WEB SERVICES

A Web service is a method of communication between two electronic devices over a network. It is a software function provided at a network address over the Web with the service always-on as in the concept of utility computing

Service Oriented Architecture (SOA)

SOA is **an architectural style for building software applications** that use services available in a network such as the web. It promotes loose coupling between software components so that they can be reused

- **Services -** The services are the logical entities defined by one or more published interfaces.
- **Service provider -** It is a software entity that implements a service specification.
- **Service consumer -** It can be called as a requestor or client that calls a service provider. A service consumer can be another service or an end-user application.
- **Service locator -** It is a service provider that acts as a registry. It is responsible for examining service provider interfaces and service locations.
- **Service broker -** It is a service provider that pass service requests to one or more additional service providers.

Characteristics of SOA

The services have the following characteristics:

- They are loosely coupled.
- They support interoperability.
- They are location-transparent
- They are self-contained.

Advantages of SOA

SOA has the following advantages:

- Easy to integrate - In a service-oriented architecture, the integration is a service specification that provides implementation transparency.
- Manage Complexity - Due to service specification, the complexities get isolated, and integration becomes more manageable.
- Platform Independence - The services are platform-independent as they can communicate with other applications through a common language.
- Loose coupling - It facilitates to implement services without impacting other applications or services.
-

CHARACTERISTICS of web services

XML-Based

Web services use XML at data representation and data transportation layers. Using XML eliminates any networking, operating system, or platform binding. Web services based applications are highly interoperable at their core level.

Loosely Coupled

A consumer of a web service is not tied to that web service directly. The web service interface can change over time without compromising the client's ability to interact with the service. A tightly coupled system implies that the client and server logic are closely tied to one another, implying that if one interface changes, the other must be updated. Adopting a loosely coupled architecture tends to make software systems more manageable and allows simpler integration between different systems.

Coarse-Grained

Object-oriented technologies such as Java expose their services through individual methods. An individual method is too fine an operation to provide any useful capability at a corporate level. Building a Java program from scratch requires the creation of several fine-grained methods that are then composed into a coarse-grained service that is consumed by either a client or another service.

Businesses and the interfaces that they expose should be coarse-grained. Web services technology provides a natural way of defining coarse-grained services that access the right amount of business logic.

Ability to be Synchronous or Asynchronous

Synchronicity refers to the binding of the client to the execution of the service. In synchronous invocations, the client blocks and waits for the service to complete its operation before continuing. Asynchronous operations allow a client to invoke a service and then execute other functions.

Components of Web Services

XML-RPC

Remote Procedure Calls.

This is the simplest XML-based protocol for exchanging information between computers.

- XML-RPC is a simple protocol that uses XML messages to perform RPCs.
- Requests are encoded in XML and sent via HTTP POST.
- XML responses are embedded in the body of the HTTP response.
- XML-RPC is platform-independent.
- XML-RPC allows diverse applications to communicate.
- A Java client can speak XML-RPC to a Perl server.
- XML-RPC is the easiest way to get started with web services.

To learn more about XML-RPC, visit our XML-RPC Tutorial.

SOAP Simple Object Access Protocol

SOAP stands for Simple Object Access Protocol. It is a **XML-based protocol for accessing web services**. SOAP is a W3C recommendation for communication between two applications.

SOAP is an XML-based protocol for exchanging information between computers.

- SOAP is a communication protocol.
- SOAP is for communication between applications.
- SOAP is a format for sending messages.
- SOAP is designed to communicate via Internet.
- SOAP is platform independent.
- SOAP is language independent.
- SOAP is simple and extensible.
- SOAP allows you to get around firewalls.
- SOAP will be developed as a W3C standard.

To learn more about SOAP, visit our SOAP Tutorial.

WSDL

WSDL stands for **Web Services** Description Language. It is the standard format for describing a **web service**.

WSDL is an XML-based language for describing web services and how to access them.

- WSDL stands for Web Services Description Language.
- WSDL was developed jointly by Microsoft and IBM.
- WSDL is an XML based protocol for information exchange in decentralized and distributed environments.
- WSDL is the standard format for describing a web service.
- WSDL definition describes how to access a web service and what operations it will perform.
- WSDL is a language for describing how to interface with XML-based services.
- WSDL is an integral part of UDDI, an XML-based worldwide business registry.
- WSDL is the language that UDDI uses.
- WSDL is pronounced as 'wiz-dull' and spelled out as 'W-S-D-L'.

To learn more about WSDL, visit our WSDL Tutorial.

UDDI

UDDI stands for Universal Description, Discovery, and Integration.

UDDI is an XML-based standard for describing, publishing, and finding web services.

- UDDI stands for Universal Description, Discovery, and Integration.
- UDDI is a specification for a distributed registry of web services.
- UDDI is platform independent, open framework.
- UDDI can communicate via SOAP, CORBA, and Java RMI Protocol.
- UDDI uses WSDL to describe interfaces to web services.
- UDDI is seen with SOAP and WSDL as one of the three foundation standards of web services.
- UDDI is an open industry initiative enabling businesses to discover each other and define how they interact over the Internet.

Web Service Architecture

Web Service Roles

There are three major roles within the web service architecture −

Service Provider

This is the provider of the web service. The service provider implements the service and makes it available on the Internet.

Service Requestor

This is any consumer of the web service. The requestor utilizes an existing web service by opening a network connection and sending an XML request.

Service Registry

This is a logically centralized directory of services. The registry provides a central place where developers can publish new services or find existing ones. It therefore serves as a centralized clearing house for companies and their services.

Web Service Protocol Stack

A second option for viewing the web service architecture is to examine the emerging web service protocol stack. The stack is still evolving, but currently has four main layers.

Service Transport

This layer is responsible for transporting messages between applications. Currently, this layer includes Hyper Text Transport Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP), and newer protocols such as Blocks Extensible Exchange Protocol (BEEP).

XML Messaging

This layer is responsible for encoding messages in a common XML format so that messages can be understood at either end. Currently, this layer includes XML-RPC and SOAP.

Service Description

This layer is responsible for describing the public interface to a specific web service. Currently, service description is handled via the Web Service Description Language (WSDL).

Service Discovery

This layer is responsible for centralizing services into a common registry and providing easy publish/find functionality. Currently, service discovery is handled via Universal Description, Discovery, and Integration (UDDI).

As web services evolve, additional layers may be added and additional technologies may be added to each layer.

The next chapter explains the components of web services.

Few Words about Service Transport

The bottom of the web service protocol stack is service transport. This layer is responsible for actually transporting XML messages between two computers.

Hyper Text Transfer Protocol (HTTP)

Currently, HTTP is the most popular option for service transport. HTTP is simple, stable, and widely deployed. Furthermore, most firewalls allow HTTP traffic. This allows XMLRPC or SOAP messages to masquerade as HTTP messages. This is good if you want to integrate remote applications, but it does raise a number of security concerns.ise a number of security concerns.

Blocks Extensible Exchange Protocol (BEEP)

This is a promising alternative to HTTP. BEEP is a new Internet Engineering Task Force (IETF) framework for building new protocols. BEEP is layered directly on TCP and includes a number of built-in features, including an initial handshake protocol, authentication, security, and error handling. Using BEEP, one can create new protocols for a variety of applications, including instant messaging, file transfer, content syndication, and network management.

XML

What is XML?

XML is a software- and hardware-independent tool for storing and transporting data.
XML stands for eXtensible Markup Language
XML is a markup language much like HTML
XML was designed to store and transport data
XML was designed to be self-descriptive
XML is a W3C Recommendation
The XML above is quite self-descriptive:

It has sender information.
It has receiver information
It has a heading
It has a message body.
But still, the XML above does not DO anything. XML is just information wrapped in tags.
https://www.w3schools.com/xml/xml_whatis.asp

ATTACKS IN CLOUD COMPUTING

- **Data threats**

Cloud users store various types of data in cloud environments, and a lot of that data contains sensitive information about users or business activities. However, this data is susceptible to loss, breach, or damage as the result of human actions, application vulnerabilities, and unforeseen emergencies. It’s obvious that a cloud service provider can’t prevent all data threats, but cloud developers should apply modern encryption algorithms to ensure the integrity of data in transit from the user to the cloud.

- **Cloud API vulnerabilities**

Application programming interfaces (APIs) allow users to interact with cloud-based services. However, vulnerabilities in APIs may significantly impact the security of cloud orchestration, management, provisioning, and monitoring. Cloud developers need to implement strong controls over APIs.

- **Malicious insiders**

Legitimate cloud users who act maliciously have many ways to arrange attacks or leak data in cloud environments. This threat can be minimized by cloud developers, however, by implementing identity and access management (IAM) technologies.

- **Shared technology vulnerabilities**

Cloud computing involves the use of shared technologies such as virtualization and cloud orchestration. Thus, by exploiting vulnerabilities in any part of these technologies, attackers can cause significant damage to many cloud users. Weaknesses in a hypervisor can allow hackers to gain control over virtual machines or even the host itself. In the case of a virtual machine escape, hackers can gain unrestricted access to the host through shared resources. So it’s necessary to pay attention to the security of the cloud provider that you entrust with your cloud solution.

10 Most Common Types of Attacks on Cloud Computing

1. Cloud malware injection attacks

Malware injection attacks are done to take control of a user’s information in the cloud. For this purpose, hackers add an infected service implementation module to a SaaS or PaaS solution or a virtual machine instance to an IaaS solution. If the cloud system is successfully deceived, it will redirect the cloud user’s requests to the hacker’s module or instance, initiating the execution of malicious code. Then the attacker can begin their malicious activity such as manipulating or stealing data or eavesdropping.

The most common forms of malware injection attacks are cross-site scripting attacks and SQL injection attacks. During a cross-site scripting attack, hackers add malicious scripts (Flash, JavaScript, etc.) to a vulnerable web page. German researchers arranged an XSS attack against the Amazon Web Services cloud computing platform in 2011. In the case of SQL injection, attackers target SQL servers with vulnerable database applications. In 2008, Sony's PlayStation website became the victim of a SQL injection attack.

2. Abuse of cloud services

Hackers can use cheap cloud services to arrange DoS and brute force attacks on target users, companies, and even other cloud providers. For instance, security experts Bryan and Anderson arranged a DoS attack by exploiting capacities of Amazon's EC2 cloud infrastructure in 2010. As a result, they managed to make their client unavailable on the internet by spending only $6 to rent virtual services.

An example of a brute force attack was demonstrated by Thomas Roth at the 2011 Black Hat Technical Security Conference. By renting servers from cloud providers, hackers can use powerful cloud capacities to send thousands of possible passwords to a target user's account.

3. Denial of service attacks

DoS attacks are designed to overload a system and make services unavailable to its users. These attacks are especially dangerous for cloud computing systems, as many users may suffer as the result of flooding even a single cloud server. In case of high workload, cloud systems begin to provide more computational power by involving more virtual machines and service instances. While trying to prevent a cyber attack, the cloud system actually makes it more devastating. Finally, the cloud system slows down and legitimate users lose any availability to access their cloud services. In the cloud environment, DDoS attacks may be even more dangerous if hackers use more zombie machines to attack a large number of systems.

4. Side channel attacks

A side channel attack is arranged by hackers when they place a malicious virtual machine on the same host as the target virtual machine. During a side channel attack, hackers target system implementations of cryptographic algorithms. However, this type of threat can be avoided with a secure system design.

5. Wrapping attacks

A wrapping attack is an example of a man-in-the-middle attack in the cloud environment. Cloud computing is vulnerable to wrapping attacks because cloud users typically connect to services via a web browser. An XML signature is used to protect users' credentials from unauthorized access, but this signature doesn't secure the positions in the document. Thus, XML signature element wrapping allows attackers to manipulate an XML document.

Tips on How to Ensure the Security of Cloud-Based Solutions

1. Enhance security policies

When providing cloud services, software vendors should limit the scope of their responsibility for protecting user data and operations in the cloud in their security policies. Inform your clients about what you do to ensure cloud security as well as what security measures they need to take on their side.

2. Use strong authentication

Stealing passwords is the most common way to access users' data and services in the cloud. Thus, cloud developers should implement strong authentication and identity management. Establish multi-factor authentication. There are various tools that require both static passwords and dynamic passwords. The latter confirms a user's credentials by providing a one-time password on a mobile phone or using biometric schemes or hardware tokens.

3. Implement access management

To increase the security of services, cloud developers should let cloud users assign role-based permissions to different administrators so that users only have the capabilities assigned to them. Moreover, cloud orchestration should enable privileged users to establish the scope of other users' permissions according to their duties within the company.

Read also:
Multi-Cloud Computing: Pros and Cons for Enterprise

4. Protect data

Data in the cloud environment needs to be encrypted at all stages of its transfer and storage:

- at the source (on the user’s side)
- in transit (during its transfer from the user to the cloud server)
- at rest (when stored in the cloud database)

Data needs to be encrypted even before it goes to the cloud. Modern data encryption and tokenization technologies are an effective defense against account hijacking. Moreover, it’s important to prove end-to-end encryption for protecting data in transit against man-in-the-middle attacks. Using strong encryption algorithms that contain salt and hashes can effectively deflect cyber attacks.

Data stored in the cloud is also vulnerable to unintentional damage, so you can also ensure its recovery by providing a data backup service.

5. Detect intrusions

Provide your cloud-based solution with a fully managed intrusion detection system that can detect and inform about the malicious use of cloud services by intruders. Use an intrusion detection system that provides network monitoring and notifies about the abnormal behavior of insiders.

6. Secure APIs and access

Cloud developers should be sure that clients can access the application only through secure APIs. This might require limiting the range of IP addresses or providing access only through corporate networks or VPNs. However, this approach can be difficult to implement for public-facing applications. Thus, you can implement security protection via an API using special scripts, templates, and recipes. You can even go further and build security protection into your API.

SLA SLO SLI

SLA **or Service Level Agreement** is a contract that the service provider promises customers on service availability, performance, etc. SLO or Service Level Objective is a goal that service provider wants to reach. SLI or Service Level Indicator is a measurement the service provider uses for the goal.

PLATFORM AS A SERVICE

Platform as a Service (PaaS) provides a runtime environment. It allows programmers to easily create, test, run, and deploy web applications.

In PaaS, back end scalability is managed by the cloud service provider, so end- users do not need to worry about managing the infrastructure.

PaaS includes infrastructure (servers, storage, and networking) and platform (middleware, development tools, database management systems, business intelligence, and more) to support the web application life cycle.

Example: Google App Engine, Force.com, Joyent, Azure.

Advantages of PaaS

There are the following advantages of PaaS -

1) Simplified Development

PaaS allows developers to focus on development and innovation without worrying about infrastructure management.

2) Lower risk

No need for up-front investment in hardware and software. Developers only need a PC and an internet connection to start building applications.

3) Prebuilt business functionality

Some PaaS vendors also provide already defined business functionality so that users can avoid building everything from very scratch and hence can directly start the projects only.

4) Instant community

PaaS vendors frequently provide online communities where the developer can get the ideas to share experiences and seek advice from others.

5) Scalability

Applications deployed can scale from one to thousands of users without any changes to the applications.

Disadvantages of PaaS cloud computing layer

1) Vendor lock-in

One has to write the applications according to the platform provided by the PaaS vendor, so the migration of an application to another PaaS vendor would be a problem.

2) Data Privacy

Corporate data, whether it can be critical or not, will be private, so if it is not located within the walls of the company, there can be a risk in terms of privacy of data.

3) Integration with the rest of the systems applications

It may happen that some applications are local, and some are in the cloud. So there will be chances of increased complexity when we want to use data which in the cloud with the local data

Software as a Service | SaaS

SaaS is also known as "**On-Demand Software**". It is a software distribution model in which services are hosted by a cloud service provider. These services are available to end-users over the internet so, the end-users do not need to install any software on their devices to access these services.

There are the following services provided by SaaS providers -
Business Services - SaaS Provider provides various business services to start-up the business. The SaaS business services include **ERP** (Enterprise Resource Planning), **CRM** (Customer Relationship Management), **billing**, and **sales**.
Document Management - SaaS document management is a software application offered by a third party (SaaS providers) to create, manage, and track electronic documents.
Example: Slack, Samepage, Box, and Zoho Forms.
Social Networks - As we all know, social networking sites are used by the general public, so social networking service providers use SaaS for their convenience and handle the general public's information.
Mail Services - To handle the unpredictable number of users and load on e-mail services, many e-mail providers offering their services using SaaS.

Advantages of SaaS cloud computing layer

1) SaaS is easy to buy

SaaS pricing is based on a monthly fee or annual fee subscription, so it allows organizations to access business functionality at a low cost, which is less than licensed applications.

Unlike traditional software, which is sold as a licensed based with an up-front cost (and often an optional ongoing support fee), SaaS providers are generally pricing the applications using a subscription fee, most commonly a monthly or annually fee.

2. One to Many

SaaS services are offered as a one-to-many model means a single instance of the application is shared by multiple users.

. Less hardware required for SaaS

The software is hosted remotely, so organizations do not need to invest in additional hardware.

4. Low maintenance required for SaaS

Disadvantages of SaaS cloud computing layer

1) Security

Actually, data is stored in the cloud, so security may be an issue for some users. However, cloud computing is not more secure than in-house deployment.

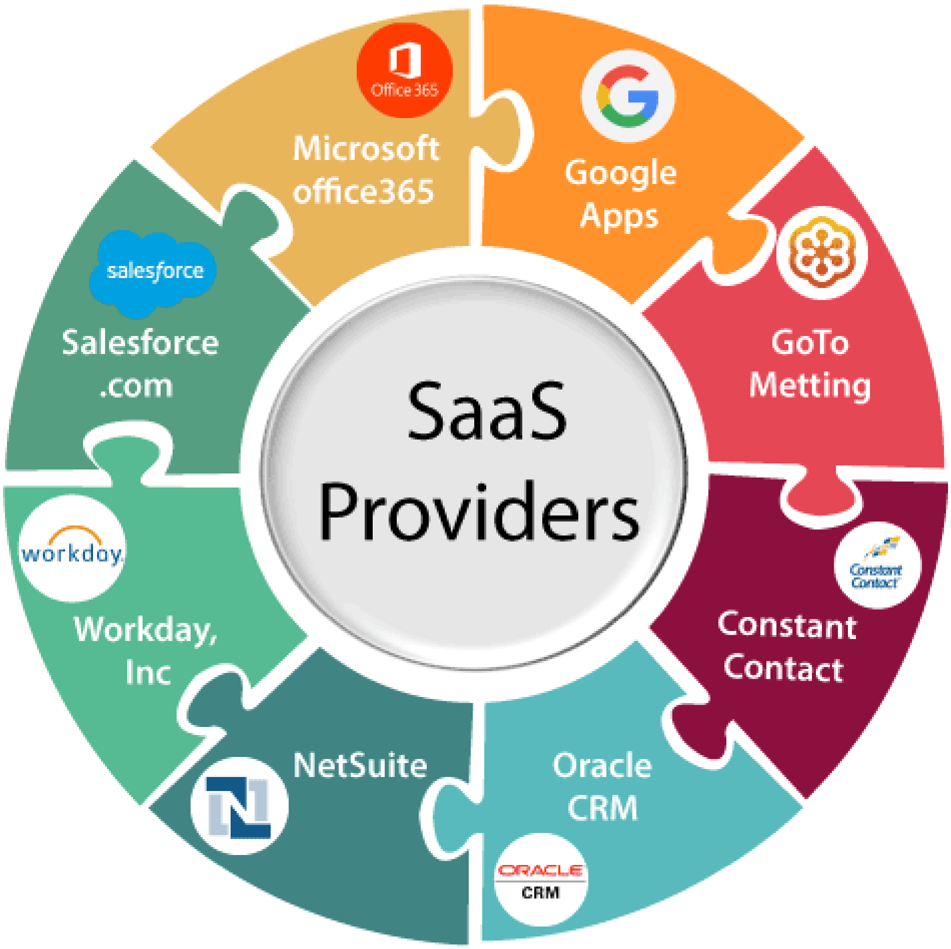
2) Latency issue

Since data and applications are stored in the cloud at a variable distance from the end-user, there is a possibility that there may be greater latency when interacting with the application compared to local deployment. Therefore, the SaaS model is not suitable for applications whose demand response time is in milliseconds.

3) Total Dependency on Internet

Without an internet connection, most SaaS applications are not usable.

4) Switching between SaaS vendors is difficult



Infrastructure as a Service | IaaS

Iaas is also known as **Hardware as a Service (HaaS)**. It is one of the layers of the cloud computing platform. It allows customers to outsource their IT infrastructures such as servers, networking, processing, storage, virtual machines, and other resources. Customers access these resources on the Internet using a pay-as-per use model. IaaS cloud computing platform layer eliminates the need for every organization to maintain the IT infrastructure.

IaaS is offered in three models: **public, private, and hybrid cloud.** The private cloud implies that the infrastructure resides at the customer-premise. In the case of public cloud, it is located at the cloud computing platform vendor's data center, and the hybrid cloud is a combination of the two in which the customer selects the best of both public cloud or private cloud.

IaaS provider provides the following services -

Compute: Computing as a Service includes virtual central processing units and virtual main memory for the Vms that is provisioned to the end- users.

Storage: IaaS provider provides back-end storage for storing files.

Network: Network as a Service (NaaS) provides networking components such as routers, switches, and bridges for the Vms.

Load balancers: It provides load balancing capability at the infrastructure layer.

Advantages of IaaS cloud computing layer

There are the following advantages of IaaS computing layer -

1. Shared infrastructure

IaaS allows multiple users to share the same physical infrastructure.

2. Web access to the resources

Iaas allows IT users to access resources over the internet.

3. Pay-as-per-use model

IaaS providers provide services based on the pay-as-per-use basis. The users are required to pay for what they have used.

Disadvantages of IaaS cloud computing layer

1. Security

Security is one of the biggest issues in IaaS. Most of the IaaS providers are not able to provide 100% security.

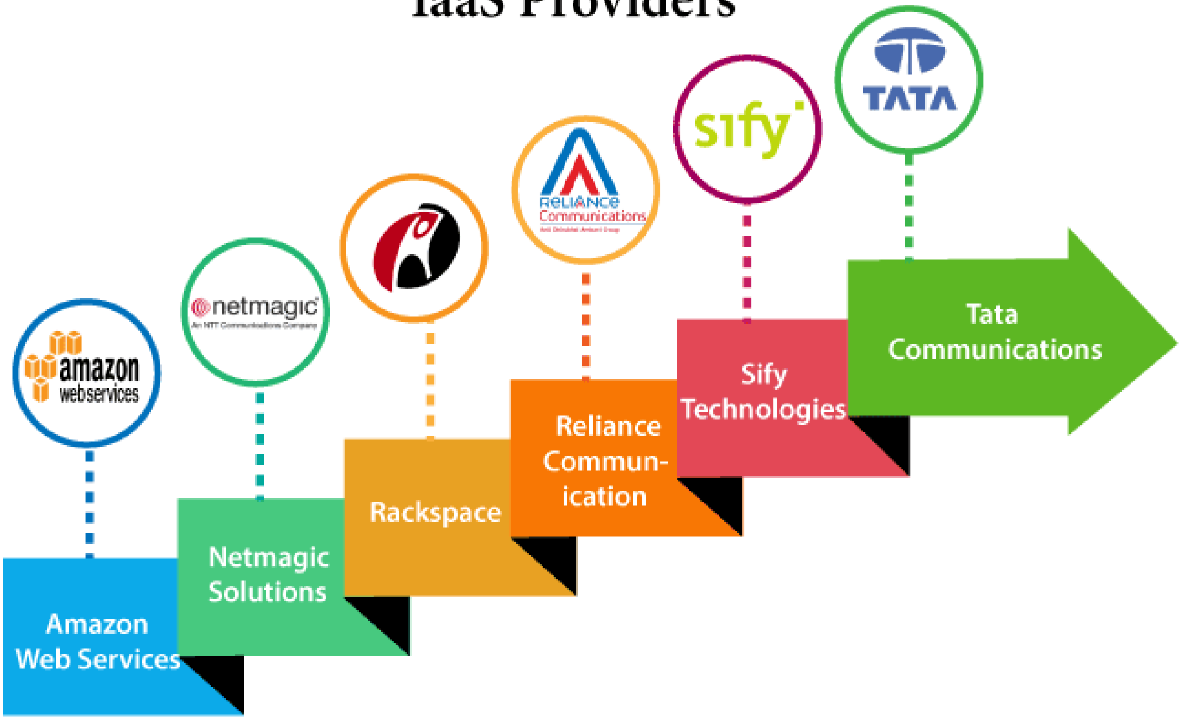
2. Maintenance & Upgrade

Although IaaS service providers maintain the software, but they do not upgrade the software for some organizations.

3. Interoperability issues

It is difficult to migrate VM from one IaaS provider to the other, so the customers might face problem related to vendor lock-in.

IaaS Providers


amazon webservices
netmagic An NTT Communications Company
RELIANCE Communications
sify
TATA
Amazon Web Services
Netmagic Solutions
Rackspace
Reliance Commun-ication
Sify Technologies
Tata Communications