UNIT - IV

4.1 INTER-CLOUD RESOURCE MANAGEMENT

Cloud of Clouds (Inter cloud)

Inter cloud or 'cloud of clouds'-refer to a theoretical model for cloud computing services.

Combining many different individual clouds into one seamless mass in terms of on-demand operations.

The inter cloud would simply make sure that a cloud could use resources beyond its reach.

Taking advantage of pre-existing contracts with other cloud providers.

Each single cloud does not have infinite physical resources or ubiquitous geographic footprint.

A cloud may be saturated to the computational and storage resources of its infrastructure.

It would still be able satisfy such requests for service allocations sent from its clients.

A single cloud cannot always fulfill the requests or provide required services.

When two or more clouds have to communicate with each other, or another intermediary comes into play and federates the resources of two or more clouds.

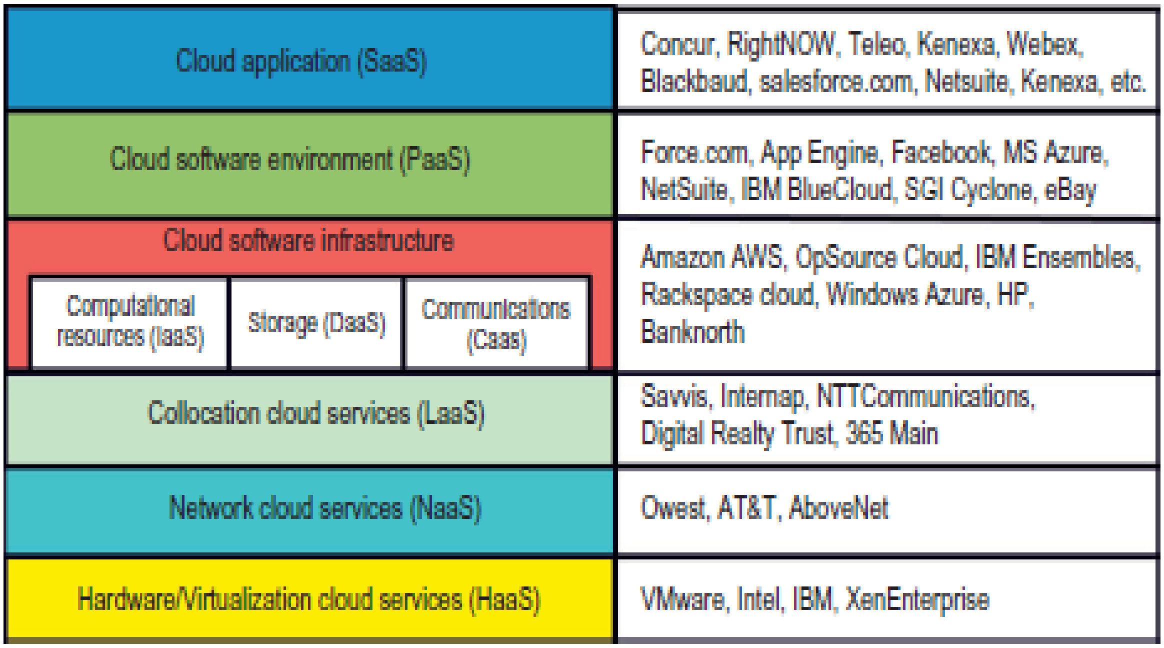
In inter cloud, intermediary is known as “cloud broker” or simply “broker.”

Broker is the entity which introduces the cloud service customer (CSC) to the cloud service provider (CSP)

Inter-Cloud Resource Management Consists of

- Extended Cloud Computing Services
- Resource Provisioning and Platform Management
- Virtual Machine Creation and Management
- Global Exchange of Cloud Resources

4.1.1 Extended Cloud Computing Services



Cloud application (SaaS)	Concur, RightNOW, Teleo, Kenexa, Webex, Blackbaud, salesforce.com, Netsuite, Kenexa, etc.
Cloud software environment (PaaS)	Force.com, App Engine, Facebook, MS Azure, NetSuite, IBM BlueCloud, SGI Cyclone, eBay
Cloud software infrastructure Computational resources (IaaS) | Storage (DaaS) | Communications (Caas)	Amazon AWS, OpSource Cloud, IBM Ensembles, Rackspace cloud, Windows Azure, HP, Banknorth
Collocation cloud services (LaaS)	Savvis, Internap, NTTCommunications, Digital Realty Trust, 365 Main
Network cloud services (NaaS)	Owest, AT&T, AboveNet
Hardware/Virtualization cloud services (HaaS)	VMware, Intel, IBM, XenEnterprise

Fig: Six layers of cloud services and their providers

Six layers of cloud services

- Software as a Service(SaaS)
- Platform as a Service(PaaS)
- Infrastructure as a Service(IaaS)
- Hardware / Virtualization Cloud Services(HaaS)
- Network Cloud Services (NaaS)
- Collocation Cloud Services(LaaS)

- ➢ The top layer offers SaaS which provides cloud application.
- ➢ PaaS sits on top of IaaS infrastructure.
- ➢ The bottom three layers are more related to physical requirements.
- ➢ The bottommost layer provides Hardware as a Service (HaaS).
- ➢ NaaS is used for interconnecting all the hardware components.

- Location as a Service (LaaS), provides security to all the physical hardware and network resources. This service is also called as Security as a Service.
- The cloud infrastructure layer can be further subdivided as
  - Data as a Service (DaaS)
  - Communication as a Service (CaaS)
  - Infrastructure as a Service(IaaS)
- Cloud players are divided into three classes:
  - Cloud service providers and IT administrators
  - Software developers or vendors
  - End users or business users.

Cloud Players	IaaS	PaaS	SaaS
IT administrators/ Cloud Providers	Monitor SLAs	Monitor SLAs and enable service platforms	Monitor SLAs and deploy software
Software developers (Vendors)	To deploy and store data	Enabling platforms	Develop and deploy software
End users or business users	To deploy and store data	To develop and test software	Use business software

Table: Cloud Differences in Perspective of Providers, Vendors, and Users

4.1.1 Cloud Service Tasks and Trends

- SaaS is mostly used for Business Applications
- Eg: CRM (Customer Relationship Management) used for business promotion, direct sales, and marketing services
- PaaS is provided by Google, Salesforce.com, and Facebook etc.
- IaaS is provided by Amazon, Windows Azure, and RackRack etc.
- Collocation services Provides security to lower layers.
- Network cloud services provide communications.

4.1.2 Software Stack for Cloud Computing

- The software stack structure of cloud computing software can be viewed as layers.
- Each layer has its own purpose and provides the interface for the upper layers.
- The lower layers are not completely transparent to the upper layers.

4.1.3 Runtime Support Services

- Runtime support refers to software needed in applications.
- The SaaS provides the software applications as a service, rather than allowing users purchase the software.
- On the customer side, there is no upfront investment in servers.

<u>4.1.2 Resource Provisioning (Providing) and Platform Deployment</u>

There are techniques to provision computer resources or VMs. Parallelism
is exploited at the cluster node level.

4.1.2.1 Provisioning of Compute Resources (VMs)

- ➢ Providers supply cloud services by signing SLAs with end users.
- ➢ The SLAs must specify resources such as
  - CPU
  - Memory
  - Bandwidth

Users can use these for a preset (fixed) period.

- ➢ Under provisioning of resources will lead to broken SLAs and penalties.
- ➢ Over provisioning of resources will lead to resource underutilization, and consequently, a decrease in revenue for the provider.
- ➢ Provisioning of resources to users is a challenging problem. The difficulty comes from the following
  - o Unpredictability of consumer demand
  - o Software and hardware failures
  - o Heterogeneity of services

- ○ Power management
- ○ Conflict in signed SLAs between consumers and service providers.

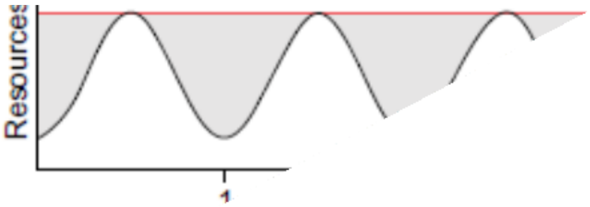
4.1.2.2 Provisioning Methods

Three cases of static cloud resource provisioning policies are considered.

Static cloud resource provisioning

case (a)

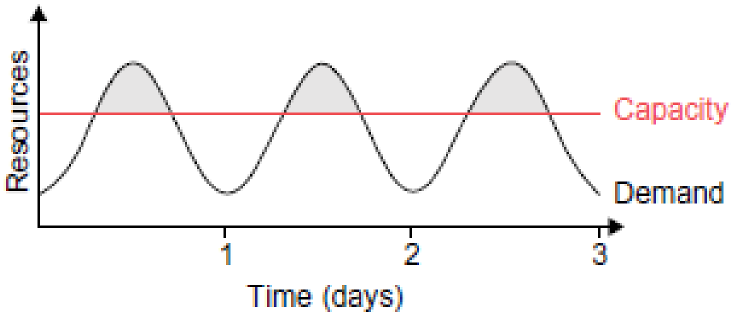
- ➢ over provisioning(Providing) with the peak load causes heavy resource waste (shaded area).



..

case (b)

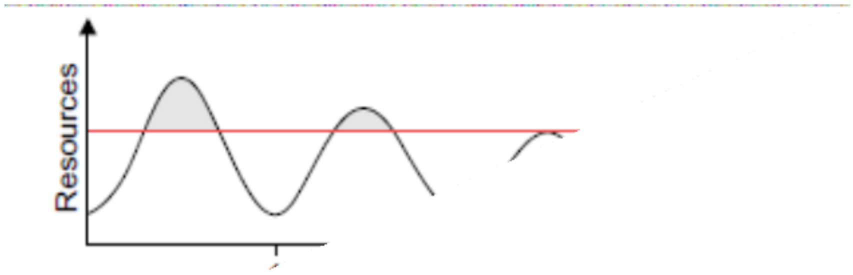
Under provisioning of resources results in losses by both user and provider. Users have paid for the demand (the shaded area above the capacity) is not used by users.



(b) Underprovisioning 1

case (c)

Declining in user demand results in worse resource waste.



Constant provisioning

Fixed capacity to a declining user demand could result in even worse resource waste.

The user may give up the service by canceling the demand, resulting in reduced revenue for the provider.

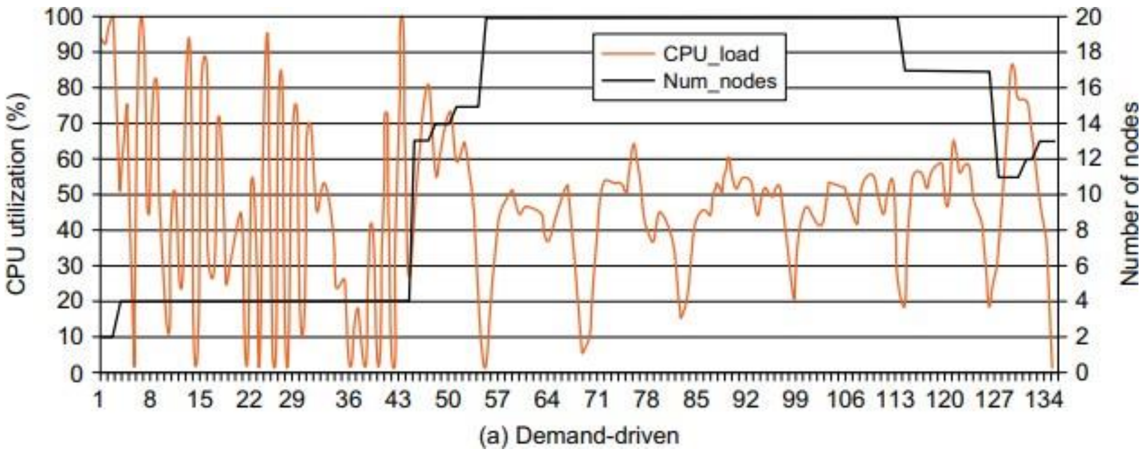
Both the user and provider may be losers in resource provisioning without elasticity.

Resource-provisioning methods are

- Demand-driven method - Provides static resources and has been used in grid computing
- Event-driven method - Based on predicted workload by time.
- Popularity-Driven Resource Provisioning – Based on Internet traffic monitored

4.1.2.3 Demand Driven Methods

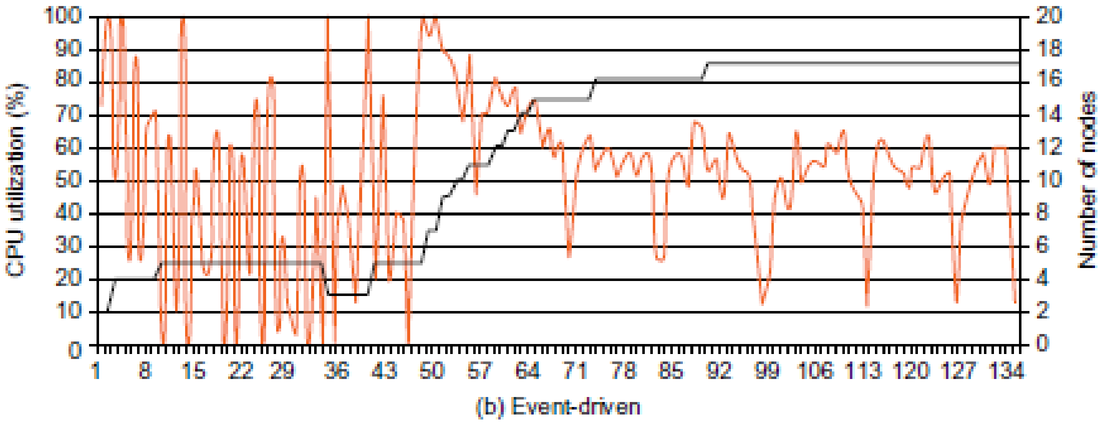
- Provides Static resources
- This method adds or removes nodes (VM) based on the current utilization(Use) level of the allocated resources.
- When a resource has surpassed (exceeded) a threshold (Upperlimit) for a certain amount of time, the scheme increases the resource (nodes) based on demand.
- When a resource is below a threshold for a certain amount of time, then resources could be decreased accordingly.
- This method is easy to implement.
- The scheme does not work out properly if the workload changes abruptly.



(a) Demand-driven

4.1.2.4 Event-Driven Resource Provisioning

- This scheme adds or removes machine instances based on a specific time event.
- The scheme works better for seasonal or predicted events such as Christmastime in the West and the Lunar New Year in the East.
- During these events, the number of users grows before the event period and then decreases during the event period. This scheme anticipates peak traffic before it happens.
- The method results in a minimal loss of QoS, if the event is predicted correctly

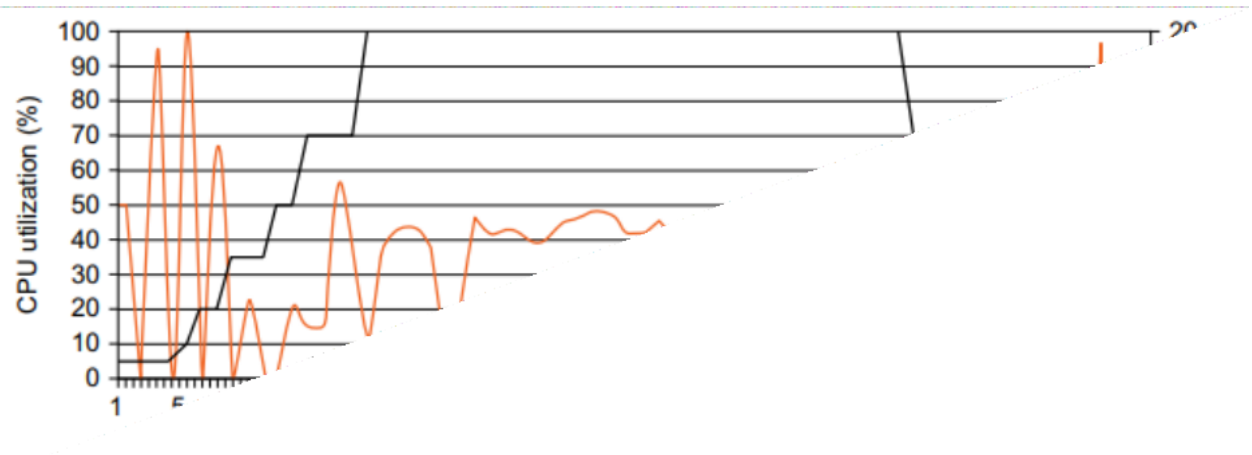


(b) Event-driven

4.1.2.5 Popularity-Driven Resource Provisioning

- Internet searches for popularity of certain applications and allocates resources by popularity demand.

- This scheme has a minimal loss of QoS, if the predicted popularity is correct.
- Resources may be wasted if traffic does not occur as expected.
- Again, the scheme has a minimal loss of QoS, if the predicted popularity is correct.
- Resources may be wasted if traffic does not occur as expected.



4.1.2.6 Dynamic Resource Deployment

- The cloud uses VMs as building blocks to create an execution environment across multiple resource sites.
- Dynamic resource deployment can be implemented to achieve scalability in performance.
- Peering arrangements established between gateways enable the allocation of resources from multiple grids to establish the execution environment.
- Dynamic resource deployment can be implemented to achieve scalability in performance.
- InterGrid is used for interconnecting distributed computing infrastructures.
- InterGrid provides an execution environment on top of the interconnected infrastructures.
- IGG(InterGridGateway) allocates resources from an
  - □ Organization's local cluster (Or)
  - □ Cloud provider.

- Under peak demands, IGG interacts with another IGG that can allocate resources from a cloud computing provider.
- Component called the DVE manager performs resource allocation and management.
- Intergrid gateway (IGG) allocates resources from a local cluster three steps:

(1) Requesting the VMs(Resources)

(2) Enacting (Validate) the leases

(3) Deploying (install) the VMs as requested.

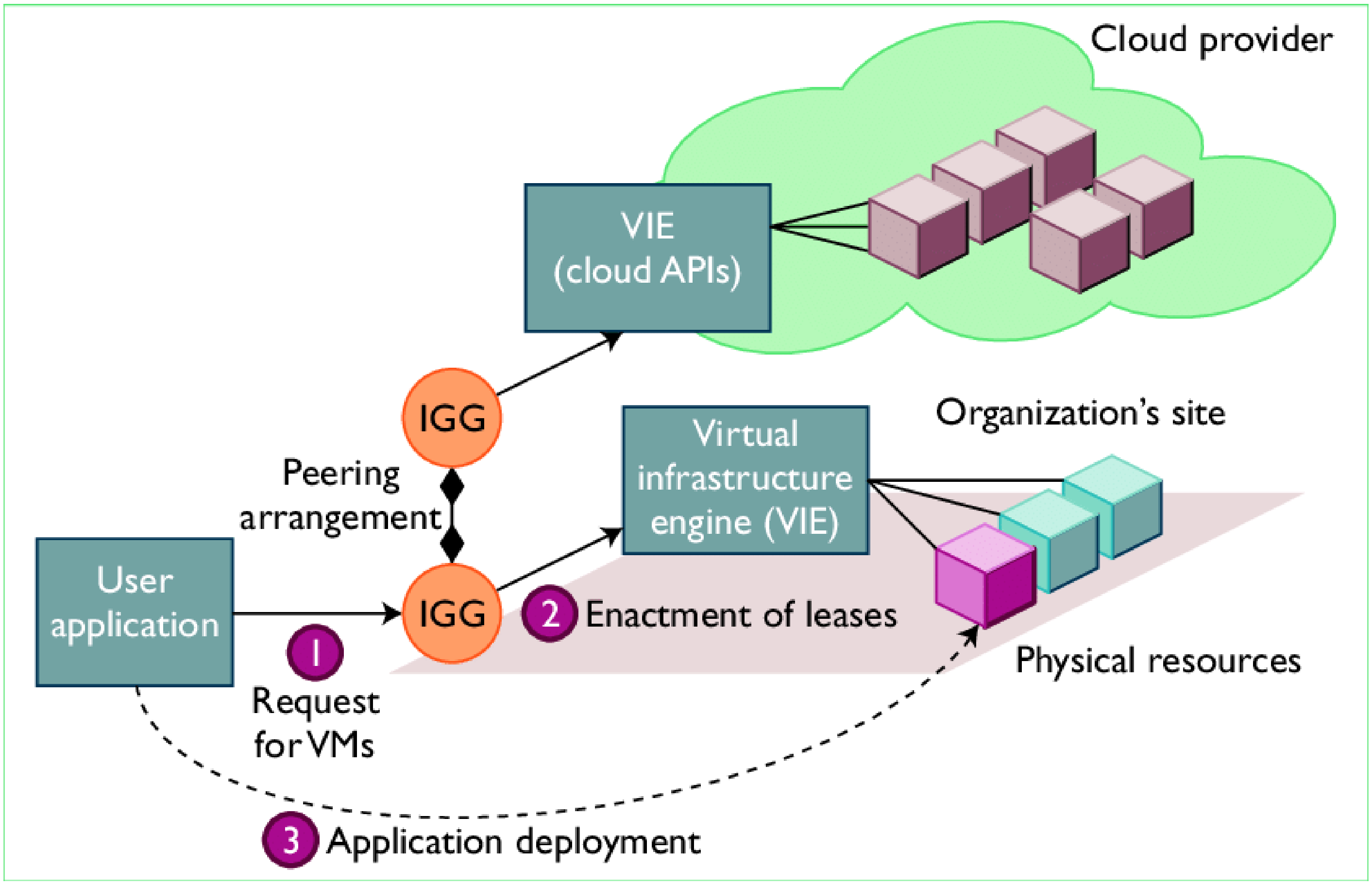


Fig: Cloud resource deployment using an IGG (intergrid gateway) to allocate the VMs from a Local cluster to interact with the IGG of a public cloud provider.

Under peak demand, this IGG interacts with another IGG that can allocate resources from a cloud computing provider.

A grid has predefined peering arrangements with other grids, which the IGG manages.

Through multiple IGGs, the system coordinates the use of InterGrid resources.

An IGG is aware of the peering terms with other grids, selects suitable grids that can provide the required resources, and replies to requests from other IGGs.

Request redirection policies determine which peering grid InterGrid selects to process a request and a price for which that grid will perform the task.

An IGG can also allocate resources from a cloud provider.

The InterGrid allocates and provides a distributed virtual environment (DVE).

This is a virtual cluster of VMs that runs isolated from other virtual clusters.

A component called the DVE manager performs resource allocation and management on behalf of specific user applications.

The core component of the IGG is a scheduler for implementing provisioning policies and peering with other gateways.

The communication component provides an asynchronous message-passing mechanism.

4.1.2.7 Provisioning of Storage Resources

- ➢ Storage layer is built on top of the physical or virtual servers.
- ➢ Data is stored in the clusters of the cloud provider.
- ➢ The service can be accessed anywhere in the world.
- Eg:
  - E-mail system might have millions of users and each user can have thousands of e-mails and consume multiple gigabytes of disk space.
  - Web searching application.
  - To store huge amount of information solid-state drives are used instead of hard disk drives

In storage technologies, hard disk drives may be augmented (increased) with solid-state drives in the future.

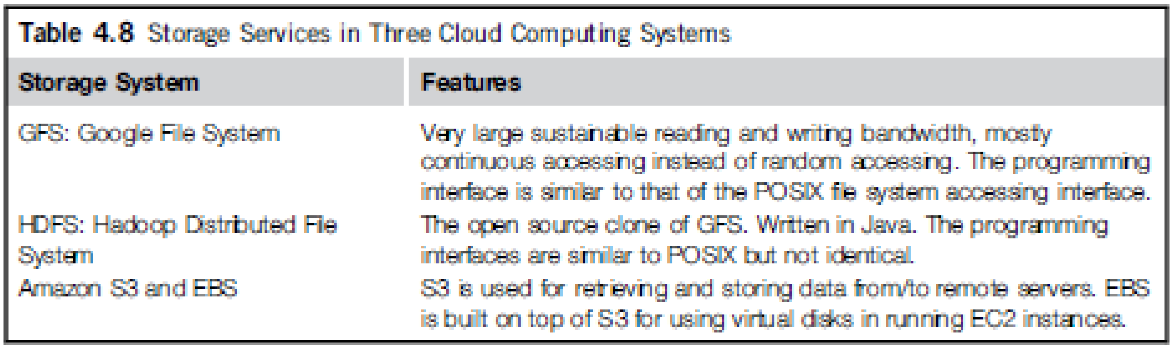


Table 4.8 Storage Services in Three Cloud Computing Systems

Storage System	Features
GFS: Google File System	Very large sustainable reading and writing bandwidth, mostly continuous accessing instead of random accessing. The programming interface is similar to that of the POSIX file system accessing interface.
HDFS: Hadoop Distributed File System	The open source clone of GFS. Written in Java. The programming interfaces are similar to POSIX but not identical.
Amazon S3 and EBS	S3 is used for retrieving and storing data from/to remote servers. EBS is built on top of S3 for using virtual disks in running EC2 instances.

4.5.3 Virtual Machine Creation and Management

The managers provide a public API for users to submit and control the VMs

Independent Service Management:

- Independent services request facilities to execute many unrelated tasks.
- Commonly, the APIs provided are some web services that the developer can use conveniently.

Running Third-Party Applications

- Cloud platforms have to provide support for building applications that are constructed by third-party
  application providers or programmers.
- The APIs are often in the form of services.
- Web service application engines are often used by programmers for building applications.
- The web browsers are the user interface for end users.

Virtual Machine Manager

The manager manage VMs deployed on a set of physical resources

- VIEs(Virtual Infrastructure Engine) can create and stop VMs on a physical cluster
- Users submit VMs on physical machines using different kinds of hypervisors

- To deploy a VM, the manager needs to use its template.
- Virtual Machine Templates contains a description for a VM with the following static information:
  - The number of cores or processors to be assigned to the VM
  - The amount of memory the VM requires
  - The kernel used to boot the VM's operating system.
  - The price per hour of using a VM
- OAR/Kadeploy is a deployment tool
- API(Application Programming Interface) - An API is a software intermediary that makes it possible for application programs to interact with each other and share data

Virtual Machine Templates

- A VM template is analogous to a computer's configuration and contains a description for a VM with the following static information:
- The number of cores or processors to be assigned to the VM
- The amount of memory the VM requires
- The kernel used to boot the VM's operating system
- The disk image containing the VM's file system
- The price per hour of using a VM

Distributed VM Management

- A distributed VM manager makes requests for VMs and queries their status.
- This manager requests VMs from the gateway on behalf of the user application.
- The manager obtains the list of requested VMs from the gateway.
- This list contains a tuple of public IP/private IP addresses for each VM with Secure Shell (SSH) tunnels.

4.1.4 Global Exchange of Cloud Resources

- Cloud infrastructure providers (i.e., IaaS providers) have established data centers in multiple geographical locations to provide redundancy and ensure reliability in case of site failures.
- Amazon does not provide seamless/automatic mechanisms for scaling its hosted services across multiple geographically distributed data centers.
- This approach has many shortcomings
- First, it is difficult for cloud customers to determine in advance the best location for hosting their services as they may not know the origin of consumers of their services.
- Second, SaaS providers may not be able to meet the QoS expectations of their service consumers originating from multiple geographical locations.
- The figure the high-level components of the Melbourne group's proposed InterCloud architecture

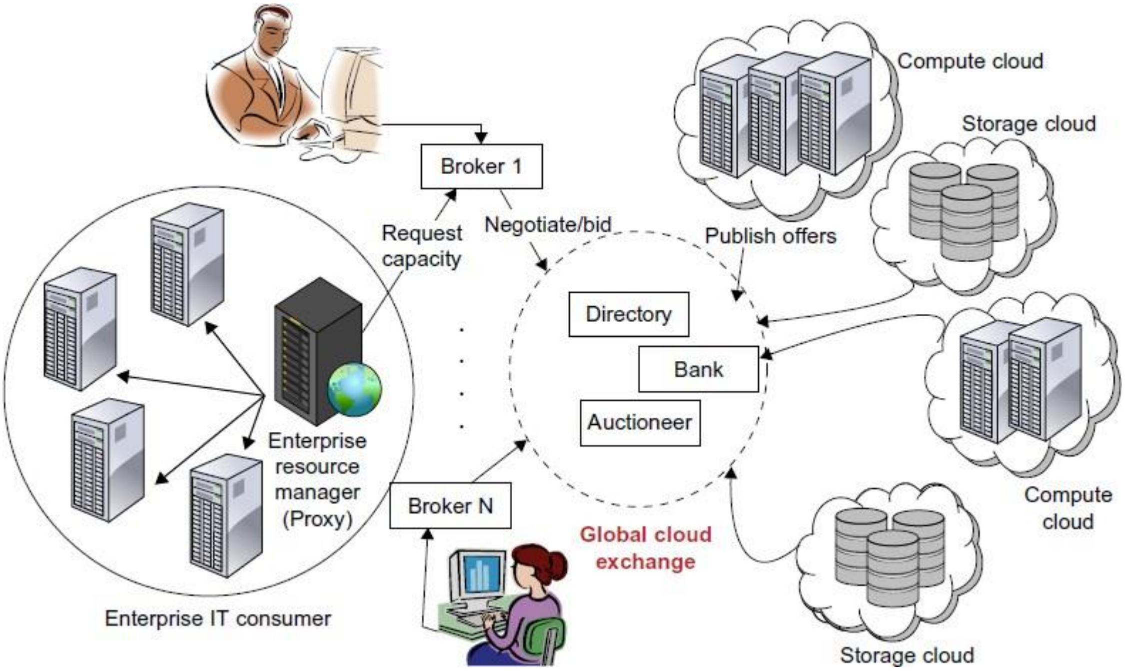


Fig: Inter-cloud exchange of cloud resources through brokering

- It is **not possible** for a cloud infrastructure provider to establish its **data centers at all possible locations** throughout the world.
- This results in **difficulty** in meeting the **QOS expectations** of their customers.

- Hence, services of **multiple cloud infrastructure** service providers are used.
- **Cloud coordinator** evaluates the available resources.
- The availability of a banking system ensures that financial transactions related to SLAs are carried out in a securely.
- By realizing InterCloud architectural principles in mechanisms in their offering, cloud providers will be able to dynamically expand or resize their provisioning capability based on sudden spikes in workload demands by leasing available computational and storage capabilities from other cloud.
- They consist of client brokering and coordinator services that support utility-driven federation of clouds:
  - application scheduling
  - resource allocation
  - migration of workloads.
- The architecture cohesively couples the administratively and topologically distributed storage and compute capabilities of clouds as part of a single resource leasing abstraction.
- The system will ease the crossdomain capability integration for on-demand, flexible, energy-efficient, and reliable access to the infrastructure based on virtualization technology
- The Cloud Exchange (CEx) acts as a market maker for bringing together service producers and consumers.
- It aggregates the infrastructure demands from application brokers and evaluates them against the available supply currently published by the cloud coordinators.
- It supports trading of cloud services based on competitive economic models such as commodity markets and auctions.
- CEx allows participants to locate providers and consumers with fitting offers.

4.2 Security

- Virtual machines from multiple organizations have to be co-located on the same physical server in order to maximize the efficiencies of virtualization.

- Cloud service providers must learn from the managed service provider (MSP) model and ensure that their customers' applications and data are secure if they hope to retain their customer base and competitiveness.
- Cloud environment should be free from abuses, cheating, hacking, viruses, rumors, and privacy and copyright violations.

4.2.1 Cloud Security Challenges

- In cloud model users lose control over physical security.
- In a public cloud, users are sharing computing resources with other companies.
- When users share the environment in the cloud, it results in data at risk of seizure (attack).
- Storage services provided by one cloud vendor may be incompatible with another vendor’s services; this results in unable to move from one to the other.
- Vendors create “sticky services”.
- Sticky services are the services which makes end user, in difficulty while transporting from one cloud vendor to another.

Example: Amazon’s “Simple Storage Service” [S3] is incompatible with IBM’s Blue Cloud, or Google, or Dell).

- Customers want their data encrypted while **data is at rest** (data stored) in the cloud vendor’s storage pool.
- Data integrity means ensuring that data is identically maintained during any operation (such as transfer, storage, or retrieval).
- Data integrity is assurance that the data is consistent and correct.
- One of the key challenges in cloud computing is data-level security.
- It is difficult for a customer to find where its data resides on a network controlled by its provider.
- Some countries have strict limits on what data about its citizens can be stored and for how long.
- Banking regulators require that customers’ financial data remain in their home country.

- Security managers will need to pay particular attention to systems that contain critical data such as corporate financial information.
- Outsourcing (giving rights to third party) loses control over data and not a good idea from a security perspective.
- Security managers have to interact with company's legal staff to ensure that appropriate contract terms are in place to protect corporate data.
- Cloud-based services will result in many mobile IT users accessing business data and services without traversing the corporate network.
- This will increase the need for enterprises to place security controls between mobile users and cloud-based services.
- Placing large amounts of sensitive data in a globally accessible cloud leavesorganizations open to large distributed threats—attackers no longer have to come onto thepremises to steal data, and they can find it all in the one "virtual" location.
- Virtualization efficiencies in the cloud require virtual machines from multiple organizations to be collocated on the same physical resources.
- Although traditional data center security still applies in the cloud environment, physical segregation and hardware-based security cannot protect against attacks between virtual machines on the same server.
- The dynamic and fluid nature of virtual machines will make it difficult to maintain the consistency of security and ensure the auditability of records.
- The ease of cloning and distribution between physical servers could result in the propagation of configuration errors and other vulnerabilities.
- Localized virtual machines and physical servers use the same operating systems as well as enterprise and web applications in a cloud server environment, increasing the threat of an attacker or malware exploiting vulnerabilities in these systems and applications remotely.
- Virtual machines are vulnerable as they move between the private cloud and the public cloud.
- Operating system and application files are on a shared physical infrastructure in a virtualized cloud environment and require system, file, and activity monitoring to provide

confidence and auditable proof to enterprise customers that their resources have not been compromised or tampered with.

- The **Intrusion Detection System(IDS)** and **Intrusion Prevention Systems(IPS)** detects malicious activity at virtual machine level.
- The co-location of multiple virtual machines increases the threat from attacker.
- If Virtual machines and physical machine use the same operating systems in a cloud environment, increases the threat from an attacker.
- A fully or partially shared cloud environment is expected to have a greater attack than own resources environment.
- Virtual machines must be self-defending.
- Cloud computing provider is incharge of customer data security and privacy.

4.2.2 Software as a Service Security (Or) Data Security (Or) Application Security (Or) Virtual Machine Security.

Cloud computing models of the future will likely combine the use of SaaS (and other XaaS's as appropriate), utility computing, and Web 2.0 collaboration technologies to leverage the Internet to satisfy their customers' needs. New business models being developed as a result of the move to cloudcomputing are creating not only new technologies and business operational processes but also newsecurity requirements and challenges

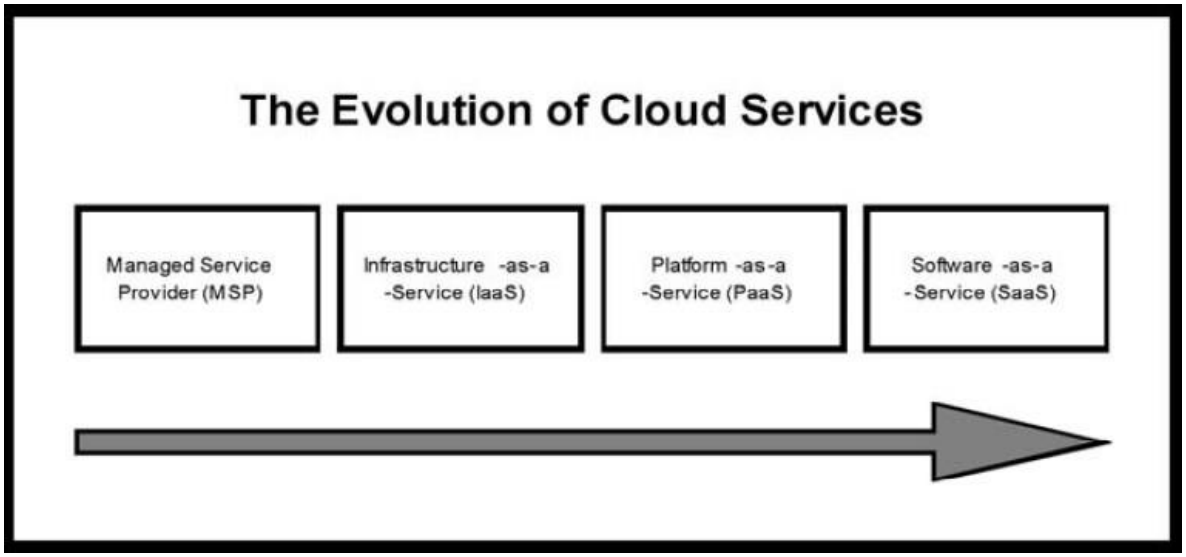

The Evolution of Cloud Services
Managed Service Provider (MSP)
Infrastructure -as-a -Service (IaaS)
Platform -as-a -Service (PaaS)
Software -as-a -Service (SaaS)

Fig: Evolution of Cloud Services

SaaS plays the dominant cloud service model and this is the area where the most critical need for security practices are required

□ Security issues that are discussed with cloud-computing vendor:

1. **Privileged user access**—Inquire about who has specialized access to data, and about the hiring and management of such administrators.
2. **Regulatory compliance**—Make sure that the vendor is willing to undergo external audits and/or security certifications.
3. **Data location**—Does the provider allow for any control over the location of data?
4. **Data segregation**—Make sure that encryption is available at all stages, and that these encryption schemes were designed and tested by experienced professionals.
5. **Recovery**—Find out what will happen to data in the case of a disaster. Do they offer complete restoration? If so, how long would that take?
6. **Investigative support**—Does the vendor have the ability to investigate any inappropriate or illegal activity?
7. **Long-term viability**—What will happen to data if the company goes out of business? How will data be returned, and in what format?

The security practices for the SaaS environment are as follows:

Security Management (People)

- One of the most important actions for a security team is to develop a formal charter for the security organization and program.
- This will foster a shared vision among the team of what security leadership is driving toward and expects, and will also foster "ownership" in the success of the collective team.
- The charter should be aligned with the strategic plan of the organization or company the security team works for.

4.2.3 Security Governance

- A security committee should be developed whose objective is to focus on providing guidance about security initiatives with business and IT strategies.
- A charter for the security team is typically one of the first deliverables from the steering committee.

- This charter must clearly define the roles and responsibilities of the security team and other groups involved in performing information security functions.
- Lack of a formalized strategy can lead to an unsustainable operating model and
- security level as it evolves.
- In addition, lack of attention to security governance can result in key needs of the business not being met, including but not limited to, risk management, security monitoring, applicationsecurity, and sales support.
- Lack of proper governance and management of duties can also result in potential security risks being left unaddressed and opportunities to improve the business being missed.
- The security team is not focused on the key security functions and activities that are critical to the business.

Cloud security governance refers to the management model that facilitates effective and efficient security management and operations in the cloud environment so that an enterprise's business targets are achieved. This model incorporates a hierarchy of executive mandates, performance expectations, operational practices, structures, and metrics that, when implemented, result in the optimization of business value for an enterprise. Cloud security governance helps answer leadership questions such as:

- Are our security investments yielding the desired returns?
- Do we know our security risks and their business impact?
- Are we progressively reducing security risks to acceptable levels?
- Have we established a security-conscious culture within the enterprise?

Strategic alignment, value delivery, risk mitigation, effective use of resources, and performance measurement are key objectives of any IT-related governance model, security included. To successfully pursue and achieve these objectives, it is important to understand the operational culture and business and customer profiles of an enterprise, so that an effective security governance model can be customized for the enterprise.

Cloud Security Governance Challenges

Whether developing a governance model from the start or having to retrofit one on existing investments in cloud, these are some of the common challenges:

Lack of senior management participation and buy-in

The lack of a senior management influenced and endorsed security policy is one of the common challenges facing cloud customers. An enterprise security policy is intended to set the executive tone, principles and expectations for security management and operations in the cloud. However, many enterprises tend to author security policies that are often laden with tactical content, and lack executive input or influence. The result of this situation is the ineffective definition and communication of executive tone and expectations for security in the cloud.

Lack of embedded management operational controls

Another common cloud security governance challenge is lack of embedded management controls into cloud security operational processes and procedures. Controls are often interpreted as an auditor's checklist or repackaged as procedures, and as a result, are not effectively embedded into security operational processes and procedures as they should be, for purposes of optimizing value and reducing day-to-day operational risks. This lack of embedded controls may result in operational risks that may not be apparent to the enterprise. For example, the security configuration of a device may be modified (change event) by a staffer without proper analysis of the business impact (control) of the modification. The net result could be the introduction of exploitable security weaknesses that may not have been apparent with this modification.

Lack of operating model, roles, and responsibilities

Many enterprises moving into the cloud environment tend to lack a formal operating model for security, or do not have strategic and tactical roles and responsibilities properly defined and operationalized. This situation stifles the effectiveness of a security management andoperational function/organization to support security in the cloud. Simply, establishing a hierarchy that includes designating an accountable official at the top, supported by a stakeholder committee, management team, operational staff, and third-party provider support (in that order) can help an enterprise to better manage and control security in the cloud, and protect associated investments in accordance with enterprise business goals.

Lack of metrics for measuring performance and risk

Another major challenge for cloud customers is the lack of defined metrics to measure security performance and risks – a problem that also stifles executive visibility into the real security risks in the cloud. This challenge is directly attributable to the combination of other challenges discussed above. For example, a metric that quantitatively measures the number of exploitable security vulnerabilities on host devices in the cloud over time can be leveraged as an indicator of risk in the host device environment. Similarly, a metric that measures the number of user-reported security incidents over a given period can be leveraged as a performance indicator of staff awareness and training efforts. Metrics enable executive visibility into the extent to which security tone and expectations (per established policy) are being met within the enterprise and support prompt decision-making in reducing risks or rewarding performance as appropriate.

The challenges described above clearly highlight the need for cloud customers to establish a framework to effectively manage and support security in cloud management, so that the pursuit of business targets are not potentially compromised. Unless tone and expectations for cloud security are established (via an enterprise policy) to drive operational processes and procedures with embedded management controls, it is very difficult to determine or evaluate business value, performance, resource effectiveness, and risks regarding security operations in the cloud. Cloud security governance facilitates the institution of a model that helps enterprises explicitly address the challenges described above.

Key Objectives for Cloud Security Governance

Building a cloud security governance model for an enterprise requires strategic-level security management competencies in combination with the use of appropriate security standards and frameworks (e.g., NIST, ISO, CSA) and the adoption of a governance framework (e.g., COBIT). The first step is to visualize the overall governance structure, inherent components, and to direct its effective design and implementation. The use of appropriate security standards and frameworks allow for a minimum standard of security controls to be implemented in the cloud, while also meeting customer and regulatory compliance obligations where applicable. A governance framework provides referential guidance and best practices for establishing thegovernance model for security in the cloud. The following represents key objectives to pursue in establishing a governance model for security in the cloud. These objectives assume that appropriate security standards and a governance framework have been chosen based on the enterprise's business targets, customer profile, and obligations for protecting data and other information assets in the cloud environment.

1. **Strategic Alignment**

   Enterprises should mandate that security investments, services, and projects in the cloud are executed to achieve established business goals (e.g., market competitiveness, financial, or operationalperformance).

2. **Value Delivery**

   Enterprises should define, operationalize, and maintain an appropriate security function/organization with appropriate strategic and tactical representation, and charged with the responsibility to maximize the business value (Key Goal Indicators, ROI) from the pursuit of security initiatives in the cloud.

3. **Risk Mitigation**

   Security initiatives in the cloud should be subject to measurements that gauge effectiveness in mitigating risk to the enterprise (Key Risk Indicators). These initiatives should also yield results that progressively demonstrate a reduction in these risks over time.

4. **Effective Use of Resources**

   It is important for enterprises to establish a practical operating model for managing and performing security operations in the cloud, including the proper definition and operationalization of due processes, the institution of appropriate roles and responsibilities, and use of relevant tools for overall efficiency and effectiveness.

5. **Sustained Performance**

   Security initiatives in the cloud should be measurable in terms of performance, value and risk to the enterprise (Key Performance Indicators, Key Risk Indicators), and yield results that demonstrate attainment of desired targets (Key Goal Indicators) over time.

Risk Management

- Effective risk management entails identification of technology assets; identification of data and its links to business processes, applications, and data stores; and assignment of ownership and custodial responsibilities.
- Actions should also include maintaining a repository of information assets
- A risk assessment process should be created that allocates security resources related to business continuity.

Risk Assessment

- Security risk assessment is critical to helping the information security organization make informed decisions when balancing the dueling priorities of business utility and protection of assets.
- Lack of attention to completing formalized risk assessments can contribute to an increase in information security audit findings, can jeopardize certification goals, and can lead to

inefficient and ineffective selection of security controls that may not adequately mitigate information security risks to an acceptable level.

Security Portfolio(selection) Management

- Security portfolio management ensures efficient and effective operation of any information.

Security Awareness

- Not providing proper awareness and training to the people who may need them can expose the company to a variety of security risks

Policies, Standards, and Guidelines

- Policies, standards, and guidelines are developed that can ensure consistency of performance.

Secure Software Development Life Cycle (SecSDLC)

- The SecSDLC involves identifying specific threats and the risks. The SDLC consists of six phases

Phase 1.Investigation:

-Define project goals, and document them.

Phase 2.Analysis:

-Analyze current threats and perform risk analysis.

Phase 3.Logical design:

-Develop a security blueprint(plan) and business responses to disaster.

Phase 4.Physical design:

-Select technologies to support the security blueprint(plan).

Phase 5.Implementation:

- Buy or develop security solutions.

Phase 6.Maintenance:

-Constantly monitor, test, modify, update, and repair to respond to changing threats.

Security Monitoring and Incident Response

- Centralized security management systems should be used to provide notification of security vulnerabilities and to monitor systems continuously.

Business Continuity Plan

Business continuity plan, ensures uninterrupted operations of business.

Forensics

Forensics includes recording and analyzing events to determine the nature and source of information abuse, security attacks, and other such incidents.

Security Architecture Design

A security architecture framework should be established with the following consideration

1. Authentication
2. Authorization
3. Availability
4. Confidentiality
5. Integrity
6. Privacy

Vulnerability Assessment

- Vulnerability assessment classifies network assets to more efficiently prioritize vulnerability-mitigation programs, such as patching and system upgrading.
- It measures the effectiveness of risk mitigation by setting goals of reduced vulnerability exposure and faster mitigation

Password Assurance Testing

- If the SaaS security team or its customers want to periodically test password strength by running
- password "crackers," they can use cloud computing to decrease crack time and pay only for what they use.
-

Security Images:

- Virtualization-based cloud computing provides the ability to create "Gold image" VM secure builds and to clone multiple copies.
- Gold image VMs also provide the ability to keep security up to date and reduce exposure by patching offline.

Data Privacy

- Depending on the size of the organization and the scale of operations, either an individual or a team should be assigned and given responsibility for maintaining privacy.
- A member of the security team who is responsible for privacy or security compliance team should collaborate with the company legal team to **address data privacy issues and concerns.**

- **Hiring a consultant** in privacy area, will ensure that your organization is prepared to meet the data privacy demands of its customers and regulators.

Data Governance

The data governance framework should include:

_ Data inventory

_ Data classification

_ Data analysis (business intelligence)

_ Data protection

_ Data privacy

_ Data retention/recovery/discovery

_ Data destruction

Data Security

The challenge in cloud computing is data-level security.

Security to data is given by

□ Encrypting the data

□ Permitting only specified users to access the data.

□ Restricting the data not to cross the countries border.

For example, with data-level security, the enterprise can specify that this data is not allowed to go outside of the India.

Application Security

□ This is collaborative effort between the security and product development team.

□ Application security processes

o Secure coding guidelines

o Training

o Testing scripts

o Tools

□ Penetration Testing is done to a System or application.

□ Penetration Testing is defined as a type of Security Testing used to test the **insecure areas of the system or application**.

- The goal of this testing is to **find all the security vulnerabilities** that are present in the system being tested.
- SaaS providers should secure their web applications by following **Open Web Application Security Project (OWASP) guidelines** for secure application development, **by locking down ports** and unnecessary commands

4.3 Virtual Machine Security

In the cloud environment, physical servers are consolidated (combined) to multiple virtualmachine instances.

Following are deployed on virtual machines to ensure security

- Firewalls
- Intrusion detection and prevention
- Integrity monitoring
- Log inspection

Virtual servers have security requirements identical to those of physical servers. The same applies to the applications and services they host. Virtualization provides security benefits: each virtual machine has a private security context, potentially with separate authentication and authorization rules, and with separate process, name and file system spaces. Deploying applications onto separate virtual machines provides better security control compared to running multiple applications on the same host operating system: penetrating one virtual machine's OS doesn't necessarily compromise workload and data residing in other virtual machines. Nonetheless, some practices should be kept in mind to prevent virtualization from introducing security vulnerabilities.

One aspect is physical security. Virtual infrastructure is not as 'visible' as physical infrastructure: there is no sticky label on a virtual machine to indicate its purpose and security classification. If a datacenter identifies servers with extremely high security requirements, and physically isolates them in a locked room or cage to prevent tampering or theft of data, then the physical machines hosting their virtualized workloads should be isolated in a similar way. Even without secured areas, many institutions keep workloads of different security classes on different servers. Those same isolation rules apply for virtual machines. Care should be taken to ensure that the protected virtual machines are not migrated to a server in a less secure location. In the context of Oracle VM, this implies maintaining separate server pools, each with their own group of servers.

These rules of isolation should also be applied to networking: there are no color coded network cables to help staff identify and isolate different routes, segments and types network traffic to and from virtual machines or between them. There are no visual indicators that help ensure that application, management, and backup traffic are kept separate. Rather than plug network cables into different physical interfaces and switches, the Oracle VM administrator mustensure that the virtual network interfaces are connected to separate virtual networks. Specifically,use VLANs to isolate virtual machines from one another, and assign virtual networks for virtual machine traffic to different physical interfaces from those used for management, storage or backup. These can all be controled from the Oracle VM Manager user interface. Ensure that secure live migration is selected to guarantee that virtual machine memory data is not sent acrossthe wire unencrypted.

Additional care must be given to virtual machine disk images. In most cases the virtual disks are made available over the network for migration and failover purposes. In many cases they are files, which could easily be copied and stolen if the security of network storage is compromised. Therefore it is essential to lock down the NAS or SAN environments and prevent unauthorized access. An intruder with root access to a workstation on the storage network could mount storage assets and copy or alter their contents. Use a separate network for transmission between the storage servers and the Oracle VM hosts to ensure its traffic is not made public and subject to being snooped. Make sure that unauthorized individuals are not permitted to log into the Oracle VM Servers, as that would give them access to the guests' virtual disk images, and potentially much more.

All of these steps require controlling access to the Oracle VM Manager and Oracle VM Server domain 0 instances. Network access to these hosts should be on a private network, and the user accounts able to log into any of the servers in the Oracle VM environment should be rigorously controlled, and limited to the smallest possible number of individuals.

4.4 Identity and access management architecture(IAM)

Basic concept and definitions of IAM functions for any service:

Authentication – is a process of verifying the identity of a user or a system.Authentication usually connotes a more roburst form of identification. In some use
cases such as service – to- service interaction, authentication involves verifying the
network service.

Authorization – is a process of determining the privileges the user or system is
entitled to once the identity is established. Authorization usually follows theauthentication step and is used to determine whether the user or service has the
necessary privileges to perform certain operations.

Auditing – Auditing entails the process of review and examination ofauthentication,authorization records and activities to determine the adequacy of IAMsystem controls, to verify complaints with established security policies and procedure,to detect breaches in security services and to recommend any changes that areindicated for counter measures

IAM Architecture and Practice

IAM is not a monolithic solution that can be easily deployed to gain capabilitiesimmediately. It is as much an aspect of architecture as it is acollection of technology components, processes, and standard practices. Standardenterprise IAM architecture encompasses several layers of technology, services, andprocesses. At the core of the deployment architecture is a directory service (such as
LDAP or Active Directory) that acts as a repository for the identity, credential, and user attributes of the organization's user pool. The directory interacts with IAM technology components such as authentication, user management, provisioning, and federation services that support the standard IAM practice and processes within the organization.

The IAM processes to support the business can be broadly categorized as follows:

User management: Activities for the effective governance and management of identity life cycles

Authentication management: Activities for the effective governance and management of the process for determining that an entity is who or what it claims tobe.

Authorization management: Activities for the effective governance and management of the process for determining entitlement rights that decide what resources an entity is permitted to access in accordance with the organization's policies.

Access management: Enforcement of policies for access control in response to a request from an entity (user, services) wanting to access an IT resource within the organization.

Data management and provisioning: Propagation of identity and data for authorization to IT resources via automated or manual processes.

Monitoring and auditing: Monitoring, auditing, and reporting compliance by users regarding access to resources within the organization based on the defined policies.

IAM processes support the following operational activities:

Provisioning: Provisioning can be thought of as a combination of the duties of the
human resources and IT departments, where users are given access to data repositories orsystems, applications, and databases based on a unique user identity. Deprovisioning works in the opposite manner, resulting in the deletion or deactivation of an identity or of privileges assigned to the user identity.

Credential and attribute management: These processes are designed to manage the life cycle of credentials and user attributes— create, issue, manage, revoke—to inappropriate account use. Credentials are usually bound to an individual and are verified during the authentication process. The processes include provisioning of attributes, static (e.g., standard text password) and dynamic (e.g., one-time password) credentials that comply with a password standard (e.g., passwords resistant to dictionary attacks), handling password expiration, encryption management of credentials during transit and at rest, and access policies of user attributes (privacy and handling of attributes for various regulatory reasons).Minimize the business risk associated with identityimpersonation

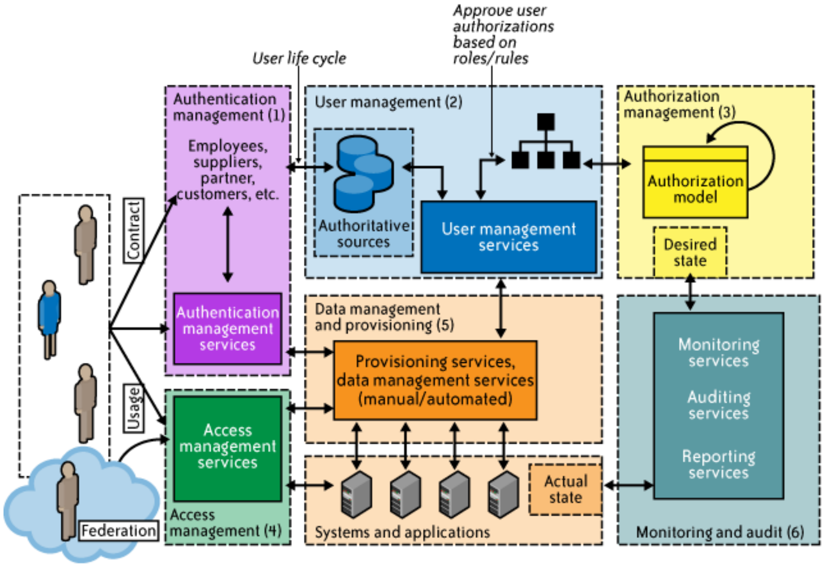

Approve user authorizations based on roles/rules
User life cycle
Authentication management (1)
Employees, suppliers, partner, customers, etc.
Authentication management services
User management (2)
Authoritative sources
User management services
Authorization management (3)
Authorization model
Desired state
Contract
Usage
Federation
Access management services
Access management (4)
Data management and provisioning (5)
Provisioning services, data management services (manual/automated)
Systems and applications
Actual state
Monitoring services
Auditing services
Reporting services
Monitoring and audit (6)

Figure 5.7 Enterprise IAM functional architecture

Entitlement management: Entitlements are also referred to as authorization policies. The processes in this domain address the provisioning and deprovisioning of privileges needed for the user to access resources including systems, applications, and databases. Proper entitlement management ensures that users are assigned only the required privileges.

Compliance management: This process implies that access rights and privileges are monitored and tracked to ensure the security of an enterprise's resources. The process also helps auditors verify compliance to various internal access control policies, and standards that include practices such as segregation of duties, access monitoring, periodic auditing, and reporting. An example is a user certification process that allows application owners to certify that only authorized users have the privileges necessary to access business-sensitive information.

Identity federation management: Federation is the process of managing the trust relationships established beyond the internal network boundaries or administrative domain boundaries among distinct organizations. A federation is an association of organizations that come together to exchange information about their users and resources to enable collaborations and transactions.

Centralization of authentication (authN) and authorization (authZ): A central authentication and authorization infrastructure alleviates the need for application developers to build custom authentication and authorization features into their applications. Furthermore, it promotes a loose coupling architecture where applications become agnostic to the authentication methods and policies. This approach is also called an ―externalization of authN and authZ from applications

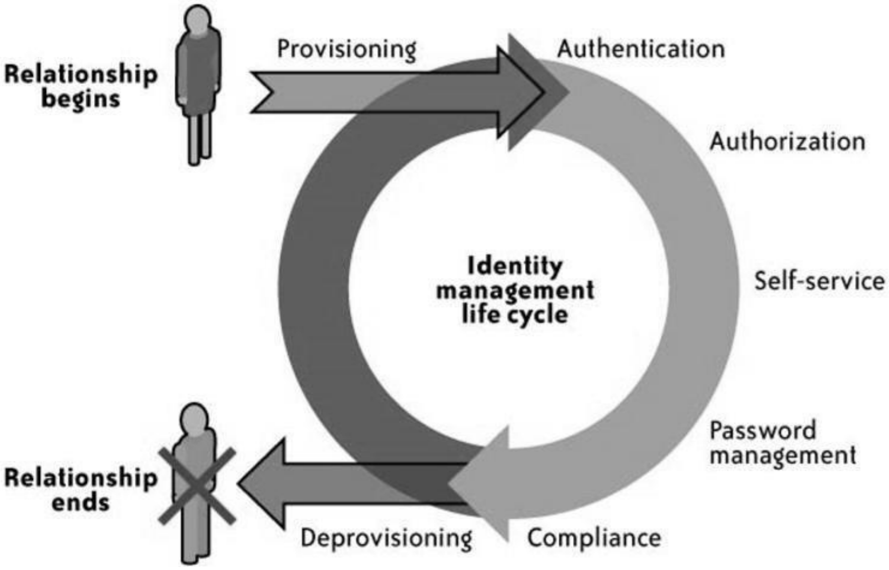


Figure 5.8 Identity Life cycle

IAM Standards and Specifications for Organisations

The following IAM standards and specifications will help organizations implement effective and efficient user access management practices and processes inthe cloud. These sections are ordered by four major challenges in user and access management faced by cloud users:

1. How can I avoid duplication of identity, attributes, and credentials and provide a single sign- on user experience for my users? SAML.

2. How can I automatically provision user accounts with cloud services and automate the process of provisoning and deprovisioning? SPML.

<u>IAM Practices in the Cloud</u>

When compared to the traditional applications deployment model within the enterprise, IAM practices in the cloud are still evolving. In the current state of IAM technology, standards support by CSPs (SaaS, PaaS, and IaaS) is not consistent across providers. Although large providers such as Google, Microsoft, and Salesforce.com seem to demonstrate basic IAM

capabilities, our assessment is that they still fall short of enterprise IAM requirements formanaging regulatory, privacy, and data protection requirements. The maturity model takes into account the dynamic nature of IAM users, systems, and applications in the cloud and
addresses the four key components of the IAM automation process:

- User Management, New Users
- User Management, User Modifications
- Authentication Management
- Authorization Management

IAM practices and processes are applicable to cloud services; they need to be adjusted to the cloud environment. Broadly speaking, user management functions in the cloud can be categorized as follows:

- Cloud identity administration, Federation or SSO
- Authorization management
- Compliance management

Cloud Identity Administration: Cloud identity administrative functions should focus on life cycle management of user identities in the cloud—provisioning, deprovisioning, identity federation, SSO, password or credentials management, profile management, and administrative management. Organizations that are not capable of supporting federation should explore cloud-based identity management services. This new breed of services usually synchronizes an organization's internal directories with its directory (usually multitenant) and acts as a proxy IdP for the organization.

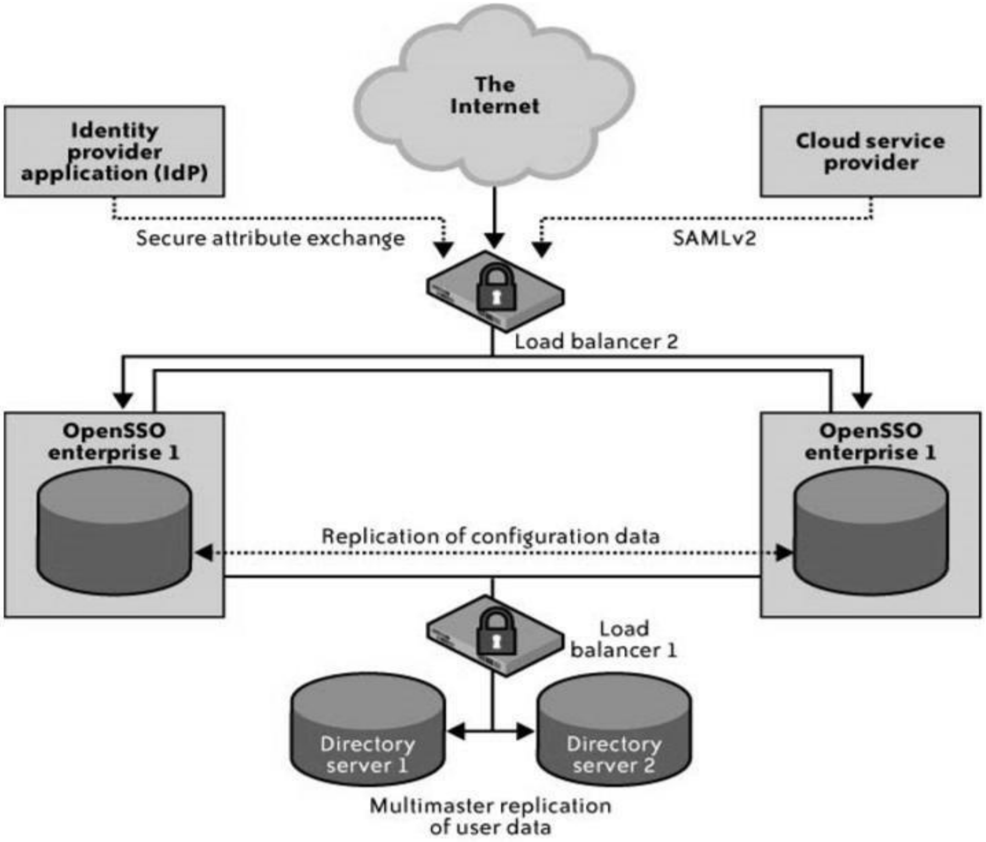
Federated Identity (SSO): Organizations planning to implement identity federation that enables SSO for users can take one of the following two paths (architectures):

- Implement an enterprise IdP within an organization perimeter.
- Integrate with a trusted cloud-based identity management service provider.

Both architectures have pros and cons.

Enterprise identity provider: In this architecture, cloud services will delegate authentication to an organization's IdP. In this delegated authentication architecture, the organization federates identities within a trusted circle of CSP domains. A circle of trust can be created with all the domains that are authorized to delegate authentication to the IdP. In this deployment architecture, where the organization will provide and support an IdP, greater control can be exercised over user identities, attributes, credentials, and policies for authenticating and authorizing users to a cloud service.

IdP deployment architecture.



4.5 Security standards

Security standards define the processes, procedures, and practices necessary for implementing a security program. These standards also apply to cloud-related IT activities and include specific steps that should be taken to ensure a secure environment is maintained that provides privacy and security of confidential information in a cloud environment. Security standards are based on a set of key principles intended to protect this type of trusted environment. Messaging standards, especially for security in the cloud, must also include nearly all the same considerations as any other IT security endeavor.

Security (SAML ,OAuth, OpenID, SSL/TLS)

A basic philosophy of security is to have layers of defense, a concept known as *defense in depth*. This means having overlapping systems designed to provide security even if one system fails. An example is a firewall working in conjunction with an intrusion-detection system (IDS). Defense in depth provides security because there is no single point of failure and no single-entry vector at which an attack can occur. No single security system is a solution by itself, so it is far better to secure all systems. This type of layered security is precisely what we are seeing develop in cloud computing. Traditionally, security was implemented at the endpoints, where the user controlled access. An organization had no choice except to put firewalls, IDSs, and antivirus software inside its own network. Today, with the advent of managed security services offered by cloud providers, additional security can be provided inside the cloud.

4.5.1 Security Assertion Markup Language (SAML)

SAML is an XML-based standard for communicating authentication, authorization, and attribute information among online partners. It allows businesses to securely send assertions between partner organizations regarding the identity and entitlements of a principal. The Organization for the Advancement of Structured Information Standards (OASIS) Security Services Technical Committee is in charge of defining, enhancing, and maintaining the SAML specifications.

SAML is built on a number of existing standards, namely, SOAP, HTTP, and XML. SAML relies on HTTP as its communications protocol and specifies the use of SOAP (currently, version 1.1). Most SAML transactions are expressed in a standardized form of XML. SAML assertions and protocols are specified using XML schema. Both SAML 1.1 and SAML 2.0 use digital signatures (based on the XML Signature standard) for authentication and message integrity. XML encryption is supported in SAML 2.0, though SAML 1.1 does not have encryption capabilities. SAML defines XML-based assertions and protocols, bindings, and profiles. The term SAML Core refers to the general syntax and semantics of SAML assertions as well as the protocol used to request and transmit those assertions from one system entity to another. SAML protocol refers to what is transmitted, not how it is transmitted. A SAML binding determines how SAML requests and responses map to standard messaging protocols. An important (synchronous) binding is the SAML SOAP binding.

SAML standardizes queries for, and responses that contain, user authentication, entitlements, and attribute information in an XML format. This format can then be used to request security information about a principal from a SAML authority. A SAML authority, sometimes called the asserting party, is a platform or application that can relay security information. The relying party (or assertion consumer or requesting party) is a partner site that receives the security information.

The exchanged information deals with a subject's authentication status, access authorization, and attribute information. A subject is an entity in a particular domain. A person identified by an email address is a subject, as might be a printer.

SAML assertions are usually transferred from identity providers to service providers. Assertions contain statements that service providers use to make access control decisions. Three types of statements are provided by SAML: authentication statements, attribute statements, and authorization decision statements. SAML assertions contain a packet of security information in this form:

```
<saml:Assertion A...>
<Authentication>
...
</Authentication>
<Attribute>
...
</Attribute>
<Authorization>
...
</Authorization>
</saml:Assertion A>
```

The assertion shown above is interpreted as follows:

Assertion A, issued at time T by issuer I, regarding subject
S, provided conditions C are valid.

Authentication statements assert to a service provider that the principal did indeed authenticate with an identity provider at a particular time using a particular method of authentication. Other information about the authenticated principal (called the authentication context) may be disclosed in an authentication statement. An attribute statement asserts that a subject is associated with certain attributes. An attribute is simply a name-value pair. Relying parties use attributes to make access control decisions. An authorization decision statement asserts that a subject is permitted to perform action A on resource R given evidence E. The expressiveness of authorization decision statements in SAML is intentionally limited.

A SAML protocol describes how certain SAML elements (including assertions) are packaged within SAML request and response elements. It provides processing rules that SAML entities must adhere to when using these elements. Generally, a SAML protocol is a simple request-response protocol. The most important type of SAML protocol request is a query. A service provider makes a query directly to anidentity provider over a secure back channel. For this reason, query messages are typically bound to SOAP. Corresponding to the three types of statements, there are three types of SAML queries: the authentication query, the attribute query, and the authorization decision query. Of these, the attribute query is perhaps most important. The result of an attribute query is a SAML response containing anassertion, which itself contains an attribute statement.

4.5.2 Open Authentication (OAuth)

OAuth is an open protocol, initiated by Blaine Cook and Chris Messina, to allow secure API authorization in a simple, standardized method for various types of web applications. Cook and Messina had concluded that there were no open standards for API access delegation. The OAuth discussion group was created in April 2007, for the small group of implementers to write the draft proposal for an open protocol. DeWitt Clinton of Google learned of the OAuth project and expressed interest in supporting the effort. In July 2007 the team drafted an initial specification, and it was released in October of the same year. OAuth is a method for publishing and interacting with protected data. For developers, OAuth provides

users access to their data while protecting account credentials. OAuth allows users to grant access to their information, which is shared by the service provider and consumers without sharing all of their identity. The Core designation is used to stress that this is the baseline, and other extensions and protocols can build on it. By design, OAuth Core 1.0 does not provide many desired features (e.g., automated discovery of endpoints, language support, support for XML-RPC and SOAP, standard definition of resource access, OpenID integration, signing algorithms, etc.). This intentional lack of feature support is viewed by the authors as a significant benefit. The Core deals with fundamental aspects of the protocol, namely, to establish a mechanism for exchanging a user name and password for a token with defined rights and to provide tools to protect the token. . In fact, OAuth by itself *provides no privacy at all* and depends on other protocols such as SSL to accomplish that.

4.5.3OpenID

OpenID is an open, decentralized standard for user authentication and access control that allows users to log onto many services using the same digital identity. It is a single-sign-on (SSO) method of access control. As such, it replaces the common log-in process (i.e., a log-in name and a password) by allowing users to log in once and gain access to resources across participating systems. The original OpenID authentication protocol was developed in May 2005 by Brad Fitzpatrick, creator of the popular community web site Live-Journal. In late June 2005, discussions began between OpenID developers and other developers from an enterprise software company named Net-Mesh. These discussions led to further collaboration on interoperability between OpenID and NetMesh's similar Light-Weight Identity (LID) protocol. The direct result of the collaboration was the Yadis discovery protocol, which was announced on October 24, 2005.

The Yadis specification provides a general-purpose identifier for a person and any other entity, which canbe used with a variety of services. It provides a syntax for a resource description document identifying services available using that identifier and an interpretation of the elements of that document. Yadis discovery protocol is used for obtaining a resource description document, given that identifier. Together these enable coexistence and interoperability of a rich variety of services using a single identifier. The identifier uses a standard syntax and a well- established namespace and requires no additional namespace administration infrastructure.

An OpenID is in the form of a unique URL and is authenticated by the entity hosting the OpenID URL.The OpenID protocol does not rely on a central authority to authenticate a user's identity. Neither the OpenID protocol nor any web sites requiring identification can mandate that a specific type of authentication be used; nonstandard forms of authentication such as smart cards, biometrics, or ordinary passwords are allowed. A typical scenario for using OpenID might be something like this: A user visits a web site that displays an OpenID log-in form somewhere on the page. Unlike a typical log-in form, which has fields for user name and password, the OpenID log-in form has only one field for the OpenID identifier (which is an OpenID URL). This form is connected to an implementation of an OpenID client library.

A user will have previously registered an OpenID identifier with an OpenID identity provider. The user types this OpenID identifier into the OpenID log-in form. The relying party then requests the web page located at that URL and reads an HTML link tag to discover the identity provider service URL. With OpenID 2.0, the client discovers the identity provider service URL by requesting the XRDS document (also called the Yadis document) with the content type **application/xrds+xml,** which may be available at the target URL but is always available for a target XRI.

There are two modes by which the relying party can communicate with the identity provider: **checkid_immediate** and **checkid_setup.** In **checkid_immediate,** the relying party requests that the provider not interact with the user. All communication is relayed through the user's browser without explicitly notifying the user. In **checkid_setup,** the user communicates with the provider server directly using the same web browser as is used to access the relying party site. The second option is more popular on the web.

To start a session, the relying party and the identity provider establish a shared secret— referenced by an associate handle—which the relying party then stores. Using **checkid_setup,** the relying party redirects the user's web browser to the identity provider so that the user can authenticate with the provider. The method of authentication varies, but typically, an OpenID identity provider prompts the user for a password, then asks whether the user trusts the relying party web site to receive his or her credentials and identity details. If the user declines the identity provider's request to trust the relying party web site, the browser is redirected to the relying party with a message indicating that authentication was rejected.

The site in turn refuses to authenticate the user. If the user accepts the identity provider's request to trust the relying party web site, the browser is redirected to the designated return page on the relying party web site along with the user's credentials. That relying party must then confirm that the credentials really came from the identity provider. If they had previouslyestablished a shared secret, the relying party can validate the shared secret received with the credentials against the one previously stored. In this case, the relying party is considered to be stateful, because it stores the shared secret between sessions (a process sometimes referred to as persistence). In comparison, a stateless relying party must make background requests using the **check_authentication** method to be sure that the data came from the identity provider.

4.5.4 SSL/TLS

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographically secure protocols designed to provide security and data integrity for communications over TCP/IP. TLS and SSL encrypt the segments of network connections at the transport layer. Several versions of the protocols are in general use in web browsers, email, instant messaging, and voice-over-IP. TLS is an IETF standard protocol which was last updated in RFC 5246.

The TLS protocol allows client/server applications to communicate across a network in a way specifically designed to prevent eavesdropping, tampering, and message forgery. TLS provides endpoint authentication and data confidentiality by using cryptography. TLS authentication is one-way—the server is authenticated, because the client already knows the server's identity. In this case, the client remains unauthenticated. At the browser level, this means that the browser has validated the server's certificate—more specifically, it has checked the digital signatures of the server certificate's issuing chain ofCertification Authorities (CAs).

Validation does not identify the server to the end user. For true identification, the end user must verify the identification information contained in the server's certificate (and, indeed, its whole issuing CA chain).This is the only way for the end user to know the "identity" of the server, and this is the only way identity can be securely established, verifying that the URL, name, or address that is being used is specified inthe server's certificate. Malicious web sites cannot use the valid certificate of another web site becausethey have no means to encrypt the transmission in a way that it can be decrypted with the valid certificate.

Since only a trusted CA can embed a URL in the certificate, this ensures that checking the apparent URL with the URL specified in the certificate is an acceptable way of identifying the site.TLS also supports a more secure bilateral connection mode whereby both ends of theconnection can be assured that they are communicating with whom they believe they are connected. This is known asmutual (assured) authentication. Mutual authentication requires the TLS client-side to also maintain a certificate.

TLS involves three basic phases:

1. Peer negotiation for algorithm support
2. Key exchange and authentication
3. Symmetric cipher encryption and message authentication

During the first phase, the client and server negotiate cipher suites, which determine which ciphers are used; makes a decision on the key exchange and authentication algorithms to be used; and determines the message authentication codes. The key exchange and authentication algorithms are typically public key algorithms. The message authentication codes are made up from cryptographic hash functions. Once these decisions are made, data transfer may begin.