CHAPTER 3

Managed Service

A managed service provider (MSP) is a third-party contractor that delivers network-based services, applications and equipment to enterprises, residences or other service providers.

Managed service providers can be hosting companies or access providers that offer IT services such as fully outsourced network management arrangements, including IP telephony, messaging and call center management, virtual private networks (VPNs), managed firewalls and monitoring/reporting of network servers. Most of these services can be performed from outside a company's internal network with a special emphasis placed on integration and certification of Internet security for applications and content. MSPs serve a outsourcing agents for companies, especially other service providers like ISPs, that don't have the resources to constantly upgrade or maintain faster and faster computer networks.

Managed services providers can offer services such as alerts, security, patch management, data backup and recovery for different client devices: desktops, notebooks, servers, storage systems, networks and applications. Offloading routine infrastructure management to an experienced managed services professional lets you concentrate on running your business, with fewer interruptions due to IT issues.

MSPs act as an extension of your IT department, taking care of routine IT infrastructure monitoring and management around the clock—freeing up your IT staff to focus on higher-value projects. By proactively monitoring and maintaining your systems, an MSP can help you avoid many technology problems in the first place. Should an issue occur, an experienced MSP can troubleshoot and resolve it more efficiently.

Evolution from Managed Service to Cloud Computing

Managed service providers are feeling a bit of pressure from cloud computing. After years of investing in infrastructure and business model changes to deliver remote managed services, MSPs are seeing the cloud as a threat to their livelihood. In many cases, the cloud requires no remote infrastructure or on-premise equipment. The fear is the cloud could render MSPs obsolete.

Cloud computing could also be a managed service. Cloud computing consists of security monitoring, storage management, network administration and the all managed services need not necessarily be cloud computing. Cloud computing puts its efforts in creating a technical solution. It is a technical model that delivers technical access to the computing resources. Thus, cloud computing can be defined as a technical solution.

Managed services, on the other hand, is a contract based relationship. The definition and delivery of the service is done on a repeating revenue basis. It means that managed services are recurring revenues from well-defined services that are predictable.

Some types of services that managed service provider’s offer are help desk assistance and network administration services. Managed services allow the Information Technology staff of the company to focus their efforts and energy in to the core activities of the company rather than dealing with the IT challenges. A managed service is the management of technology like telephony, IT, applications and others. However, the definition of a managed service is changing.

There are, however a lot of similarities that exist between a managed service and cloud computing. While cloud computing is generally a technical implementation that decides how the infrastructure and applications are to be delivered over the private and public networks, it can also be a business model. The same contract agreement holds ground in a cloud computing relationship. Managed services can also be a technical implementation.

Cloud Optimized Infrastructure is based around 5 key belief that offer capabilities that differentiate as an MSP from cloud computing:

- Flexible licensing which supports the elastic expansion/contraction of cloud-based services and accommodates the billing implications
- Low overhead in deployment and use across server, network, and storage resources, making it a great fit for virtual machine environments that are a key supporting technology in cloud-based computing
- Non-disruptive scalability which accommodates the need for server, storage and other infrastructure growth on both the end user and cloud provider sides without impacting client-side production servers
- Enterprise multi tenancy that provides for the secure delivery of reliable services to multiple customers with a scalable management model
- Broad heterogeneous support that maximizes cloud provider market opportunities by covering a wide range of server, storage, and application environments found in customer settings

Single Purpose Architecture to Multipurpose Architecture

In the early days of MSPs, the providers would actually go onto customer sites and perform their services on customer-owned premises. Over time, these MSPs specialized in implementation of infrastructure and quickly figured out ways to build out data centers and sell those capabilities off in small chunks commonly known as monthly recurring services, in addition to the basic fees charged.

With virtualization and other supporting technologies, MSP’s quickly convinced their customers to shift their data centers to multipurpose and multitenant architecture. Not only multipurpose architecture was effective for their customers but also was huge cost saving initiative.

Data Center Virtualization

Data center virtualization is a method of moving information storage from physical servers to virtual ones, often in a different location. In the past, large companies would keep physical servers on site that held huge amounts of corporate information. These servers were expensive, both to purchase and maintain. With data center virtualization, it became possible to separate both the hardware and location from the data. This cuts costs and increases the data's availability.

Data center virtualization actually comes from a combination of two different technologies; high-speed data transfer and server virtualization. Without both of these components, data center virtualization becomes highly impractical.

There are three areas of IT where virtualization is making inroads, network virtualization, storage virtualization and server virtualization:

- **Network virtualization** is a method of combining the available resources in a network by splitting up the available bandwidth into channels, each of which is independent from the others, and each of which can be assigned (or reassigned) to a particular server or device in real time. The idea is that virtualization disguises the true complexity of the network by separating it into manageable parts, much like your partitioned hard drive makes it easier to manage your files.

- **Storage virtualization** is the pooling of physical storage from multiple network storage devices into what appears to be a single storage device that is managed from a central console. Storage virtualization is commonly used in storage area networks (SANs).

- **Server virtualization** is the masking of server resources, including the number and identity of individual physical servers, processors, and operating systems, from server users. The server administrator uses a software application to divide one physical server into multiple isolated virtual environments. The virtual environments are sometimes called virtual private servers, but they are also known as guests, instances, containers or emulations.

Benefits of Data Center Virtualization

In addition to the cost savings that result from reducing the number of servers, the following benefits are typically realized as well:

- Enables the consolidation of physical servers, slashing the costs of operating a data center. This includes reducing the costs of server upgrades, management, power, space, and storage.

- Reduction in data center space and in data center equipment such as PDUs, air conditioning units, etc.
- Reduction in the number of network, HBAs and SAN switches.
- Provides true high-availability for all servers without requiring duplicate hardware and clustering software.
- Integrates the test/development and production environments while significantly enhancing the test/development process.
- Facilitates true disaster recovery for all servers.
- Eliminates the need for maintenance windows for physical server troubleshooting or upgrades and enables faster server provisioning.
- Enhances security and provides regulatory compliance benefits.

Cloud Data Center

A cloud data center has three distinct characteristics that differentiate it from traditional DC. It is sold on demand, typically by the minute or the hour; it is elastic -- a user can have as much or as little of a service as they want at any given time; and the service is fully managed by the provider (the consumer needs nothing but a personal computer and Internet access). Significant innovations in virtualization and distributed computing, as well as improved access to high-speed Internet and a weak economy, have accelerated interest in cloud computing.

The multiplier effect of Internet data, trends of enterprises information transformation and tremendous load bring forth traditional DC huge challenges: how to reduce operation and maintenance cost, how to meet demand of high capacity, high security and high efficiency. Attributing to resource on-demand, flexible and dynamic structure, cloud computing is the right technology to resolve all these issues. By continuously improving core technologies including virtualization, elastic computing and high-density computing etc, cloud vendors have creatively developed Modular cloud computing Data Center of trusty, efficient, smart, ultra-bandwidth and green end to end solution.

Cloud Data Center is now evolving beyond being merely a model of technology delivery to becoming a new operating model where business decision makers are empowered to procure infrastructure on demand, and where IT becomes an internal service provider delivering increased business agility without compromising security or control.

Ultimately, cloud services are attractive because the cost is likely to be far lower than providing the same service from your traditional data center.

Traditional Corporate Data Center	**Cloud Data Center**
Thousands of different applications	Few applications (maybe even just one)
Mixed hardware environment	Homogeneous hardware environment
Multiple management tools	Standardized management tools
Frequent application patching and updating	Minimal application patching and updating
Complex workloads	Simple workloads
Multiple software architectures	Single standard software architecture

Service Oriented Architecture

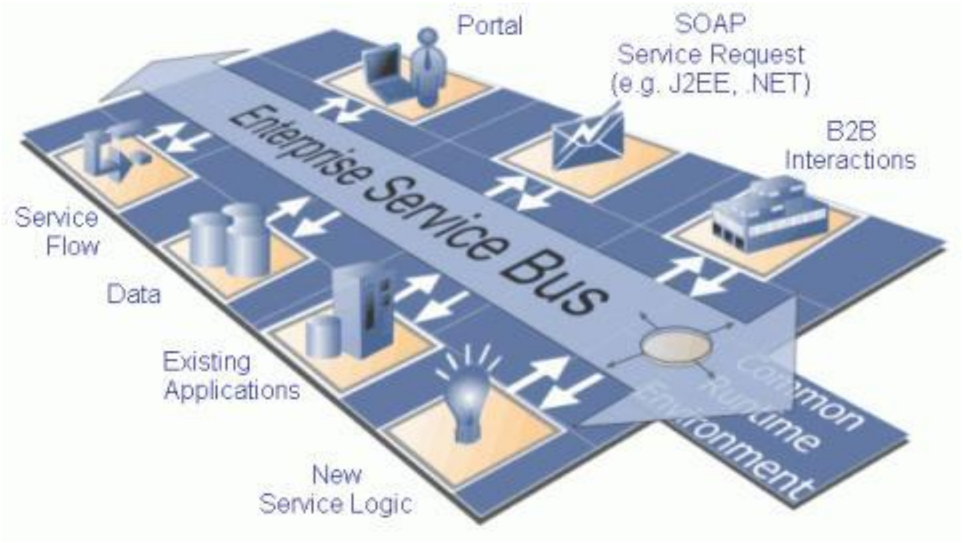
If a service-oriented architecture is to be effective, we need a clear understanding of the term **service**. A service is a function that is well-defined, self-contained, and does not depend on the context or state of other services.

A service-oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. SOA defines how to integrate widely disparate applications for a Web-based environment and uses multiple implementation platforms. **Rather than defining an API**, SOA defines the interface in terms of protocols and functionality.

Why SOA?

The reality in IT enterprises is that infrastructure is heterogeneous across operating systems, applications, system software, and application infrastructure. Some existing applications are used to run current business processes, so starting from scratch to build new infrastructure isn't an option.

Enterprises should quickly respond to business changes with agility; leverage existing investments in applications and application infrastructure to address newer business requirements; support new channels of interactions with customers, partners, and suppliers; and feature an architecture that supports organic business. SOA with its loosely coupled nature allows enterprises to plug in new services or upgrade existing services in a granular fashion to address the new business requirements, provides the option to make the services consumable across different channels, and exposes the existing



enterprise and legacy applications as services, thereby safeguarding existing IT infrastructure investments.

For example, a core banking application provides a Fund Transfer service, then the other banking applications such as Treasury, Payment Gateway, ATM Switching, and so on can call or invoke Fund Transfer service without need to worry about where the Fund Transfer is located in the network. This contrasts with the Tight Coupling approach. Each application defining their own Fund Transfer, the problem come when the Transfer Fund logic is change. It will be difficult and require high cost (and time, of course) to set the new logic into each application.

Service-oriented architectures are not new. The first service-oriented architectures are usually considered to be the Distributed Component Object Model (DCOM) or Object Request Brokers (ORBs), which were based on the Common Object Requesting Broker Architecture (CORBA) specification. The introduction of SOA provides a platform for technology and business units to meet business requirements of the modern enterprise. With SOA, your organization can use existing application systems to a greater extent and may respond faster to change requests. These benefits are attributed to several critical elements of SOA:

1. Free-standing, independent components
2. Combined by loose coupling
3. Message (XML)-based instead of API-based
4. Physical location, etc., not important

Characteristics of SOA

- In SOA, Services should be **independent** of other services. Altering a service should not affect calling service.
- Services should be **self-contained**. When we talk about a Register Customer service it means, service will do all the necessary work for us, we are not required to care about anything.
- Services should be able to **define themselves**. Services should be able to answer a question what is does? It should be able to tell client what all operations it does, what all data types it uses and what kind of responses it will return.
- Services should be **published** into a location (directory) where anyone can search for it.
- As said, SOA comprises of collection services which communicate via **standard Messages**. Standard messages make them platform independent. (Here standard doesn't mean standard across Microsoft it means across all programming languages and technologies.)
- Services should be able to communicate with each other **asynchronously**.
- Services should support **reliable messaging**.Means there should be a guarantee that request will be reached to correct destination and correct response will be obtained.
- Services should support **secure communication**.

About Open Source Software

Open-source software (OSS) is computer software with its source code made available and licensed with an open-source license in which the copyright holder provides the rights to study, change and distribute the software for free to anyone and for any purpose. Open-source software is very often developed in a public, collaborative manner.

The basics behind the Open Source Initiative is that when programmers can read, redistribute and modify the source code for a piece of software, the software evolves. Open source sprouted in the technological community as a response to proprietary software owned by corporations.

Proprietary software is privately owned and controlled. In the computer industry, proprietary is considered the opposite of open. A proprietary design or technique is one that is owned by a company. It also implies that the company has not divulged specifications that would allow other companies to duplicate the product.

Open Source is a certification standard issued by the Open Source Initiative (OSI) that indicates that the source code of a computer program is made available free of charge to the general public. OSI dictates that in order to be considered "OSI Certified" a product must meet the following criteria:

Some Points to Know about Open Source Software

- The author or holder of the license of the source code cannot collect royalties on the distribution of the program.
- The distributed program must make the source code accessible to the user.
- The author must allow modifications and derivations of the work under the program's original name.
- No person, group or field of endeavor can be denied access to the program.
- The rights attached to the program must not depend on the program's being part of a particular software distribution.
- The licensed software cannot place restrictions on other software that is distributed with it

Linux, Apache and other open-source applications have long been used to power Web and file servers. But when it comes to managing the data center, many companies have held back. Now, though, some users have turned into big believers that open source works here, too.

The following open source packages take a more holistic approach by integrating all of the necessary functionality into a single package (including virtualization, management, interfaces, and security). When added to a network of servers and storage, these packages produce flexible cloud computing and storage infrastructures (IaaS).

Eucalyptus

One of the most popular open source packages for building cloud computing infrastructures is Eucalyptus (for Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems). What makes it unique is that its interface is compatible with Amazon Elastic Compute Cloud (Amazon EC2—Amazon's cloud computing interface). Additionally, Eucalyptus includes Walrus, which is a cloud storage application compatible with Amazon Simple Storage Service (Amazon S3—Amazon's cloud storage interface).

Eucalyptus supports KVM/Linux and Xen for hypervisors and includes the Rocks cluster distribution for cluster management.

OpenNebula

OpenNebula is another interesting open source application (under the Apache license) developed at the Universidad Complutense de Madrid. In addition to supporting private cloud construction, OpenNebula supports the idea of hybrid clouds. A hybrid cloud permits combining a private cloud infrastructure with a public cloud infrastructure (such as Amazon) to enable even higher degrees of scaling.

OpenNebula supports Xen, KVM/Linux, and VMware and relies on elements like libvirt for management and introspection.

Nimbus

Nimbus is another IaaS solution focused on scientific computing. With Nimbus, you can lease remote resources (such as those provided by Amazon EC2) and manage them locally (configure, deploy VMs, monitor, etc.). Nimbus morphed from the Workspace Service project (part of Globus.org). Being dependent on Amazon EC2, Nimbus supports Xen and KVM/Linux.

Xen Cloud Platform

Citrix has integrated Xen into an IaaS platform, using Xen as the hypervisor while incorporating other open source capabilities such as the Open vSwitch. An interesting advantage to the Xen solution is the focus on standards-based management (including OVF, Distributed Management Task Force [DTMF], the Common Information Model [CIM], and Virtualization Management Initiative [VMAN]) from the project Kensho. The Xen management stack supports SLA guarantees, along with detailed metrics for charge-back.

OpenQRM

Our penultimate solution is OpenQRM, which is categorized as a data center management platform. OpenQRM provides a single console to manage an entire virtualized data center that is architecturally pluggable to permit integration of third-party tools. OpenQRM integrates support for high availability (through redundancy) and supports a variety of hypervisors, including KVM/Linux, Xen, VMware, and Linux VServer.

OpenStack

Today, the leading IaaS solution is called OpenStack. OpenStack was released in July 2010, and has quickly become the standard open-source IaaS solution. OpenStack is a combination of two cloud initiatives from RackSpace Hosting (Cloud Files) and NASA's Nebula platform. OpenStack is being developed in the Python language, and is under active development under the Apache license.

Apache

The Apache HTTP Server, commonly referred to as Apache , is a web server software program notable for playing a key role in the initial growth of the World Wide Web .In 2009 it became the first web server software to surpass the 100 million website milestone. Apache was the first viable alternative to the Netscape Communications Corporation web server (currently named Oracle iPlanet Web Server). Typically Apache is run on a Unix-like operating system and was developed for use on Linux. The Apache HTTP Server Project is a collaborative software development effort aimed at creating a robust, commercial-grade, feature-rich and freely-available source code implementation of an HTTP (Web) server.

Advantages of OSS

Open-source software is free to use, distribute, and modify. It has lower costs, and in most cases this is only a fraction of the cost of their proprietary counterparts.

Open-source software is more secured as the code is accessible to everyone. Anyone can fix bugs as they are found, and users do not have to wait for the next release. The fact that is continuously analyzed by a large community produces secure and stable code.

Open source is not dependent on the company or author that originally created it. Even if the company fails, the code continues to exist and be developed by its users. Also, it uses open standards accessible to everyone; thus, it does not have the problem of incompatible formats that exist in proprietary software.

Lastly, the companies using open-source software do not have to think about complex licensing models and do not need anti-piracy measures like product activation or serial number.

Disadvantages of OSS

The main disadvantage of open-source software is not being straightforward to use. Open-source operating systems like Linux cannot be learned in a day. They require effort and possibly training from your side before you are able to master them. You may need to hire a trained person to make things easier, but this will incur additional costs.

There is a shortage of applications that run both on open source and proprietary software; therefore, switching to an open-source platform involves a compatibility analysis of all the other software used that run on proprietary platforms. In addition, there are many ongoing parallel developments on open source software. This creates confusion on what functionalities are present in which versions.

Lastly, many of the latest hardware are incompatible to the open-source platform; so you have to rely on third-party drivers.