Cloud Delivery Models: The Cloud Provider Perspective

This section explores the architecture and administration of IaaS, PaaS, and SaaS cloud delivery models from the point of view of the cloud provider. The integration and management of these cloud-based environments as part of greater environments and how they can relate to different technologies and cloud mechanism combinations are examined.

Building IaaS Environments

The virtual server and cloud storage device mechanisms represent the two most fundamental IT resources that are delivered as part of a standard rapid provisioning architecture within IaaS environments. They are offered in various standardized configurations that are defined by the following properties:

- operating system
- primary memory capacity
- processing capacity
- virtualized storage capacity

Memory and virtualized storage capacity is usually allocated with increments of 1 GB to simplify the provisioning of underlying physical IT resources. When limiting cloud consumer access to virtualized environments, IaaS offerings are preemptively assembled by cloud providers via virtual server images that capture the pre-defined configurations. Some cloud providers may offer cloud consumers direct administrative access to physical IT resources, in which case the bare-metal provisioning architecture may come into play.

Snapshots can be taken of a virtual server to record its current state, memory, and configuration of a virtualized IaaS environment for backup and replication purposes, in support of horizontal and vertical scaling requirements. For example, a virtual server can use its snapshot to become reinitialized in another hosting environment after its capacity has been increased to allow for vertical scaling. The snapshot can alternatively be used to duplicate a virtual server. The management of custom virtual server images is a vital feature that is provided via the remote administration system mechanism. Most cloud providers also support importing and exporting options for custom-built virtual server images in both proprietary and standard formats.

Data Centers

Cloud providers can offer IaaS-based IT resources from multiple geographically diverse data centers, which provides the following primary benefits:

- Multiple data centers can be linked together for increased resiliency. Each data center is placed in a different location to lower the chances of a single failure forcing all of the data centers to go offline simultaneously.
- Connected through high-speed communications networks with low latency, data centers can perform load balancing, IT resource backup and replication, and increase storage capacity, while improving availability and reliability. Having multiple data centers spread over a greater area further reduces network latency.
- Data centers that are deployed in different countries make access to IT resources more convenient for cloud consumers that are constricted by legal and regulatory requirements.

When an IaaS environment is used to provide cloud consumers with virtualized network environments, each cloud consumer is segregated into a tenant environment that isolates IT resources from the rest of the cloud through the Internet. VLANs and network access control software collaboratively realize the corresponding logical network perimeters.

Scalability and Reliability

Within IaaS environments, cloud providers can automatically provision virtual servers via the dynamic vertical scaling type of the dynamic scalability architecture. This can be performed through the VIM, as long as the host physical servers have sufficient capacity. The VIM can scale virtual servers out using resource replication as part of a resource pool architecture, if a given physical server has insufficient capacity to support vertical scaling. The load balancer mechanism, as part of a workload distribution architecture, can be used to distribute the workload among IT resources in a pool to complete the horizontal scaling process.

Manual scalability requires the cloud consumer to interact with a usage and administration program to explicitly request IT resource scaling. In contrast, automatic scalability requires the automated scaling listener to monitor the workload and reactively scale the resource capacity. This mechanism typically acts as a monitoring agent that tracks IT resource usage in order to notify the resource management system when capacity has been exceeded.

Replicated IT resources can be arranged in high-availability configuration that forms a failover system for implementation via standard VIM features. Alternatively, a high-availability/high-performance resource cluster can be created at the physical or virtual server level, or both simultaneously. The multipath resource access architecture is commonly employed to enhance reliability via the use of redundant access paths, and some cloud providers further offer the provisioning of dedicated IT resources via the resource reservation architecture.

Monitoring

Cloud usage monitors in an IaaS environment can be implemented using the VIM or specialized monitoring tools that directly comprise and/or interface with the virtualization platform. Several common capabilities of the IaaS platform involve monitoring:

- ***Virtual Server Lifecycles*** - Recording and tracking uptime periods and the allocation of IT resources, for pay-per-use monitors and time-based billing purposes.
- ***Data Storage*** - Tracking and assigning the allocation of storage capacity to cloud storage devices on virtual servers, for pay-per-use monitors that record storage usage for billing purposes.
- ***Network Traffic*** - For pay-per-use monitors that measure inbound and outbound network usage and SLA monitors that track QoS metrics, such as response times and network losses.
- ***Failure Conditions*** - For SLA monitors that track IT resource and QoS metrics to provide warning in times of failure.
- ***Event Triggers*** - For audit monitors that appraise and evaluate the regulatory compliance of select IT resources.

Monitoring architectures within IaaS environments typically involve service agents that communicate directly with backend management systems.

Security

Cloud security mechanisms that are relevant for securing IaaS environments include:

- encryption, hashing, digital signature, and PKI mechanisms for overall protection of data transmission
- IAM and SSO mechanisms for accessing services and interfaces in security systems that rely on user identification, authentication, and authorization capabilities
- cloud-based security groups for isolating virtual environments through hypervisors and network segments via network management software
- hardened virtual server images for internal and externally available virtual server environments
- various cloud usage monitors to track provisioned virtual IT resources to detect abnormal usage patterns.

Equipping PaaS Environments

PaaS environments typically need to be outfitted with a selection of application development and deployment platforms in order to accommodate different programming models, languages, and frameworks. A separate ready-made environment is usually created for each programming stack that contains the necessary software to run applications specifically developed for the platform.

Each platform is accompanied by a matching SDK and IDE, which can be custom-built or enabled by IDE plugins supplied by the cloud provider. IDE toolkits can simulate the cloud runtime locally within the PaaS environment and usually include executable application servers. The security restrictions that are inherent to the runtime are also simulated in the development environment, including checks for unauthorized attempts to access system IT resources.

Cloud providers often offer a resource management system mechanism that is customized for the PaaS platform so that cloud consumers can create and control customized virtual server images with ready-made environments. This mechanism also provides features specific to the PaaS platform, such as managing deployed applications and configuring multitenancy. Cloud providers further rely on a variation of the rapid provisioning architecture known as platform provisioning, which is designed specifically to provision ready-made environments.

Scalability and Reliability

The scalability requirements of cloud services and applications that are deployed within PaaS environments are generally addressed via dynamic scalability and workload distribution architectures that rely on the use of native automated scaling listeners and load balancers. The resource pooling architecture is further utilized to provision IT resources from resource pools made available to multiple cloud consumers.

Cloud providers can evaluate network traffic and server-side connection usage against the instance's workload, when determining how to scale an overloaded application as per parameters and cost limitations provided by the cloud consumer. Alternatively, cloud consumers can configure the application designs to customize the incorporation of available mechanisms themselves.

The reliability of ready-made environments and hosted cloud services and applications can be supported with standard failover system mechanisms (Figure 142), as well as the non-disruptive service relocation

architecture, so as to shield cloud consumers from failover conditions. The resource reservation architecture may also be in place to offer exclusive access to PaaS-based IT resources. As with other IT resources, ready-made environments can also span multiple data centers and geographical regions to further increase availability and resiliency

Monitoring

Specialized cloud usage monitors in PaaS environments are used to monitor the following:

- ***Ready-Made Environment Instances*** - The applications of these instances are recorded by pay-per-use monitors for the calculation of time-based usage fees.
- ***Data Persistence*** - This statistic is provided by pay-per-use monitors that record the number of objects, individual occupied storage sizes, and database transactions per billing period.
- ***Network Usage*** - Inbound and outbound network usage is tracked for pay- per-use monitors and SLA monitors that track network-related QoS metrics.
- ***Failure Conditions*** - SLA monitors that track the QoS metrics of IT resources need to capture failure statistics.
- ***Event Triggers*** - This metric is primarily used by audit monitors that need to respond to certain types of events.

Security

The PaaS environment, by default, does not usually introduce the need for new cloud security mechanisms beyond those that are already provisioned for IaaS environments.

Optimizing SaaS Environments

In SaaS implementations, cloud service architectures are generally based on multitenant environments that enable and regulate concurrent cloud consumer access (Figure 14.3).

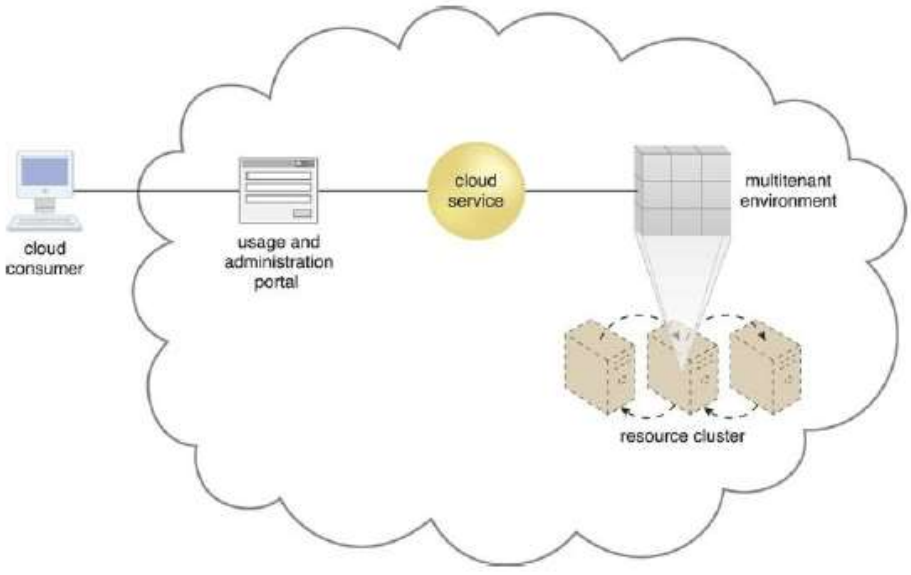

Figure 14.3. The SaaS-based cloud service is hosted by a multitenant environment deployed in a high-performance virtual server cluster. A usage and administration portal is used by the cloud consumer to access and configure the cloud service.

SaaS IT resource segregation does not typically occur at the infrastructure level in SaaS environments, as it does in IaaS and PaaS environments.

SaaS implementations rely heavily on the features provided by the native dynamic scalability and workload distribution architectures, as well as nondisruptive service relocation to ensure that failover conditions do not impact the availability of SaaS-based cloud services.

However, it is vital to acknowledge that, unlike the relatively vanilla designs of IaaS and PaaS products, each SaaS deployment will bring with it unique architectural, functional, and runtime requirements. These requirements are specific to the nature of the business logic the SaaS-based cloud service is programmed with, as well as the distinct usage patterns it is subjected to by its cloud service consumers.

For example, consider the diversity in functionality and usage of the following recognized online SaaS offerings:

- collaborative authoring and information-sharing (Wikipedia, Blogger)
- collaborative management (Zimbra, Google Apps)
- conferencing services for instant messaging, audio/video communications (Skype, Google Talk)
- enterprise management systems (ERP, CRM, CM)
- file-sharing and content distribution (YouTube, Dropbox)
- industry-specific software (engineering, bioinformatics)
- messaging systems (e-mail, voicemail)
- mobile application marketplaces (Android Play Store, Apple App Store)
- office productivity software suites (Microsoft Office, Adobe Creative Cloud)
- search engines (Google, Yahoo)
- social networking media (Twitter, LinkedIn)

Now consider that many of the previously listed cloud services are offered in one or more of the following implementation mediums:

- mobile application
- REST service
- Web service

Each of these SaaS implementation mediums provide Web-based APIs for interfacing by cloud consumers. Examples of online SaaS-based cloud services with Web-based APIs include:

- electronic payment services (PayPal)
- mapping and routing services (Google Maps)
- publishing tools (WordPress)

Mobile-enabled SaaS implementations are commonly supported by the multidevice broker mechanism, unless the cloud service is intended exclusively for access by specific mobile devices.

The potentially diverse nature of SaaS functionality, the variation in implementation technology, and the tendency to offer a SaaS-based cloud service redundantly with multiple different implementation mediums makes the design of SaaS environments highly specialized. Though not essential to a SaaS

implementation, specialized processing requirements can prompt the need to incorporate architectural models, such as:

- ***Service Load Balancing*** - for workload distribution across redundant SaaS-based cloud service implementations
- ***Dynamic Failure Detection and Recovery*** - to establish a system that can automatically resolve some failure conditions without disruption in service to the SaaS implementation.
- ***Storage Maintenance Window*** - to allow for planned maintenance outages that do not impact SaaS implementation availability
- ***Elastic Resource Capacity/Elastic Network Capacity*** - to establish inherent elasticity within the SaaS-based cloud service architecture that enables it to automatically accommodate a range of runtime scalability requirements
- ***Cloud Balancing*** - to instill broad resiliency within the SaaS implementation, which can be especially important for cloud services subjected to extreme concurrent usage volumes

Specialized cloud usage monitors can be used in SaaS environments to track the following types of metrics:

- ***Tenant Subscription Period*** - This metric is used by pay-per-use monitors to record and track application usage for time-based billing. This type of monitoring usually incorporates application licensing and regular assessments of leasing periods that extend beyond the hourly periods of IaaS and PaaS environments.
- ***Application Usage*** - This metric, based on user or security groups, is used with pay-per-use monitors to record and track application usage for billing purposes.
- ***Tenant Application Functional Module*** - This metric is used by pay-per- use monitors for function-based billing. Cloud services can have different functionality tiers according to whether the cloud consumer is free-tier or a paid subscriber.

Cloud Delivery Models: The Cloud Consumer Perspective

This section raises various considerations concerning the different ways in which cloud delivery models are administered and utilized by cloud consumers.

Working with IaaS Environments

Virtual servers are accessed at the operating system level through the use of remote terminal applications. Accordingly, the type of client software used directly depends on the type of operating system that is running at the virtual server, of which two common options are:

- ***Remote Desktop (or Remote Desktop Connection) Client*** - for Windows- based environments and presents a Windows GUI desktop
- ***SSH Client*** - for Mac and other Linux-based environments to allow for secure channel connections to text-based shell accounts running on the server OS

Figure 14.4 illustrates a typical usage scenario for virtual servers that are being offered as IaaS services after having been created with management interfaces

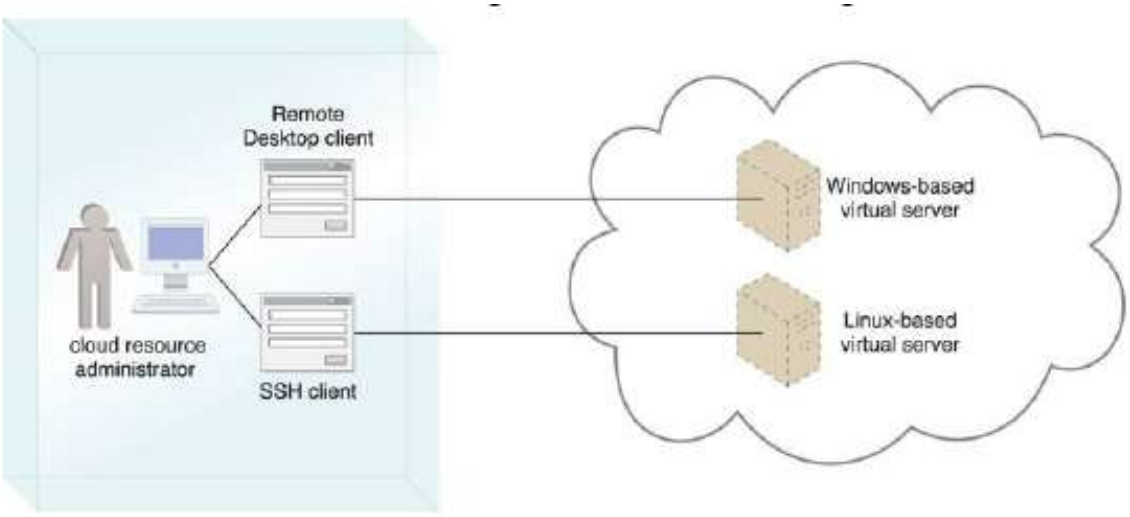

Figure 14.4. A cloud resource administrator uses the Windows-based Remote Desktop client to administer a Windows-based virtual server and the SSH client for the Linux-based virtual server.

A cloud storage device can be attached directly to the virtual servers and accessed through the virtual servers' functional interface for management by the operating system. Alternatively, a cloud storage device can be attached to an IT resource that is being hosted outside of the cloud, such as an on-premise device over a WAN or VPN. In these cases, the following formats for the manipulation and transmission of cloud storage data are commonly used:

- ***Networked File System*** - System-based storage access, whose rendering of files is similar to how folders are organized in operating systems (NFS, CIFS)
- ***Storage Area Network Devices*** - Block-based storage access collates and formats geographically diverse data into cohesive files for optimal network transmission (iSCSI, Fibre Channel)
- ***Web-Based Resources*** - Object-based storage access by which an interface that is not integrated into the operating system logically represents files, which can be accessed through a Web-based interface (Amazon S3)

IT Resource Provisioning Considerations

Cloud consumers have a high degree of control over how and to what extent IT resources are provisioned as part of their IaaS environments.

For example:

- controlling scalability features (automated scaling, load balancing)
- controlling the lifecycle of virtual IT resources (shutting down, restarting, powering up of virtual devices)
- controlling the virtual network environment and network access rules (firewalls, logical network perimeters)
- establishing and displaying service provisioning agreements (account conditions, usage terms)
- managing the attachment of cloud storage devices

- managing the pre-allocation of cloud-based IT resources (resource reservation)
- managing credentials and passwords for cloud resource administrators
- managing credentials for cloud-based security groups that access virtualized IT resources through an IAM
- managing security-related configurations
- managing customized virtual server image storage (importing, exporting, backup)
- selecting high-availability options (failover, IT resource clustering)
- selecting and monitoring SLA metrics
- selecting basic software configurations (operating system, pre-installed software for new virtual servers) selecting IaaS resource instances from a number of available hardware- related configurations and options (processing capabilities, RAM, storage)
- selecting the geographical regions in which cloud-based IT resources should be hosted
- tracking and managing costs

The management interface for these types of provisioning tasks is usually a usage and administration portal, but may also be offered via the use of command line interface (CLI) tools that can simplify the execution of many scripted administrative actions.

Even though standardizing the presentation of administrative features and controls is typically preferred, using different tools and user-interfaces can sometimes be justified. For example, a script can be made to turn virtual servers on and off nightly through a CLI, while adding or removing storage capacity can be more easily carried out using a portal.

Working with PaaS Environments

A typical PaaS IDE can offer a wide range of tools and programming resources, such as software libraries, class libraries, frameworks, APIs, and various runtime capabilities that emulate the intended cloud-based deployment environment. These features allow developers to create, test, and run application code within the cloud or locally (on-premise) while using the IDE to emulate the cloud deployment environment. Compiled or completed applications are then bundled and uploaded to the cloud, and deployed via the ready-made environments. This deployment process can also be controlled through the IDE.

PaaS also allows for applications to use cloud storage devices as independent data storing systems for holding development-specific data (for example in a repository that is available outside of the cloud environment). Both SQL and NoSQL database structures are generally supported.

IT Resource Provisioning Considerations

PaaS environments provide less administrative control than IaaS environments, but still offer a significant range of management features.

For example:

- establishing and displaying service provisioning agreements, such as account conditions and usage terms

- selecting software platform and development frameworks for ready-made environments

selecting instance types, which are most commonly frontend or backend instances

- selecting cloud storage devices for use in ready-made environments
- controlling the lifecycle of PaaS-developed applications (deployment, starting, shutdown, restarting, and release)
- controlling the versioning of deployed applications and modules
- configuring availability and reliability-related mechanisms
- managing credentials for developers and cloud resource administrators using IAM
- managing general security settings, such as accessible network ports
- selecting and monitoring PaaS-related SLA metrics
- managing and monitoring usage and IT resource costs
- controlling scalability features such as usage quotas, active instance thresholds, and the configuration and deployment of the automated scaling listener and load balancer mechanisms

Working with SaaS Services

Because SaaS-based cloud services are almost always accompanied by refined and generic APIs, they are usually designed to be incorporated as part of larger distributed solutions. A common example of this is Google Maps, which offers a comprehensive API that enables mapping information and images to be incorporated into Web sites and Web-based applications.

Many SaaS offerings are provided free of charge, although these cloud services often come with data collecting sub-programs that harvest usage data for the benefit of the cloud provider. When using any SaaS product that is sponsored by third parties, there is a reasonable chance that it is performing a form of background information gathering. Reading the cloud provider's agreement will usually help shed light on any secondary activity that the cloud service is designed to perform.

Cloud consumers using SaaS products supplied by cloud providers are relieved of the responsibilities of implementing and administering their underlying hosting environments. Customization options are usually available to cloud consumers; however, these options are generally limited to the runtime usage control of the cloud service instances that are generated specifically by and for the cloud consumer.

For example:

- managing security-related configurations
- managing select availability and reliability options
- managing usage costs
- managing user accounts, profiles, and access authorization
- selecting and monitoring SLAs

setting manual and automated scalability options and limitations.

Cost Metrics and Pricing Models

Reducing operating costs and optimizing IT environments are pivotal to understanding and being able to compare the cost models behind provisioning on-premise and cloud-based environments. The pricing structures used by public clouds are typically based on utility-centric pay-per-usage models, enabling organizations to avoid up-front infrastructure investments. These models need to be assessed against the financial implications of on-premise infrastructure investments and associated total cost-of-ownership commitments.

Business Cost Metrics

Cloud Usage Cost Metrics

Cost Management Considerations

Business Cost Metrics

This section begins by describing the common types of metrics used to evaluate the estimated costs and business value of leasing cloud-based IT resources when compared to the purchase of on-premise IT resources.

Up-Front and On-Going Costs

Up-front costs are associated with the initial investments that organizations need to make in order to fund the IT resources they intend to use. This includes both the costs associated with obtaining the IT resources, as well as expenses required to deploy and administer them.

- Up-front costs for the purchase and deployment of on-premise IT resources tend to be high. Examples of up-front costs for on-premise environments can include hardware, software, and the labor required for deployment.
- Up-front costs for the leasing of cloud-based IT resources tend to be low. Examples of up-front costs for cloud-based environments can include the labor costs required to assess and set up a cloud environment.

On-going costs represent the expenses required by an organization to run and maintain IT resources it uses.

- On-going costs for the operation of on-premise IT resources can vary. Examples include licensing fees, electricity, insurance, and labor.
- On-going costs for the operation of cloud-based IT resources can also vary, but often exceed the on-going costs of on-premise IT resources (especially over a longer period of time). Examples include virtual hardware leasing fees, bandwidth usage fees, licensing fees, and labor.

Additional Costs

To supplement and extend a financial analysis beyond the calculation and comparison of standard up-front and on-going business cost metrics, several other more specialized business cost metrics can be taken into account.

For example:

- ***Cost of Capital*** - The ***cost of capital*** is a value that represents the cost incurred by raising

required funds. For example, it will generally be more expensive to raise an initial investment of $150,000 than it will be to raise this amount over a period of three years. The relevancy of this cost depends on how the organization goes about gathering the funds it requires. If the cost of capital for an initial investment is high, then it further helps justify the leasing of cloud-based IT resources.

- ***Sunk Costs*** - An organization will often have existing IT resources that are already paid for and operational. The prior investment that has been made in these on-premise IT resources is referred to as ***sunk costs.*** When comparing up-front costs together with significant sunk costs, it can be more difficult to justify the leasing of cloud-based IT resources as an alternative.
- ***Integration Costs*** - Integration testing is a form of testing required to measure the effort required to make IT resources compatible and interoperable within a foreign environment, such as a new cloud platform. Depending on the cloud deployment model and cloud delivery model being considered by an organization, there may be the need to further allocate funds to carry out integration testing and additional labor related to enable interoperability between cloud service consumers and cloud services. These expenses are referred to as ***integration costs.*** High integration costs can make the option of leasing cloud-based IT resources less appealing.
- ***Locked-in Costs*** - As explained in the ***Risks and Challenges*** section in Chapter 3. cloud environments can impose portability limitations. When performing a metrics analysis over a longer period of time, it may be necessary to take into consideration the possibility of having to move from one cloud provider to another. Due to the fact that cloud service consumers can become dependent on proprietary characteristics of a cloud environment, there are ***locked-in costs*** associated with this type of move. Locked-in costs can further decrease the long-term business value of leasing cloud-based IT resources.

Cloud Usage Cost Metrics

The following sections describe a set of usage cost metrics for calculating costs associated with cloud-based IT resource usage measurements:

- ***Network Usage*** - inbound and outbound network traffic, as well as intracloud network traffic
- ***Server Usage*** - virtual server allocation (and resource reservation)
- ***Cloud Storage Device*** - storage capacity allocation
- ***Cloud Service*** - subscription duration, number of nominated users, number of transactions (of cloud services and cloud-based applications)

For each usage cost metric a description, measurement unit, and measurement frequency is provided, along with the cloud delivery model most applicable to the metric. Each metric is further supplemented with a brief example.

1 Network Usage

Defined as the amount of data that is transferred over a network connection, network usage is typically calculated using separately measured ***inbound network usage traffic*** and ***outbound network usage traffic*** metrics in relation to cloud services or other IT resources.

Inbound Network Usage Metric

- ***Description*** - inbound network traffic
- ***Measurement*** - £, inbound network traffic in bytes

- ***Frequency*** - continuous and cumulative over a predefined period
- ***Cloud Delivery Model*** - IaaS, PaaS, SaaS
- ***Example*** - up to 1 GB free, $0.001/GB up to 10 TB a month

Outbound Network Usage Metric

- ***Description*** - outbound network traffic
- ***Measurement*** - £, outbound network traffic in bytes
- ***Frequency*** - continuous and cumulative over a predefined period
- ***Cloud Delivery Model*** - IaaS, PaaS, SaaS
- ***Example*** - up to 1 GB free a month, $0.01/GB between 1 GB to 10 TB per month

Network usage metrics can be applied to WAN traffic between IT resources of one cloud that are located in different geographical regions in order to calculate costs for synchronization, data replication, and related forms of processing. Conversely, LAN usage and other network traffic among IT resources that reside at the same data center are typically not tracked.

Intra-Cloud WAN Usage Metric

- ***Description*** - network traffic between geographically diverse IT resources of the same cloud
- ***Measurement*** - £, intra-cloud WAN traffic in bytes
- ***Frequency*** - continuous and cumulative over a predefined period
- ***Cloud Delivery Model*** - IaaS, PaaS, SaaS
- ***Example*** - up to 500 MB free daily and $0.01/GB thereafter, $0.005/GB after 1 TB per month

Many cloud providers do not charge for inbound traffic in order to encourage cloud consumers to migrate data to the cloud. Some also do not charge for WAN traffic within the same cloud.

Network-related cost metrics are determined by the following properties:

- ***Static IP Address Usage*** - IP address allocation time (if a static IP is required)
- ***Network Load-Balancing*** - the amount of load-balanced network traffic (in bytes)
- ***Virtual Firewall*** - the amount of firewall-processed network traffic (as per allocation time)

2-Server Usage

The allocation of virtual servers is measured using common pay-per-use metrics in IaaS and PaaS environments that are quantified by the number of virtual servers and ready-made environments. This form of server usage measurement is divided into ***on-demand virtual machine instance allocation*** and ***reserved virtual machine instance allocation*** metrics.

The former metric measures pay-per-usage fees on a short-term basis, while the latter metric calculates up-front reservation fees for using virtual servers over extended periods. The up-front reservation fee is usually used in conjunction with the discounted pay-per-usage fees.

On-Demand Virtual Machine Instance Allocation Metric

- ***Description*** - uptime of a virtual server instance
- ***Measurement*** - E, virtual server start date to stop date

- ***Frequency*** - continuous and cumulative over a predefined period
- ***Cloud Delivery Model*** - IaaS, PaaS
- ***Example*** - $0.10/hour small instance, $0.20/hour medium instance, $0.90/hour large instance

Reserved Virtual Machine Instance Allocation Metric

- ***Description*** - up-front cost for reserving a virtual server instance
- ***Measurement*** - E, virtual server reservation start date to expiry date
- ***Frequency*** - daily, monthly, yearly
- ***Cloud Delivery Model*** - IaaS, PaaS
- ***Example*** - $55.10/small instance, $99.90/medium instance, $249.90/large instance

Another common cost metric for virtual server usage measures performance capabilities. Cloud providers of IaaS and PaaS environments tend to provision virtual servers with a range of performance attributes that are generally determined by CPU and RAM consumption and the amount of available dedicated allocated storage.

3-Cloud Storage Device Usage

Cloud storage is generally charged by the amount of space allocated within a predefined period, as measured by the ***on-demand storage allocation*** metric. Similar to IaaS-based cost metrics, on-demand storage allocation fees are usually based on short time increments (such as on an hourly basis). Another common cost metric for cloud storage is ***I/O data transferred,*** which measures the amount of transferred input and output data.

On-Demand Storage Space Allocation Metric

- ***Description*** - duration and size of on-demand storage space allocation in bytes
- ***Measurement*** - E, date of storage release / reallocation to date of storage allocation (resets upon change in storage size)
- ***Frequency*** – continuous
- ***Cloud Delivery Model*** - IaaS, PaaS, SaaS
- ***Example*** - $0.01/GB per hour (typically expressed as GB/month)

I/O Data Transferred Metric

- ***Description*** - amount of transferred I/O data
- ***Measurement*** - E, I/O data in bytes
- ***Frequency*** - continuous
- ***Cloud Delivery Model*** - IaaS, PaaS
- ***Example*** - $0.10/TB

4-Cloud Service Usage

Cloud service usage in SaaS environments is typically measured using the following three metrics:

Application Subscription Duration Metric

- ***Description*** - duration of cloud service usage subscription
- ***Measurement*** - E, subscription start date to expiry date
- ***Frequency*** - daily, monthly, yearly
- ***Cloud Delivery Model*** - SaaS
- ***Example*** - $69.90 per month

Number of Nominated Users Metric

- ***Description*** - number of registered users with legitimate access
- ***Measurement*** - number of users
- ***Frequency*** - monthly, yearly
- ***Cloud Delivery Model*** - SaaS
- ***Example*** - $0.90/additional user per month

Number of Transactions Users Metric

- ***Description*** - number of transactions served by the cloud service
- ***Measurement*** - number of transactions (request-response message exchanges)
- ***Frequency*** - continuous
- ***Cloud Delivery Model*** **- PaaS, SaaS**
- **Example -** ***$0.05 per 1,000 transaction***

Cost Management Considerations

Cost management is often centered around the lifecycle phases of cloud services, as follows:

- ***Cloud Service Design and Development*** - During this stage, the vanilla pricing models and cost templates are typically defined by the organization delivering the cloud service.
- ***Cloud Service Deployment*** - Prior to and during the deployment of a cloud service, the backend architecture for usage measurement and billing- related data collection is determined and implemented, including the positioning of pay-per-use monitor and billing management system mechanisms.
- ***Cloud Service Contracting*** - This phase consists of negotiations between the cloud consumer and cloud provider with the goal of reaching a mutual agreement on rates based on usage cost metrics.
- ***Cloud Service Offering*** - This stage entails the concrete offering of a cloud service's pricing models through cost templates, and any available customization options.
- ***Cloud Service Provisioning*** - Cloud service usage and instance creation thresholds may be imposed by the cloud provider or set by the cloud consumer. Either way, these and other provisioning options can impact usage costs and other fees.
- ***Cloud Service Operation*** - This is the phase during which active usage of the cloud service produces usage cost metric data.
- ***Cloud Service Decommissioning*** - When a cloud service is temporarily or permanently

deactivated, statistical cost data may be archived.

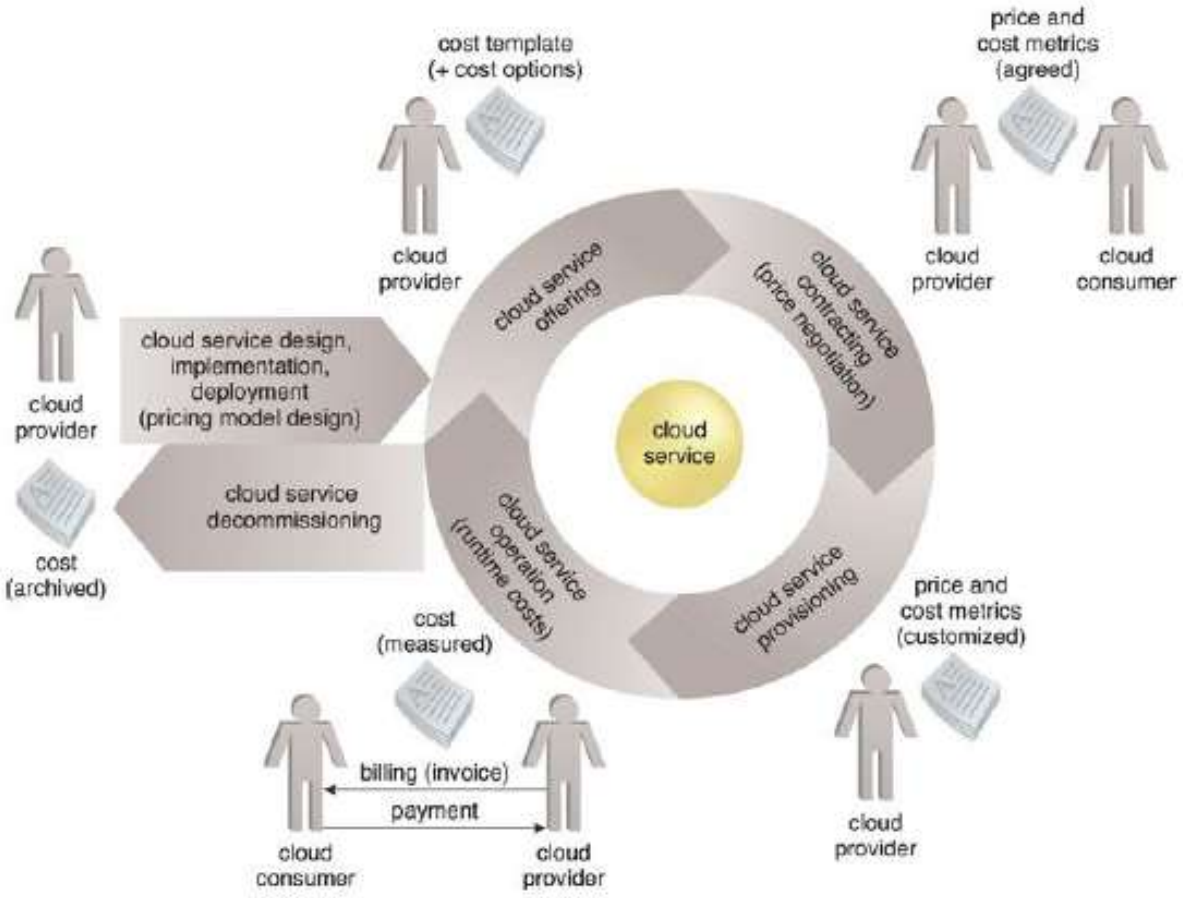

Figure 15.1. Common cloud service lifecycle stages as they relate to cost management considerations.

Pricing Models

The pricing models used by cloud providers are defined using templates that specify unit costs for fine-grained resource usage according to usage cost metrics. Various factors can influence a pricing model, such as:

- market competition and regulatory requirements
- overhead incurred during the design, development, deployment, and operation of cloud services and other IT resources
- opportunities to reduce expenses via IT resource sharing and data center optimization

Most major cloud providers offer cloud services at relatively stable, competitive prices even though their own expenses can be volatile. A price template or pricing plan contains a set of standardized costs and metrics that specify how cloud service fees are measured and calculated. Price templates define a pricing model's structure by setting various units of measure, usage quotas, discounts, and other codified fees. A pricing model can contain multiple price templates, whose formulation is determined by variables like:

- ***Cost Metrics and Associated Prices*** - These are costs that are dependent on the type of IT resource allocation (such as on-demand versus reserved allocation).

- ***Fixed and Variable Rates Definitions*** - Fixed rates are based on resource allocation and define the usage quotas included in the fixed price, while variable rates are aligned with actual resource usage.
- ***Volume Discounts*** - More IT resources are consumed as the degree of IT resource scaling progressively increases, thereby possibly qualifying a cloud consumer for higher discounts.
- ***Cost and Price Customization Options*** - This variable is associated with payment options and schedules. For example, cloud consumers may be able to choose monthly, semi-annual, or annual payment installments.

Additional Considerations

- ***Negotiation*** - Cloud provider pricing is often open to negotiation, especially for customers willing to commit to higher volumes or longer terms. Price negotiations can sometimes be executed online via the cloud provider's Web site by submitting estimated usage volumes along with proposed discounts. There are even tools available for cloud consumers to help generate accurate IT resource usage estimates for this purpose.
- ***Payment Options*** - After completing each measurement period, the cloud provider's billing management system calculates the amount owed by a cloud consumer. There are two common payment options available to cloud consumers: pre-payment and post-payment. With pre-paid billing, cloud consumers are provided with IT resource usage credits that can be applied to future usage bills. With the post-payment method, cloud consumers are billed and invoiced for each IT resource consumption period, which is usually on a monthly basis.
- ***Cost Archiving*** - By tracking historical billing information both cloud providers and cloud consumers can generate insightful reports that help identify usage and financial trends.

Service Quality Metrics and SLAs

Service-level agreements (SLAs) are a focal point of negotiations, contract terms, legal obligations, and runtime metrics and measurements. SLAs formalize the guarantees put forth by cloud providers, and correspondingly influence or determine the pricing models and payment terms. SLAs set cloud consumer expectations and are integral to how organizations build business automation around the utilization of cloud-based IT resources.

The guarantees made by a cloud provider to a cloud consumer are often carried forward, in that the same guarantees are made by the cloud consumer organization to its clients, business partners, or whomever will be relying on the services and solutions hosted by the cloud provider. It is therefore crucial

for SLAs and related service quality metrics to be understood and aligned support of the cloud consumer's business requirements, while also ensuring that the guarantees can, in fact, be realistically fulfilled consistently and reliably by the cloud provider. The latter consideration is especially relevant for cloud providers that host shared IT resources for high volumes of cloud consumers, each of which will have been issued its own SLA guarantees.

Service Quality Metrics
SLA Guidelines

Service Quality Metrics

SLAs issued by cloud providers are human-readable documents that describe quality-of-service (QoS) features, guarantees, and limitations of one or more cloud-based IT resources.

SLAs use service quality metrics to express measurable QoS characteristics.

For example:

- ***Availability*** - up-time, outages, service duration
- ***Reliability*** - minimum time between failures, guaranteed rate of successful responses
- ***Performance*** - capacity, response time, and delivery time guarantees
- ***Scalability*** - capacity fluctuation and responsiveness guarantees
- ***Resiliency*** - mean-time to switchover and recovery

SLA management systems use these metrics to perform periodic measurements that verify compliance with SLA guarantees, in addition to collecting SLA- related data for various types of statistical analyses. Each service quality metric is ideally defined using the following characteristics:

- ***Quantifiable*** - The unit of measure is clearly set, absolute, and appropriate so that the metric can be based on quantitative measurements.
- ***Repeatable*** - The methods of measuring the metric need to yield identical results when repeated under identical conditions.
- ***Comparable*** - The units of measure used by a metric need to be standardized and comparable. For example, a service quality metric cannot measure smaller quantities of data in bits and larger quantities in bytes.
- ***Easily Obtainable*** - The metric needs to be based on a non-proprietary, common form of measurement that can be easily obtained and understood by cloud consumers.

Service Availability Metrics

Availability Rate Metric

The overall availability of an IT resource is usually expressed as a percentage of up-time. For example, an IT resource that is always available will have an uptime of 100%.

- ***Description*** - percentage of service up-time
- ***Measurement*** - total up-time / total time
- ***Frequency*** - weekly, monthly, yearly

- ***Cloud Delivery Model*** - IaaS, PaaS, SaaS
- ***Example*** - minimum 99.5% up-time

Availability rates are calculated cumulatively, meaning that unavailability periods are combined in order to compute the total downtime (Table 16.1)

Table 16.1. Sample availability rates measured in units of seconds.

Availability (%)	Downtime/Week (Seconds)	Downtime/Month (Seconds)	Downtime/Year (Seconds)
99.5	3024	216	158112
99.8	1210	5174	63072
99.9	606	2592	31536
99.95	302	1294	15768
99.99	60.6	259.2	3154
99.999	6.05	25.9	316.6
99.9999	0.605	2.59	31.5

Outage Duration Metric

This service quality metric is used to define both maximum and average continuous outage service-level targets.

- ***Description*** - duration of a single outage
- ***Measurement*** - date/time of outage end - date/time of outage start
- ***Frequency*** - per event
- ***Cloud Delivery Model*** - IaaS, PaaS, SaaS

Example - 1 hour maximum, 15 minute average

Service Reliability Metrics

A characteristic closely related to availability, reliability is the probability that an IT resource can perform its intended function under pre-defined conditions without experiencing failure. Reliability focuses on how often the service performs as expected, which requires the service to remain in an operational and available state. Certain reliability metrics only consider runtime errors and exception conditions as failures, which are commonly measured only when the IT resource is available.

Mean-Time Between Failures (MTBF) Metric

- ***Description*** - expected time between consecutive service failures
- ***Measurement*** - £, normal operational period duration / number of failures
- ***Frequency*** - monthly, yearly

- ***Cloud Delivery Model*** - IaaS, PaaS
- ***Example*** - 90 day average

Reliability Rate Metric

Overall reliability is more complicated to measure and is usually defined by a reliability rate that represents the percentage of successful service outcomes.

This metric measures the effects of non-fatal errors and failures that occur during up-time periods. For example, an IT resource's reliability is 100% if it has performed as expected every time it is invoked, but only 80% if it fails to perform every fifth time.

- ***Description*** - percentage of successful service outcomes under pre-defined conditions
- ***Measurement*** - total number of successful responses / total number of requests
- ***Frequency*** - weekly, monthly, yearly
- ***Cloud Delivery Model*** - SaaS
- ***Example*** - minimum 99.5%

Service Performance Metrics

Service performance refers to the ability on an IT resource to carry out its functions within expected parameters. This quality is measured using service capacity metrics, each of which focuses on a related measurable characteristic of IT resource capacity. A set of common performance capacity metrics is provided in this section. Note that different metrics may apply, depending on the type of IT resource being measured.

Network Capacity Metric

- ***Description*** - measurable characteristics of network capacity
- ***Measurement*** - bandwidth / throughput in bits per second
- ***Frequency*** - continuous
- ***Cloud Delivery Model*** - IaaS, PaaS, SaaS
- ***Example*** - 10 MB per second

Storage Device Capacity Metric

- ***Description*** - measurable characteristics of storage device capacity
- ***Measurement*** - storage size in GB
- ***Frequency*** - continuous
- ***Cloud Delivery Model*** - IaaS, PaaS, SaaS
- ***Example*** - 80 GB of storage

Server Capacity Metric

- ***Description*** - measurable characteristics of server capacity
- ***Measurement*** - number of CPUs, CPU frequency in GHz, RAM size in GB, storage size in GB
- ***Frequency*** - continuous
- ***Cloud Delivery Model*** - IaaS, PaaS

- ***Example*** - 1 core at 1.7 GHz, 16 GB of RAM, 80 GB of storage

Web Application Capacity Metric

- ***Description*** - measurable characteristics of Web application capacity
- ***Measurement*** - rate of requests per minute
- ***Frequency*** - continuous
- ***Cloud Delivery Model*** - SaaS
- ***Example*** - maximum 100,000 requests per minute

Instance Starting Time Metric

- ***Description*** - length of time required to initialize a new instance
- ***Measurement*** - date/time of instance up - date/time of start request
- ***Frequency*** - per event
- ***Cloud Delivery Model*** - IaaS, PaaS
- ***Example*** - 5 minute maximum, 3 minute average

Response Time Metric

- ***Description*** - time required to perform synchronous operation
- ***Measurement*** - (date/time of request - date/time of response) / total number of requests
- ***Frequency*** - daily, weekly, monthly
- ***Cloud Delivery Model*** - SaaS
- ***Example*** - 5 millisecond average

Completion Time Metric

- ***Description*** - time required to complete an asynchronous task
- ***Measurement*** - (date of request - date of response) / total number of requests
- ***Frequency*** - daily, weekly, monthly
- ***Cloud Delivery Model*** **- PaaS, SaaS**
- **Example -** ***1 second average***

Service Scalability Metrics

Service scalability metrics are related to IT resource elasticity capacity, which is related to the maximum capacity that an IT resource can achieve, as well as measurements of its ability to adapt to workload fluctuations. For example, a server can be scaled up to a maximum of 128 CPU cores and 512 GB of RAM, or scaled out to a maximum of 16 load-balanced replicated instances.

The following metrics help determine whether dynamic service demands will be met proactively or reactively, as well as the impacts of manual or automated IT resource allocation processes.

Storage Scalability (Horizontal) Metric

- ***Description*** - permissible storage device capacity changes in response to increased workloads
- ***Measurement*** - storage size in GB

- ***Frequency*** - continuous
- ***Cloud Delivery Model*** - IaaS, PaaS, SaaS
- ***Example*** - 1,000 GB maximum (automated scaling)

Server Scalability (Horizontal) Metric

- ***Description*** - permissible server capacity changes in response to increased workloads
- ***Measurement*** - number of virtual servers in resource pool
- ***Frequency*** - continuous
- ***Cloud Delivery Model*** - IaaS, PaaS
- ***Example*** - 1 virtual server minimum, 10 virtual server maximum (automated scaling)

Server Scalability (Vertical) Metric

- ***Description*** - permissible server capacity fluctuations in response to workload fluctuations
- ***Measurement*** - number of CPUs, RAM size in GB
- ***Frequency*** - continuous
- ***Cloud Delivery Model*** - IaaS, PaaS
- ***Example*** - 512 core maximum, 512 GB of RAM

Service Resiliency Metrics

The ability of an IT resource to recover from operational disturbances is often measured using service resiliency metrics. When resiliency is described within or in relation to SLA resiliency guarantees, it is often based on redundant implementations and resource replication over different physical locations, as well as various disaster recovery systems.

The type of cloud delivery model determines how resiliency is implemented and measured. For example, the physical locations of replicated virtual servers that are implementing resilient cloud services can be explicitly expressed in the SLAs for IaaS environments, while being implicitly expressed for the corresponding PaaS and SaaS environments.

Resiliency metrics can be applied in three different phases to address the challenges and events that can threaten the regular level of a service:

- ***Design Phase*** - Metrics that measure how prepared systems and services are to cope with challenges.
- ***Operational Phase*** - Metrics that measure the difference in service levels before, during, and after a downtime event or service outage, which are further qualified by availability, reliability, performance, and scalability metrics.
- ***Recovery Phase*** - Metrics that measure the rate at which an IT resource recovers from downtime, such as the meantime for a system to log an outage and switchover to a new virtual server.

Two common metrics related to measuring resiliency are as follows:

Mean-Time to Switchover (MTSO) Metric

- ***Description*** - the time expected to complete a switchover from a severe failure to a replicated instance in a different geographical area

- ***Measurement*** - (date/time of switchover completion - date/time of failure) / total number of failures
- ***Frequency*** - monthly, yearly
- ***Cloud Delivery Model*** - IaaS, PaaS, SaaS
- ***Example*** - 10 minute average

Mean-Time System Recovery (MTSR) Metric

- ***Description -*** time expected for a resilient system to perform a complete recovery from a severe failure
- ***Measurement*** - (date/time of recovery - date/time of failure) / total number of failures
- ***Frequency*** - monthly, yearly
- ***Cloud Delivery Model*** - IaaS, PaaS, SaaS
- ***Example*** - 120 minute average

SLA Guidelines

This section provides a number of best practices and recommendations for working with SLAs, the majority of which are applicable to cloud consumers:

- ***Mapping Business Cases to SLAs*** - It can be helpful to identify the necessary QoS requirements for a given automation solution and to then concretely link them to the guarantees expressed in the SLAs for IT resources responsible for carrying out the automation. This can avoid situations where SLAs are inadvertently misaligned or perhaps unreasonably deviate in their guarantees, subsequent to IT resource usage.
- ***Working with Cloud and On-Premise SLAs*** - Due to the vast infrastructure available to support IT resources in public clouds, the QoS guarantees issued in SLAs for cloud-based IT resources are generally superior to those provided for on-premise IT resources. This variance needs to be understood, especially when building hybrid distributed solutions that utilize both on on-premise and cloud-based services or when incorporating cross-environment technology architectures, such as cloud bursting.
- ***Understanding the Scope of an SLA*** - Cloud environments are comprised of many supporting architectural and infrastructure layers upon which IT resources reside and are integrated. It is important to acknowledge the extent to which a given IT resource guarantee applies. For example, an SLA may be limited to the IT resource implementation but not its underlying hosting environment.
- ***Understanding the Scope of SLA Monitoring*** - SLAs need to specify where monitoring is performed and where measurements are calculated, primarily in relation to the cloud's firewall. For example, monitoring within the cloud firewall is not always advantageous or relevant to the cloud consumer's required QoS guarantees. Even the most efficient firewalls have a measurable degree of influence on performance and can further present a point of failure.
- ***Documenting Guarantees at Appropriate Granularity*** - SLA templates used by cloud providers sometimes define guarantees in broad terms. If a cloud consumer has specific requirements, the

corresponding level of detail should be used to describe the guarantees. For example, if data replication needs to take place across particular geographic locations, then these need to be specified directly within the SLA.

- ***Defining Penalties for Non-Compliance*** - If a cloud provider is unable to follow through on the QoS guarantees promised within the SLAs, recourse can be formally documented in terms of compensation, penalties, reimbursements, or otherwise.
- ***Incorporating Non-Measurable Requirements*** - Some guarantees cannot be easily measured using service quality metrics, but are relevant to QoS nonetheless, and should therefore still be documented within the SLA. For example, a cloud consumer may have specific security and privacy requirements for data hosted by the cloud provider that can be addressed by assurances in the SLA for the cloud storage device being leased.
- ***Disclosure of Compliance Verification and Management*** - Cloud providers are often responsible for monitoring IT resources to ensure compliance with their own SLAs. In this case, the SLAs themselves should state what tools and practices are being used to carry out the compliance checking process, in addition to any legal-related auditing that may be occurring.
- ***Inclusion of Specific Metric Formulas*** - Some cloud providers do not mention common SLA metrics or the metrics-related calculations in their SLAs, instead focusing on service-level descriptions that highlight the use of best practices and customer support. Metrics being used to measure SLAs should be part of the SLA document, including the formulas and calculations that the metrics are based upon.
- ***Considering Independent SLA Monitoring*** - Although cloud providers will often have sophisticated SLA management systems and SLA monitors, it may be in the best interest of a cloud consumer to hire a third-party organization to perform independent monitoring as well, especially if there are suspicions that SLA guarantees are not always being met by the cloud provider (despite the results shown on periodically issued monitoring reports).
- ***Archiving SLA Data*** - The SLA-related statistics collected by SLA monitors are commonly stored and archived by the cloud provider for future reporting purposes. If a cloud provider intends to keep SLA data specific to a cloud consumer even after the cloud consumer no longer continues its business relationship with the cloud provider, then this should be disclosed. The cloud consumer may have data privacy requirements that disallow the unauthorized storage of this type of information. Similarly, during and after a cloud consumer's engagement with a cloud provider, it may want to keep a copy of historical SLA-related data as well. It may be especially useful for comparing cloud providers in the future.
- ***Disclosing Cross-Cloud Dependencies*** - Cloud providers may be leasing IT resources from other cloud providers, which results in a loss of control over the guarantees they are able to make to cloud consumers. Although a cloud provider will rely on the SLA assurances made to it by other cloud providers, the cloud consumer may want disclosure of the fact that the IT resources it is leasing may have dependencies beyond the environment of the cloud provider organization.