are provisioned as an integrated solution over the Web.

3.2.4 Infrastructure services

Infrastructure services allow for the provisioning of standardized compute resources. They allow a consumer to request and receive a new computer instance without needing to focus on IT concerns such as network placement and hardware availability.

3.3 Virtualized infrastructure

The virtualized infrastructure includes all infrastructure elements needed on the cloud service provider side to provide cloud services, which includes facilities, server, storage, and network resources and how these resources are wired up, placed within a data center, and so on. In case of virtualization, this also includes virtualization information, such as hypervisors. It does not include any virtualization management software because that is part of the virtualization management component of the “Operational Support Services”.



3.4 Cloud management platform

The cloud management platform is the set of tools and capabilities that provide services to the consumer. Most of the previous items were about services that were consumed. Cloud management is about providing resources to the system so that services can be consumed.

The Common Cloud Management Platform (CCMP) contains a set of business and operational management focused services that must be used by Cloud Services to actually be a cloud service. The CCMP is responsible for:

- Delivering instances of Cloud Services of any category to Cloud Service Consumers
- The ongoing management of all Cloud service instances from a provider perspective
- Allowing Cloud Service Consumers to manage their Cloud Service instances in a self-service fashion

The CCMP is split into two main elements: Operational Support Services, and Business Support Services.

3.4.1 Business support systems

The Business support systems (BSS) represents the set of business-related services exposed by the CCMP dealing with clients, supporting processes such as taking orders, processing bills, and collecting payments. It includes the components used to run business operations that are client-facing.

The BSS provides services that either enable the cloud service provider or facilitates certain task to deliver the cloud from a business perspective. It contains the services offering management, customer management, pricing and rating, order management, entitlement management, subscriber management, general accounting, invoicing, billing, peering and settlement, contract and agreement management, opportunity to order, metering, analytics and reporting, and service offer catalog.



Besides the business aspect, there is also the technical side of cloud computing because it includes a fast provisioning of standardized IT products and services. In the reference architecture, this area is called Operational Support Services.

3.4.2 Operational Support Services

OSS represents the set of operational management and technical-related services exposed by the CCMP, which must be exploited by Cloud Service Developers to take advantage of the common cloud management platform.

The OSS contains the following services: service delivery catalog, service template, service automation management, service request management, change and configuration management, image life cycle management, provisioning, incident and problem management, IT service level management, monitoring and event management, IT asset and license management, capacity and performance management, and virtualization management.

OSS

Operational
Support
Services

Service Delivery Catalog

Service Automation Management

Service Templates

Change & Configuration
Management

Service Request Management

Image Lifecycle Management

Provisioning

Incident & Problem
Management

IT Service Level Management

Monitoring &
Event Management

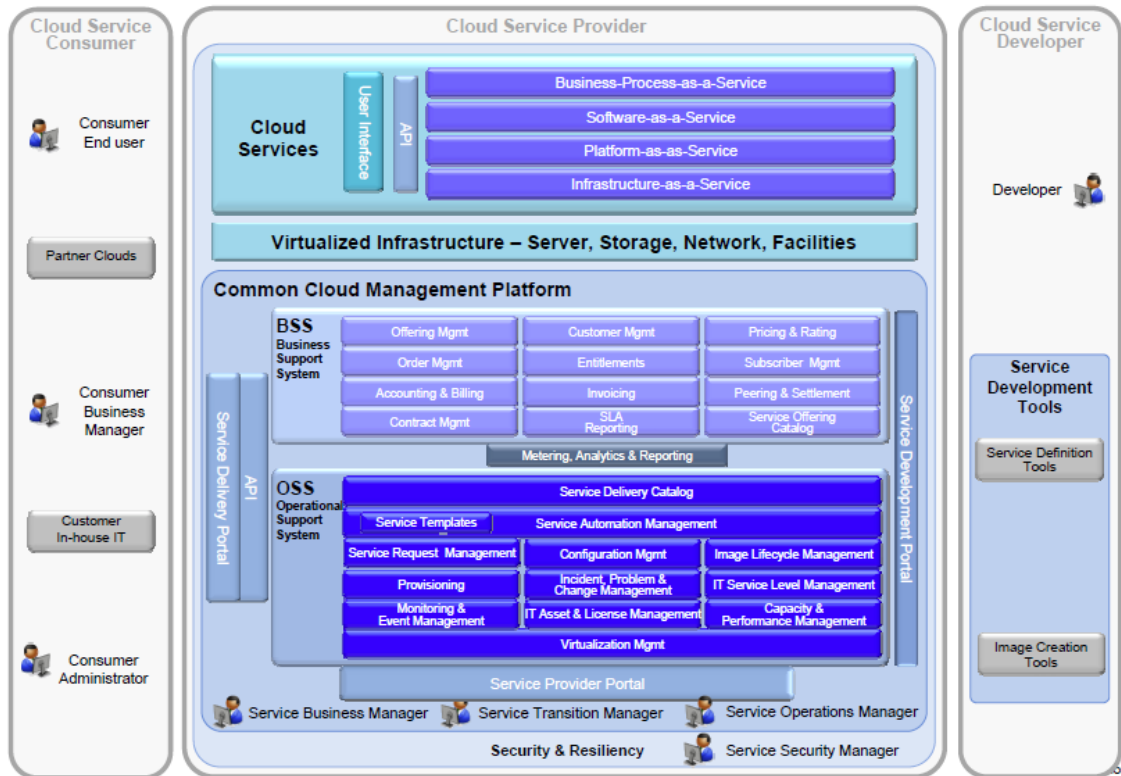
IT Asset & License Management

Virtualization Mgmt

Capacity &
Performance Management

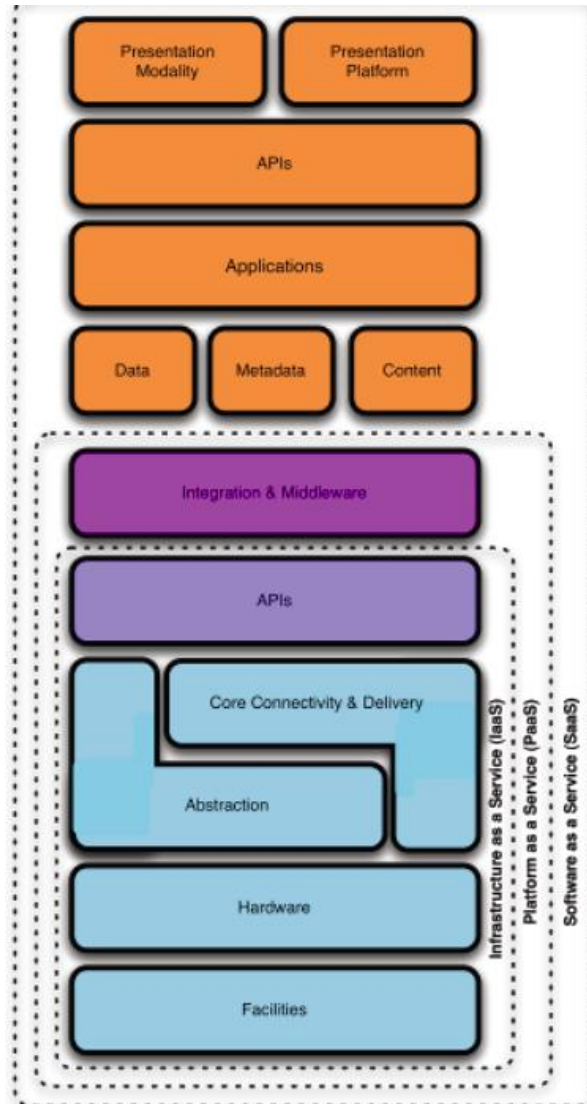
In summary, IBM cloud reference architecture is:

IBM's Common Cloud Model Platform (CCMP)



4. Cloud Security Alliance: Cloud Reference Model [4]

Understanding the relationships and dependencies between Cloud Computing models is critical to understanding Cloud Computing security risks. IaaS is the foundation of all cloud services, with PaaS building upon IaaS, and SaaS in turn building upon PaaS as described in the Cloud Reference Model diagram. In this way, just as capabilities are inherited, so are information security issues and risk. It is important to note that commercial cloud providers may not neatly fit into the layered service models. Nevertheless, the reference model is important for relating real-world services to an architectural framework and understanding the resources and services requiring security analysis.



IaaS includes the entire infrastructure resource stack from the facilities to the hardware platforms that reside in them. It incorporates the capability to abstract resources (or not), as well as deliver physical and logical connectivity to those resources. Ultimately, IaaS provides a set of APIs which allow management and other forms of interaction with the infrastructure by consumers.

PaaS sits atop IaaS and adds an additional layer of integration with application development frameworks; middleware capabilities; and functions such as database, messaging, and queuing; which allow developers to build applications upon to the platform; and whose programming languages and tools are supported by the stack.

SaaS in turn is built upon the underlying IaaS and PaaS stacks; and provides a self-contained operating environment used to deliver the entire user experience including the content, its presentation, the application(s), and management capabilities.

It should therefore be clear that there are significant trade-offs to each model in terms of integrated features, complexity vs. openness (extensibility), and security. Trade-offs between the three cloud deployment models include:

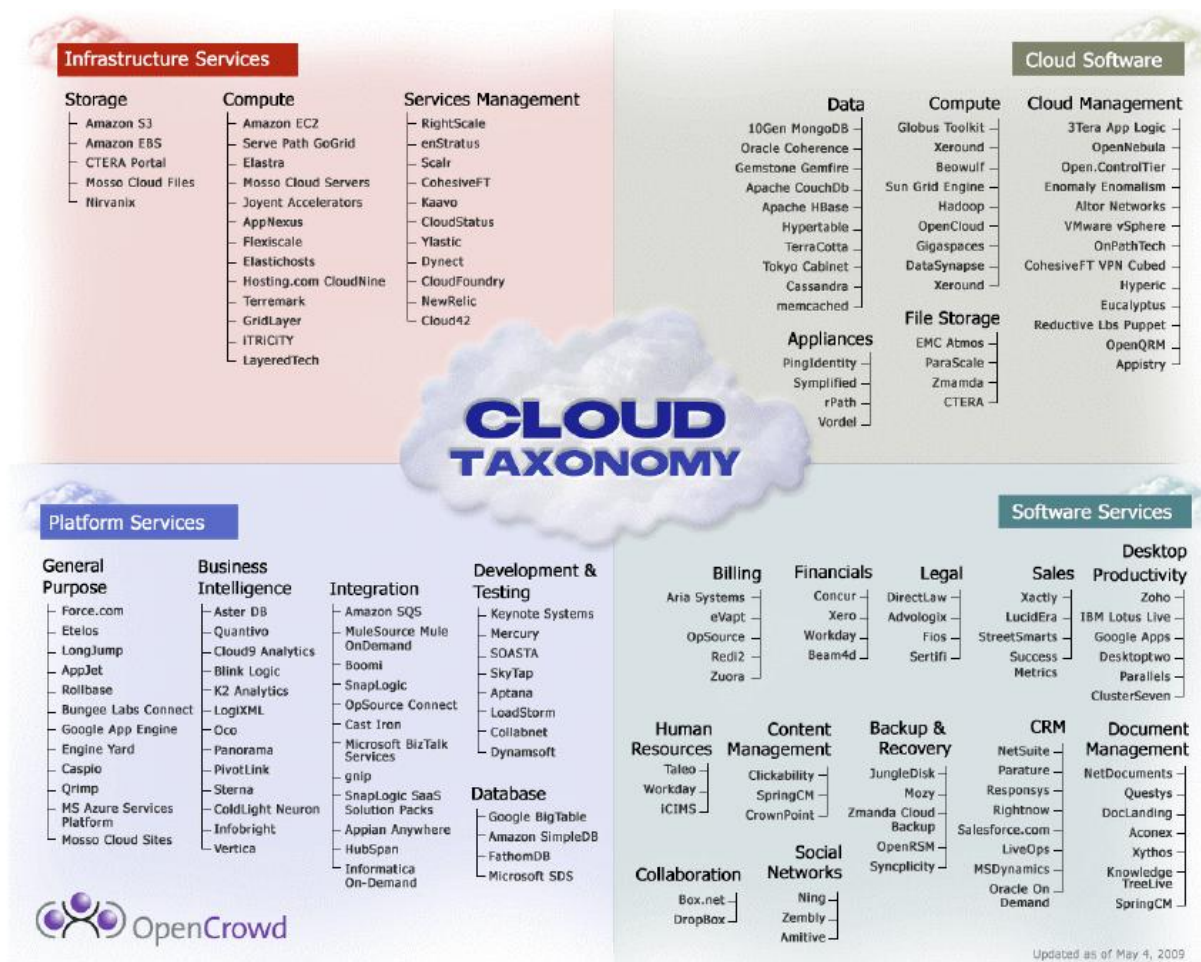
- Generally, SaaS provides the most integrated functionality built directly into the offering, with the least consumer extensibility, and a relatively high level of integrated security (at least the provider bears a responsibility for security).
- PaaS is intended to enable developers to build their own applications on top of the platform. As a result it tends to be more extensible than SaaS, at the expense of customer-ready features. This tradeoff extends to security features and capabilities, where the built-in capabilities are less complete, but there is more flexibility to layer on additional security.
- IaaS provides few if any application-like features, but enormous extensibility. This generally means less integrated security capabilities and functionality beyond protecting the infrastructure itself. This model requires that operating systems, applications, and content be managed and secured by the cloud consumer.

The key takeaway for security architecture is that the lower down the stack the cloud service provider stops, the more security capabilities and management consumers are responsible for implementing and managing themselves.

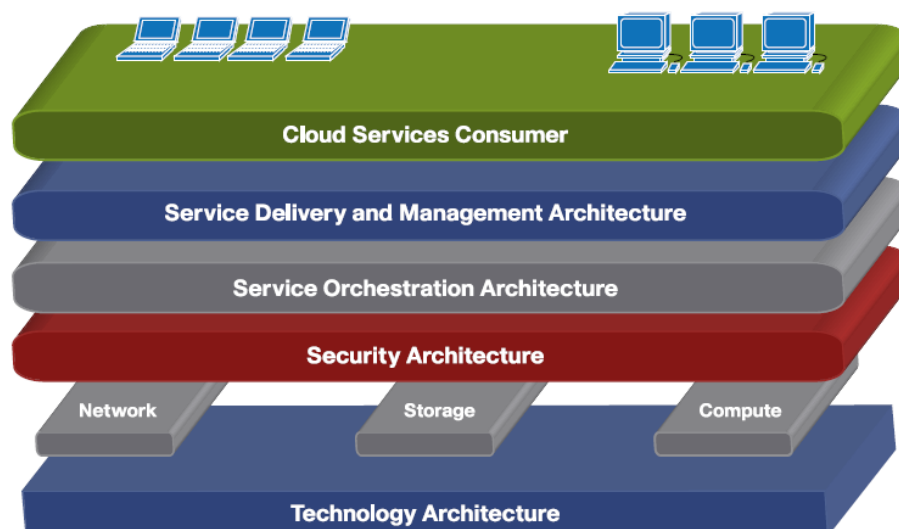
In the case of SaaS, this means that service levels, security, governance, compliance, and liability expectations of the service and provider are contractually stipulated; managed to; and enforced. In the case of PaaS or IaaS it is the responsibility of the consumer's system administrators to effectively manage the same, with some offset expected by the provider for securing the underlying platform and infrastructure components to ensure basic service availability and security. It should be clear in either case that one can assign/transfer responsibility but not necessarily accountability.

Narrowing the scope or specific capabilities and functionality within each of the cloud delivery models, or employing the functional coupling of services and capabilities across them, may yield derivative classifications. For example "Storage as a Service" is a specific sub-offering within the IaaS 'family'.

The OpenCrowd taxonomy shown below demonstrates the swelling ranks of solutions available today across each of the previously defined models.



5. Cisco Cloud Reference Architecture Framework [21]



Cisco has developed the depiction of a cloud reference architecture model, which portrays the architectural layers, connected via APIs and repositories. If we study the framework more closely, the following aspects can be articulated.

- Data center technology architecture: Consists of three salient blocks of network, compute, and storage. This layer hosts all the services that are delivered to a cloud consumer or subscriber.
- Security layer: The key takeaway in this layer is that security is blanketed as an end-to-end architecture across all aspects of the framework. Security is considered as one of the key challenges to be solved in a cloud framework; hence, it has to be accounted for in a comprehensive sense.
- Service Orchestration layer: Implemented with configuration repository enablers. The configuration repository stores key information such as service catalogue, asset inventory, and resource-to-service mappings. This layer is an important layer because it maps the technology components to the service components and serves as a reference point during service provisioning. The service orchestration layer is the “glue” that integrates the lower layers to create a service for delivery.
- Service delivery and management layer: the layer where infrastructure and service management function take place.
- Cloud services consumers layer: the consumer-facing layer, usually exposed via a portal-like solution. This is the layer where service is defined, requested, and managed by the consumer.

Let’s walk through a use case scenario where this framework is utilized.

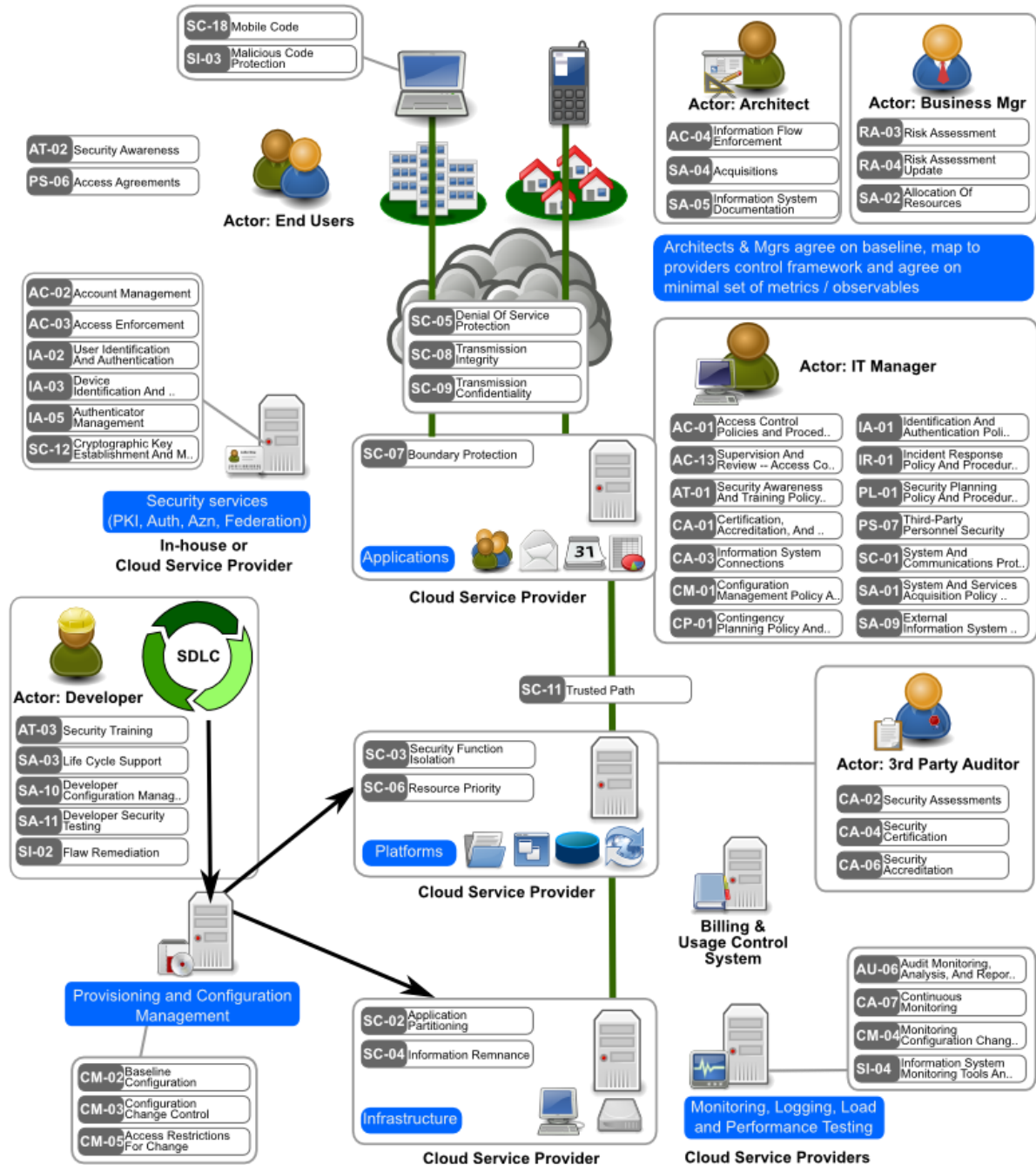
1. Consumer logs on to a cloud portal and verifies/updates credentials and information.
2. Based on the consumer entitlement, a selected set of services are identified and presented for definition.
3. The end user selects the service for consumptions and triggers a service-provisioning request.
4. Resources are marked as reserved for service, and a new request is created for services provisioning.
5. The individual domains of compute, network, and storage are configured and provisioned, with requested security and service-level agreements (SLAs), for service delivery.

Hence, this framework provides a working structure to create, define, orchestrate, and delivery IT service via a cloud.

6. Open Security Architecture (OSA): Secure Architecture Models [9]

Open Security Architecture (OSA) provides free frameworks that are easily integrated in applications, for the security architecture community. Its patterns are based on schematics that show the information traffic flow for a particular implementation as well as policies implemented at each step for security reasons. The following description of the proposed cloud computing architecture, also shown the figure below, should help the reader envision the components of cloud computing architectures along with descriptions of elements that make it secure. The important entities

involved in the data flow are end users, developers, system architect, 3rd party auditors and the cloud itself. The following summary looks at their attributions and mechanisms available for them.



08_02_Pattern_011_15_Cloud_Computing.svg

OSA is licensed according to Creative Commons Share-alike.

Please see: <http://www.opensecurityarchitecture.org/cms/community/license-terms>.

6.1 End Users

End Users need to access certain resources in the cloud and should be aware of access agreements such as acceptable use or conflict of interest. In this model, end user signatures may be used to confirm someone is committed to such policies. The client organization should run mechanisms to detect vulnerable code or protocols at entry points such as firewalls, servers, or mobile devices and upload patches on the local systems as soon as they are found. Thus, this approach ensures security on the end users and on the cloud alike. However, the cloud needs to be secure from any user with malicious intent that may attempt to gain access to information or shut down a service. For this reason, the cloud should include a denial of service (DOS) protection. One way of enforcing DOS protection is done by improving the infrastructure with more bandwidth and better computational power which the cloud has abundantly. However, in the more traditional sense, it involves filtering certain packets that have similar IP source addresses or server requests. The next issue concerning the cloud provider to end users is transmission integrity. One way of implementing integrity is by using secure socket layer (SSL) or transport layer security (TLS) to ensure that the sessions are not being altered by a man in the middle attack. At a lower level, the network can be made secure by the use of secure internet protocol (IPsec). Lastly, the final middle point between end users and the cloud is transmission confidentiality or the guarantee that no one is listening on the conversation between authenticated users and the cloud. The same mechanisms mentioned above can also guarantee confidentiality.

6.2 System Architects

System architects are employed with writing the policies that pertain to the installation and configuration of hardware components such as firewalls, servers, routers, and software such as operating systems, thin clients, etc. They designate control protocols to direct the information flow within the cloud such as router update/queuing protocols, proxy server configurations or encrypted tunnels.

6.3 Developers

Developers building an application in the cloud need to access the infrastructure where the development environment is located. They also need to access some configuration server that allows them to test applications from various views. Cloud computing can improve software development by scaling the software environment through elasticity of resources. For example, one developer can get extra hard space as an on-demand resource, instead of placing a work order and wait for several days for the permission. Developers may desire extra virtual machines to either generate test data or to perform data analysis, processes which take significant time. Also, using more processing power from the cloud can help in catching up with the development schedule. The cloud also helps developers create multiple evaluation versions environments for their applications, bypassing the need to incorporate additional security within the application and placing the burden on the cloud provider. One significant drawback of cloud computing at the moment is its limitations to Intel x86 processor architecture. Even if this may very well change in the future, it is another stumbling block that developers and cloud computing experts need to overcome. Software

monitoring may be done by monitoring API calls for server requests. With an architectural model where data is centralized, all eyes are focused in one direction, which implies better monitoring, although ultimately the issue rests with the developers/clients on how much effort will be directed in this regard. As far as security patches for the software as service approach, updating a patch is easier done in the cloud and shared with everyone seamlessly, rather than finding every machine that has the software installed locally.

6.4 Third Party Auditors

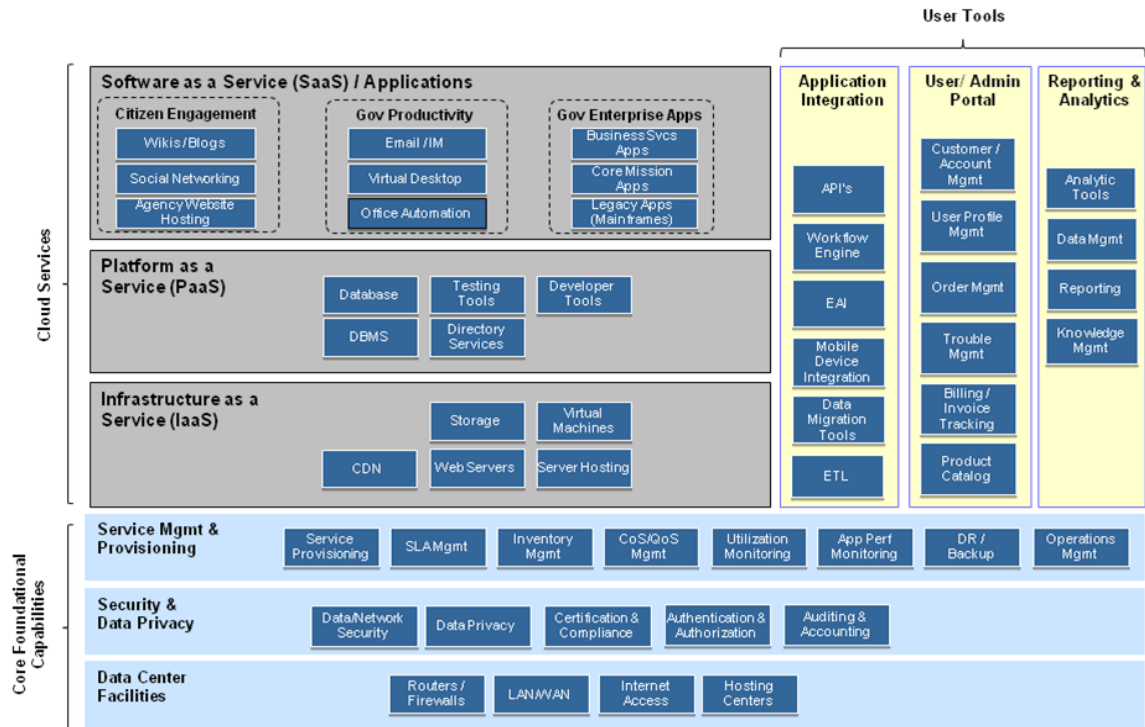
Third party auditors are used by clients and providers alike to determine the security of the cloud implementation. Depending on the level of commitment to security and usefulness in obtaining a competitive edge, a cloud vendor may choose to submit itself to regular security assessments in an attempt to obtain accreditation. The accreditation process needs to be undertaken every three years. Thus, in order to lower the constraints on the cloud vendor, some organizations may implement continuous monitoring of the cloud system.

6.5 Overview

The cloud is the resource that incorporates routers, firewalls, gateway, proxy and storage servers. The interaction among these entities needs to occur in a secure fashion. For this reason, the cloud, just like any data center, implements a boundary protection also known as the demilitarized zone (DMZ). The most sensitive information is stored behind the DMZ. Other policies that run in the cloud are resource priority and application partitioning. Resource priority allows processes or hardware requests in a higher priority queue to be serviced first. Application partitioning refers to the usage of one server or storage device for various clients that may have data encrypted differently. The cloud should have policies that divide the users' view of one application from the backend information storage. This may be solved by using virtualization, multiple processors or network adaptors.

7. GSA: FCCI (Federal Cloud Computing Initiative) [6][7]

7.1 Government Cloud Computing Framework



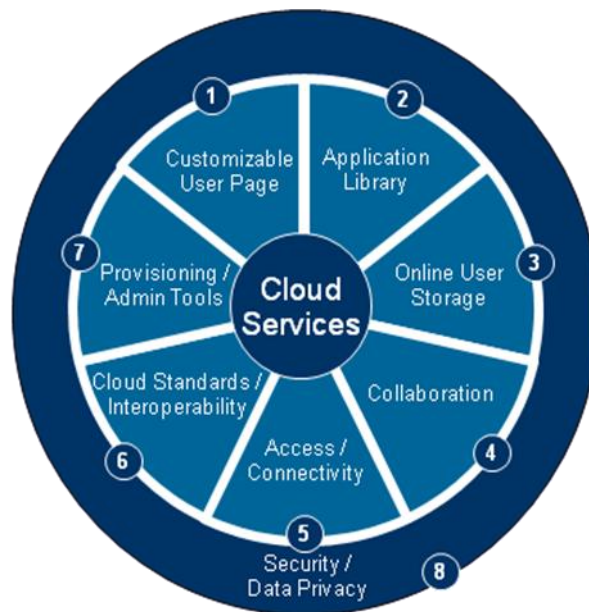
Delivery Model Overview

Model	Capability Provided	Example Services
SaaS	To use the provider's applications running on a cloud infrastructure and accessible from various client devices through a thin client interface such as a Web browser	<ul style="list-style-type: none"> ▪ Citizen Engagement (Wikis, Blogs, Data.gov) ▪ Government Productivity (Cloud based tools) ▪ Business Enablement (Salesforce.com) ▪ Enterprise Applications (Core Mission & Business Svcs)
PaaS	To deploy onto the cloud infrastructure consumer-created applications using programming languages and tools supported by the provider (e.g., java, python, .Net)	<ul style="list-style-type: none"> ▪ Database and Database Management Systems ▪ Developer / Testing Tools ▪ Virtual Environments
IaaS	To provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications	<ul style="list-style-type: none"> • Computing • Storage • Application hosting

Federal Government Considerations for Cloud Computing:

- **Security & Data Privacy:** Offer different levels of security and data privacy based on the application and nature of the services provided. Potential standardize Low, Med and High categories for Simplicity.
- **Delivery & Operations:** Enable adoption of Cloud Computing services in different Cloud models including Public, Private, Hybrid and Community models.
- **Interoperability & Integration:** Develop interoperability standards in conjunction with the industry to provide interoperability at the data infrastructure, platform and application levels.

7.1.1 Federal Cloud Computing Cloud Program Services Model



Components	Description
Customizable User Page	<ul style="list-style-type: none"> • One stop shop/ single view for Users to manage interactions with Cloud Services
Application Library	<ul style="list-style-type: none"> • Applications and content made available to Users through a simple GUI. These applications / content can be downloaded easily from the Library.
Online User Storage	<ul style="list-style-type: none"> • Online storage for Users to maintain and manage individual User files, data and objects
Collaboration	<ul style="list-style-type: none"> • Widgets that allow Users to collaborate and generate content. These widgets may include Wikis, Blogs, and IM
Access /	<ul style="list-style-type: none"> • Network connectivity and devices to access Cloud services

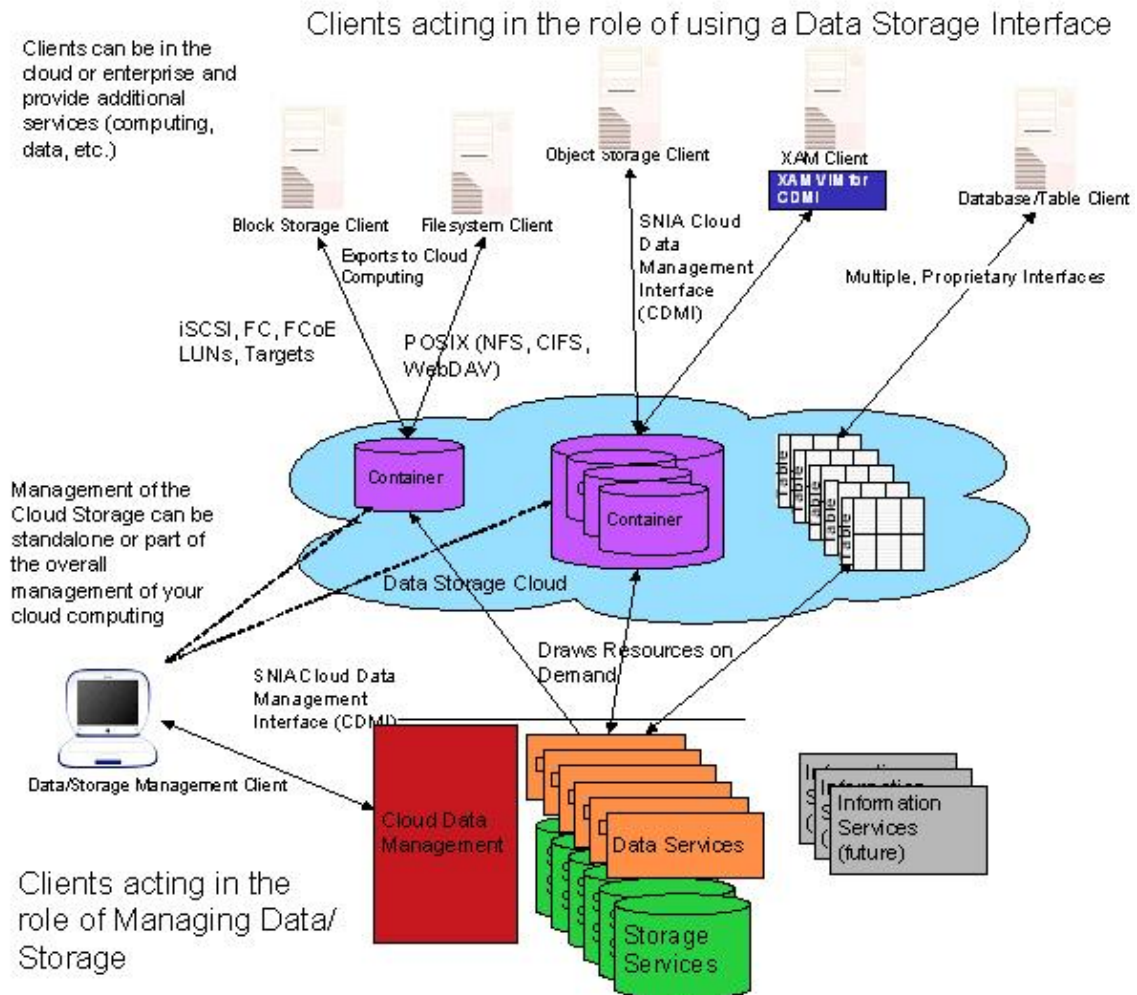
Connectivity	
Cloud Standards / Interoperability	<ul style="list-style-type: none"> • Cloud standards allowing the integration and interoperability of services from multiple Clouds
Provisioning / Admin Tools	<ul style="list-style-type: none"> • Provisioning and administrative tools to provide control of User profiles, access technical support, and manage privilege, authorization to applications and content
Security / Data Privacy	<ul style="list-style-type: none"> • Applying security framework and data privacy standards for Federal Cloud Computing

8. SNIA standard: Cloud Data Management Interface [3]

The Storage Networking Industry Association (SNIA) announced at SNW Spring 2010, the formal approval of the Cloud Data Management Interface (CDMI) as a SNIA Architecture standard. This milestone marks the first industry-developed open standard for cloud computing and will allow for interoperable cloud storage implementations from cloud service providers and storage vendors.

SNIA proposes a formal term for cloud storage, i.e. Data Storage as a Service (DaaS), which means "delivery over a network of appropriately configured virtual storage and related data services, based on a request for a given service level." (**Note:** Some organization, e.g. Cloud Security Alliance, claims "Storage as a Service" as a specific sub-offering within the IaaS 'family'.) SNIA defines CDMI, the standards for cloud storage, specifically, the standards for the interface providing the means to access cloud storage and to manage the data stored there.

The Reference Model for Cloud Storage Interfaces is shown below:



This model shows multiple types of cloud data storage interfaces that are able to support both legacy and new applications. All of the interfaces allow storage to be provided on demand, drawn from a pool of resources. The capacity is drawn from a pool of storage capacity provided by storage services. The data services are applied to individual data elements, as determined by the data system metadata. Metadata specifies the data requirements on the basis of individual data elements or on groups of data elements (containers).

8.1 Cloud Data Management Interface (CDMI):

As shown in the above figure “Cloud Storage Reference Model”, the SNIA Cloud Data Management Interface (CDMI) is the functional interface that applications may use to create, retrieve, update, and delete data elements from the cloud. As part of this interface, the client will be able to discover the capabilities of the cloud storage offering and to use this interface to manage containers and the data that is placed in them. In addition, data system metadata can be set on containers and their contained data elements through this interface.

This interface may also be used by administrative and management applications to manage containers, domains, security access, and monitoring/billing information, even for storage that is

functionally accessible by legacy or proprietary protocols. The capabilities of the underlying storage and data services are exposed so that clients can understand the offering.

8.2 Data Storage Interfaces:

8.2.1 Existing Data Storage Interfaces

An important part of any DaaS offering is the support of legacy clients. Support is accommodated with existing standard protocols such as iSCSI (and others) for block and CIFS/NFS or WebDAV for file network storage. In the case of block storage, a LUN, or virtual volume, is the granularity of allocation. For file protocols, a file system is the unit of granularity. In either case, the actual storage space can be thin provisioned and billed for, based on actual usage. Data services, such as compression and deduplication, can be used to further reduce the actual space consumed.

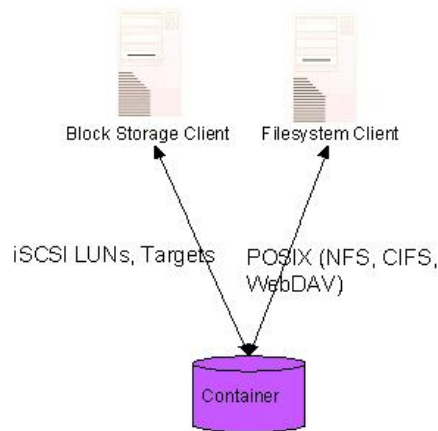


Fig. Existing Data Storage Interface Standards

In this model, we abstract the underlying storage space exposed by these interfaces using the notion of a **container**. A container is not only a useful abstraction for storage space, but also serves as a grouping of the data stored in it and a point of control for applying data services in the aggregate.

Managing this storage is typically done out of band of these standard data storage interfaces, either through an API, or more commonly, through an administrative browser-based user interface. This interface may be used to invoke other data services as well, such as snapshot and cloning.

8.2.2 Storage Interfaces for Database/Table Data

Another type of DaaS offering is one of simple table space storage, allowing for horizontal scaling of database-like operations that certain applications need. Rather than virtualizing relational database instances, table space storage offers a new data storage interface of limited functionality, with the emphasis on scalability rather than features. Scalability allows the tables to be partitioned across multiple nodes based on common key values, affording horizontal scalability at the expense of functions that can typically only be implemented by a vertically-scaled relational database.

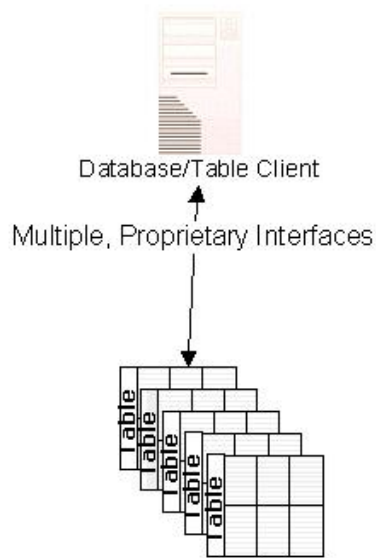


Fig. Storage Interfaces for Database/Table Data

8.2.3 Storage Interfaces for Object Storage Client Data

This type of interface treats every data object as accessible via a unique URI. It can then be fetched using the standard HTTP protocol, and a browser can be used to invoke the appropriate application to deal with the data.

Each data object is created, retrieved, updated, and deleted (CRUD semantics) as a separate resource. In this type of interface, a container, if used, is a simple grouping of data objects for convenience. Nothing prevents the concept of containers, in this case, from being hierarchical, although any given implementation might support only a single level of such.

8.3 Data Management in the Cloud

To address the needs of enterprise applications with cloud storage, there is increasing pressure to offer better quality of service and the deployment of additional data services. However, cloud storage may lose its benefit of simplicity and the abstraction of complexity, as additional data services are applied and the implication that these services need to be managed. One can hardly have cloud storage customers setting up backup schedules through dedicated user interfaces, deploying data services individually for their data elements, and so on.

The SNIA Storage Industry Resource Domain Model [SIRDM] gives us a way to minimize this complexity and address the need for cloud storage to remain simple. By using the different types of metadata discussed in that model for a cloud storage interface, we can create an interface that allows offerings to meet the requirements of the data without adding undo complexity to the management of that data.

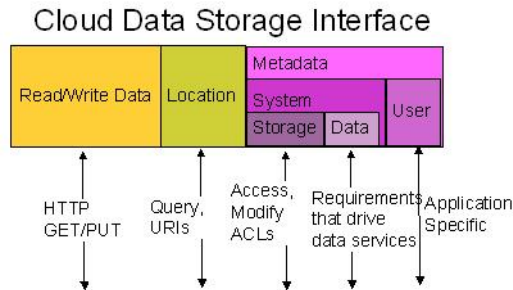


Fig. Using the Resource Domain Model

By supporting metadata in a cloud storage interface standard and prescribing how the storage system and data system metadata is interpreted to meet the requirements of the data, we can retain the simplicity required by the cloud storage paradigm and still address the requirements of enterprise applications and their data.

9. Elastra: A Cloud Technology Reference Model for Enterprise Clouds [22]



- Facilities & Logistics Management, Organizationally & Geographically Decentralized Software & Hardware: The basic data center, which is now global and possibly cross organizational and exposes power & cooling information.
- Licensing, Security, Identity & Trust: the control point for compliance and auditing, which adds trust, identity and licensing.
- Configuration management: deals with HW/SW/Network/Storage Settings, Software Packages, and Dependencies
- Resource management: deals with reservations from a pool of excess capacity in storage, computing, and network.
- Hyperlinked Models & Metadata: what uses or contains what other things

- System Lifecycles & Management Processes: when and how can things change.
- Governance: Who has authority / responsibility to change, and how?
- Constraints & Policies: How are concerns addressed in the design?
- Testing, Monitoring & Operations: How do we manage & verify?

10. Comparison

In this section, we compare all the presented models on different abstraction levels (business/architecture), and indicate each model the key difference.

Model	Comparison on Business level	Comparison on Architecture level	Key Difference
DMTF	For general cc system	Defines actors (cc provider/user/develop), interfaces for interaction bw actors, data artifacts for info exchanged via interfaces, and profiles for contexts.	Focuses on provider interfaces, i.e. how to support interaction among actors.
Use case group	For general cc system	Defines actors (cc provider/user/develop), cloud components, service models, diff interfaces for diff service models, and service management.	Compared to DMTF, adds service models/cloud components/ service managements.
IBM	For general cc system	Defines actors (cc provider/user/develop), cloud components, service models, business/operational support services for cloud providers.	Compared to use case group's model, adds more details for service management.
CSA	For general cc system	Defines resources stack, IaaS/PaaS/SaaS and their relationship /dependency.	The only one that defines the dependency & build up of IaaS to PaaS to SaaS
Cisco	For general cc system	Defines a layered model which considers resources, security, service, and consumers layer	The only layered architecture
OSA	Specifically for cloud security	Defines actors (users/system architects/developers/3 rd party auditors), security policies at each step following info traffic flow.	Focuses on security architecture.
GSA	Specifically for government cloud services	Defines government service models, cloud components, user tools, federal cloud services model.	Focuses on government's unique needs, tailored all components with government's needs
SNIA	Specifically for cloud storage	Defines "DaaS", & CDMI which is the interface for accessing and managing cloud storage.	1 st industry-developed open standard for cloud computing. Focuses on cloud storage.
Elastra	Specifically for enterprise cloud	Defines a management framework for enterprise clouds, which focuses on management of diff. components and lifecycles.	Compared to IBM, Elastra considers only cloud management.

11. Reference

- [1] Cloud Computing Use Cases White Paper, <http://groups.google.com/group/cloud-computing-use-cases>
- [2] DMTF, "Interoperable Clouds White Paper", http://www.dmtf.org/about/cloud-incubator/DSP_IS0101_1.0.0.pdf
- [3] SNIA: <http://cdmi.sniacloud.com/>
- [4] Cloud Security Alliance, "Security Guidance for Critical Areas of Focus In Cloud Computing V2.1", www.cloudsecurityalliance.org/csaguide.pdf
- [5] SNIA, "Cloud Storage for Cloud Computing", www.snia.org/cloud/CloudStorageForCloudComputing.pdf
- [6] GSA, "Cloud Computing Initiative", http://www.info.apps.gov/sites/default/files/Cloud_Computing_Initiative_Briefing_Book_0.doc
- [7] GSA, "Cloud Computing Initiative Vision and Strategy Document (DRAFT)", http://info.apps.gov/sites/default/files/Cloud_Computing_Strategy_0.ppt
- [8] Department of Homeland Security, "Cloud Computing from the Security Perspective", http://www.info.apps.gov/sites/default/files/Cloud_Computing_Security_Perspective.doc
- [9] Open Security Architecture (OSA), "Cloud Computing Patterns", <http://www.opensecurityarchitecture.org/cms/library/patternlandscape/251-pattern-cloud-computing>
- [10] Traian Andrei, "Cloud Computing Challenges and Related Security Issues", <http://www.cse.wustl.edu/~jain/cse571-09/ftp/cloud/index.html>
- [11] Jinesh Varia (@Amazon Web Services), "Cloud Architectures", <http://jineshvaria.s3.amazonaws.com/public/cloudarchitectures-varia.pdf>
- [12] Gerald Kaefer (@Siemens), "Cloud Computing Architecture", <http://www.sei.cmu.edu/library/assets/presentations/Cloud%20Computing%20Architecture%20-%20Gerald%20Kaefer.pdf>
- [13] "A Reference Architecture for Cloud Lifecycle Management", <http://www.spokenword.org/program/1278676>
- [14] IBM, "Cloud Computing", http://download.boulder.ibm.com/ibmdl/pub/software/dw/wes/hipods/Cloud_computing_wp_final_8Oct.pdf
- [15] IBM, "Private Cloud Computing Architectural Concepts", http://www.google.com/url?sa=t&source=web&cd=1&ved=0CCAQFjAA&url=http%3A%2F%2Fwww-05.ibm.com%2Ffr%2Fevents%2Fst_cloud_computing%2FCloud_Architecture_Overview_STC_C16032010.pdf&ei=tgrLTMXxEIPGIQec-4yuAQ&usg=AFQjCNHAAILq03xIZxq67J6wOHIthr-2sg
- [16] IBM, "IBM Smart Analytics Cloud", www.redbooks.ibm.com/redbooks/pdfs/sg247873.pdf
- [17] IBM, "Cloud Computing: Save Time, Money, and Resources with a Private Test Cloud", www.redbooks.ibm.com/redpapers/pdfs/redp4553.pdf
- [18] IBM, "Cloud Security Guidance - IBM Recommendations for the Implementation of Cloud Security", <http://www.redbooks.ibm.com/redpapers/pdfs/redp4614.pdf>
- [19] IBM, "IBM Cloud Computing – Service Management", <http://www.google.com/url?sa=t&source=web&cd=1&ved=0CB8QFjAA&url=https%3A%2F%2Fqrp.research.ibm.com%2F LotusQuickr%2Fslidemos%2FMain.nsf%2F0%2F6DFC3C4967F62E188525762C006F21DC%2F%24file%2FLindquist IBM Cloud Computing->

[Service_Management_v4_D.pdf&ei=4BDLTLGvGIGglAfKzrnxAQ&usg=AFQjCNFE6ANMY9VBt
nfX66FGR8sF2lzNhA](#)

- [20] IBM, "Cloud Computing - The Importance of Integrated Service Management",
[http://www.google.com/url?sa=t&source=web&cd=1&ved=0CBMQFjAA&url=https%3A%2F%2Fwww-950.ibm.com%2Fevents%2Fwww%2Fgrp%2Fgrp004.nsf%2FvLookupPDFs%2FCloud%2520Computing%2520-%2520The%2520Importance%2520of%2520Integrated%2520Service%2520Management%2520Minneapolis%25202010%2F%24file%2FCloud%2520Computing%2520-%2520The%2520Importance%2520of%2520Integrated%2520Service%2520Management%2520Minneapolis%25202010.pdf&ei=0B_LTPqyLIH_8AbD_rWrAQ&usg=AFQjCNH_CYyQgidqx
o7KwGHbtd4zWqrE7w](#)
- [21] CISCO, "Cisco Cloud Computing - Data Center Strategy, Architecture, and Solutions",
[http://www.google.com/url?sa=t&source=web&cd=1&ved=0CB0QFjAA&url=http%3A%2F%2Fwww.cisco.com%2Fweb%2Fstrategy%2Fdocs%2Fgov%2FCiscoCloudComputing_WP.pdf&ei=Mx_LTOHwOIL68AaPk_zFAQ&usg=AFQjCNEgDF69wa7aswl7k7gmINpy6QhfHg](#)
- [22] Stuart Charlton, "Cloud Computing and the Next Generation of Enterprise Architecture",
[http://www.slideshare.net/StuC/cloud-computing-and-the-nextgeneration-of-enterprise-architecture-cloud-computing-expo-2008-presentation](#)