Module-5

Understanding Services and Applications

Topics

- Understanding Service Oriented Architecture
- Moving Applications to the Cloud
- Working with Cloud-Based Storage
- Working with Productivity Software
- Using Webmail Services
- Communicating with the Cloud
- Using Media and Streaming

Understanding Service Oriented Architecture (SOA)

SOA describes a standard method for requesting services from distributed components and managing the results.

The clients requesting services, the components providing the services, the protocols used to deliver messages, and the responses can vary widely.

So, SOA provides the translation and management layer for a client to obtain desired services.

With SOA, clients and components can be written in different languages and can use multiple messaging protocols and networking protocols to communicate with one another.

SOA provides the standards that transport the messages and makes the infrastructure to support it possible.

SOA provides access to reusable Web services over a TCP/IP network.

The environment it creates is a virtual message-passing system with a loose coupling between clients and services.

SOA components are modular and can be easily added to a business process. This makes it a good modelling tools for Computer Aided Software Engineering (CASE).

Introducing Service Oriented Architecture

SOA is a methodology for providing platform and language-independent services for use in distributed applications.

A service is a repeatable task within a business process, and a business task is a composition of services.

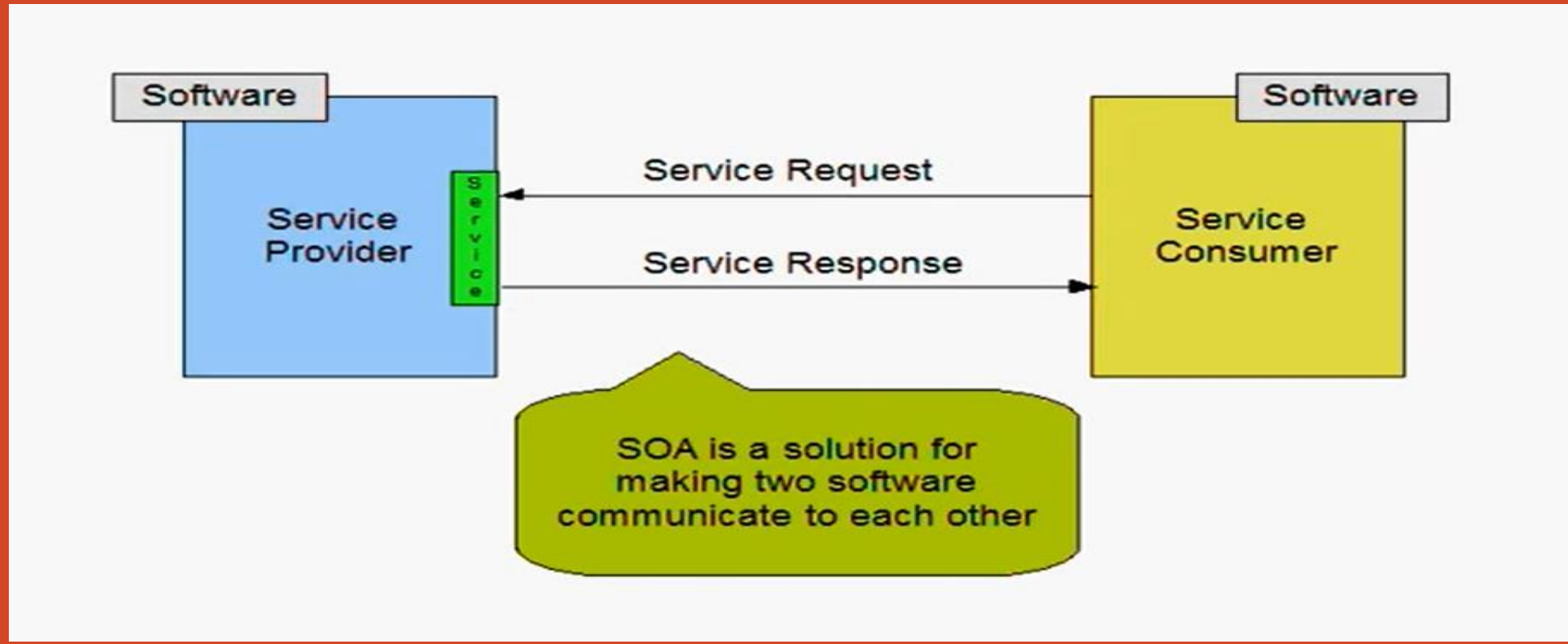
SOA describes a message-passing taxonomy for a component-based architecture that provides services to clients upon demand.

Clients access a component that complies with SOA by passing a message containing metadata to be acted upon in a standard format.

The component acts on that message and returns a response that the client then uses for its own purpose.

Implementations of SOA employ middleware software to play the role of transaction manager (or broker) and translator.

That middleware can discover and list available services, as well as potential service consumers, often in the form of a registry


Software
Service Provider
Service
Service Request
Service Response
Software
Service Consumer
SOA is a solution for making two software communicate to each other

The message presents data to the service, and the service responds.

The most commonly used message-passing format is an XML document using Simple Object Access Protocol (SOAP).

Other formats are Web Services Description Language (WSDL), Web Services Security (WSS), and Business Process Execution Language for Web Services (WS-BPEL).

SOA provides the framework which allows clients of any type to engage in a request-response mechanism with a service.

The manner in which messages are passed in SOA, or in which events are handled, are referred to as their contract.

SOA requires the use of an orchestrator or broker service to ensure that messages are correctly transacted.

When you combine Web services to create business processes, the integration must be managed.

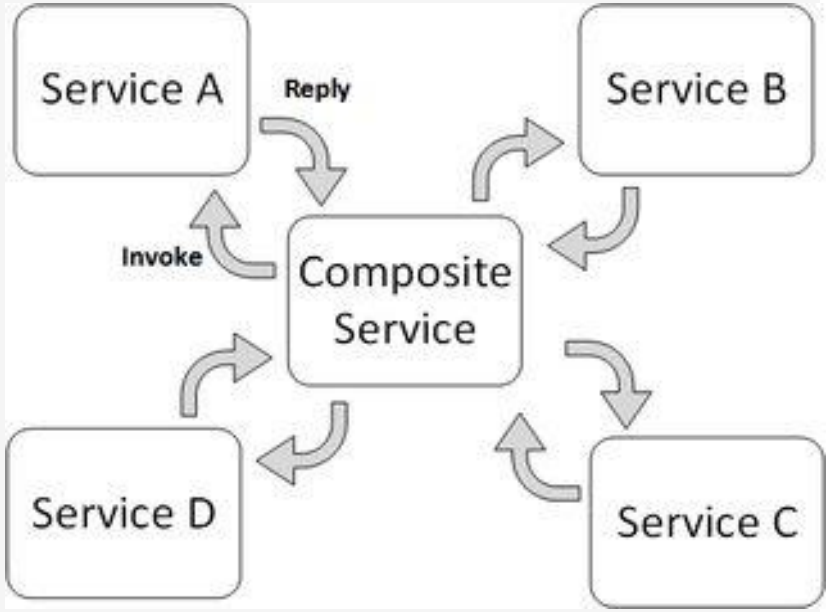
Two main methods are used to combine Web services:

- Orchestration .
- Choreography.

Orchestration

In orchestration, a middleware service (the orchestrator) centrally coordinates all the different Web service operations, and all services send messages and receive messages from the orchestrator.

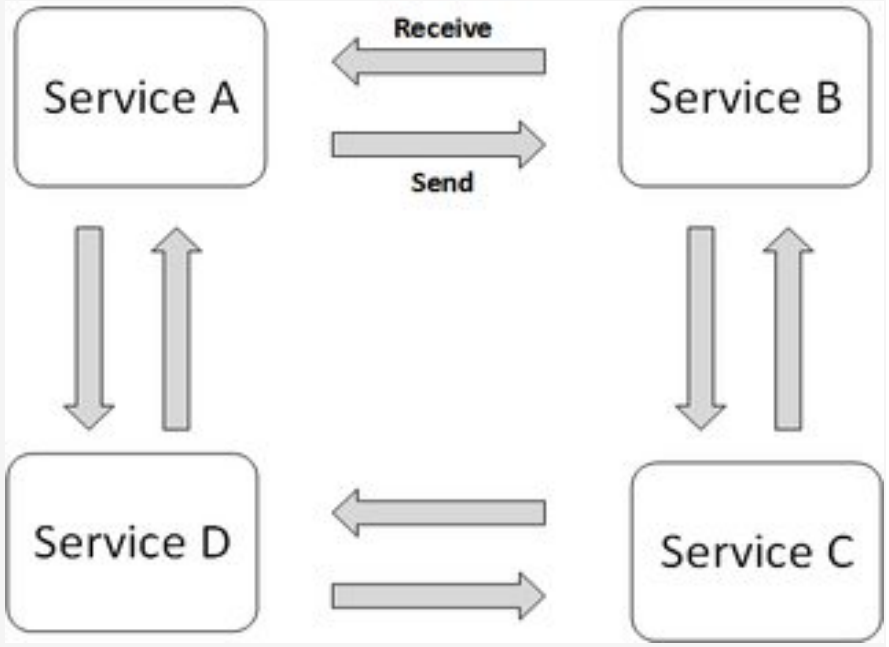
The logic of the compound business process is found at the orchestrator alone.



Choreography

A compound business process that uses choreography has no central coordination function.

In choreography, each Web service that is part of a business process is aware of when to process a message and with what client or component it needs to interact with.



Choreography is a collaborative effort where the logic of the business process is pushed out to the members who are responsible for determining which operations to execute and when to execute them, the structure of the messages to be passed and their timing, and other factors.

Most mature SOA implementations favour orchestration over choreography for a number of reasons.

Orchestration uses a centralized approach to execute the decisions and is more crystal clear and has better control.

With orchestration a single central service manages the various processes, and changes to the business logic can be made in that one location.

The integration of Web services into the architecture is easier because these services don’t need to know anything about the business process.

Centralizing the business logic makes it easier to put error handling mechanisms .

One way of performing orchestration is through the use of an Enterprise Service Bus or ESB.

An ESB provides a middleware software layer for event management with a messaging infrastructure.

Event-Driven SOA or SOA 2.0

SOA 2.0 is an extension of the SOA to respond to events that occur as a result of business processes or perhaps cause and influence a business process.

Eg: In a business process, sales at a certain Web site are processed. If the business process recognizes the rate at which sales are occurring, it could perform an analysis to determine what events might influence the buying decision.

Event-driven SOA is meant to address such sort of analysis.

SOA 2.0 can allow low-level events to trigger a business process, correlate events with information contained in the SOA design, inhibit a business process if the appropriate events don’t appear, or invoke a reaction or response based on a trigger.

A Causal Vector Engine (CVE) with some built-in artificial intelligence must be added to the SOA design to perform these tasks.

Events are analyzed in terms of event sequences, event relationships, and event timing to establish whether a certain condition has occurred.

The CVE then determines how to react to the condition using a set of rules that are built into the system.

CVE systems display events in a console in different contexts so that an observer can analyze the display and take appropriate actions.

A CVE application may include the ability to query event data.

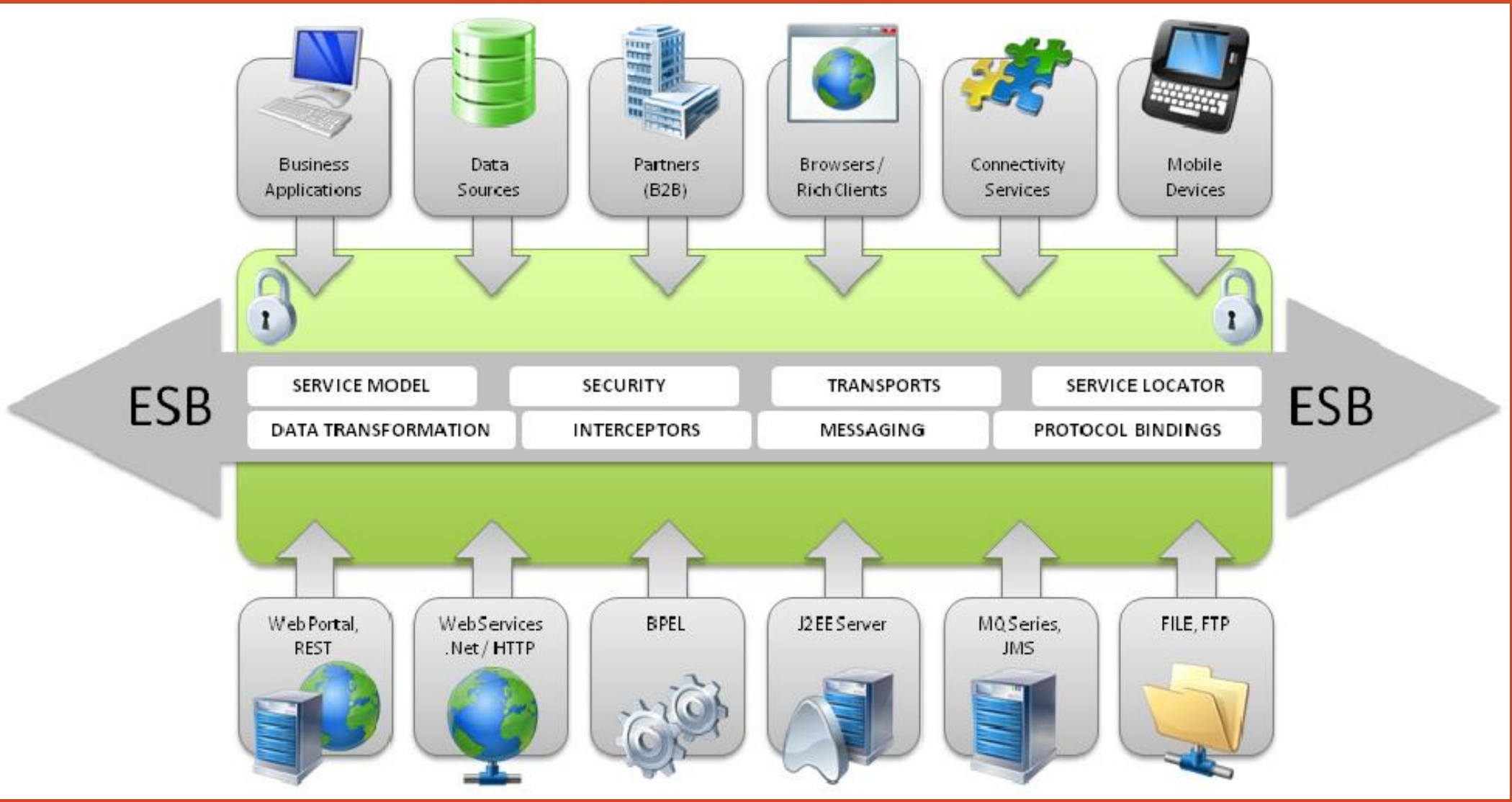
The Enterprise Service Bus

An **enterprise service bus** (**ESB**) implements a communication system between mutually interacting software applications in a service-oriented architecture (SOA)

It is an architectural pattern comprised of a set of network services that manage transactions in a SOA.

An ESB is a set of services that separate clients from components on a transactional basis.

Messages flow from client to component through the ESB, which manages these transactions.


Business Applications
Data Sources
Partners (B2B)
Browsers / Rich Clients
Connectivity Services
Mobile Devices
ESB
SERVICE MODEL
SECURITY
TRANSPORTS
SERVICE LOCATOR
DATA TRANSFORMATION
INTERCEPTORS
MESSAGING
PROTOCOL BINDINGS
ESB
Web Portal, REST
Web Services .Net / HTTP
BPEL
J2EE Server
MQ Series, JMS
FILE, FTP

An ESB plays the role of a transaction broker in SOA, ensuring that messages get to where they where supposed to go and are acted upon properly.

The service bus performs message translation, registration, routing, logging, auditing, and managing transactional integrity.

An ESB may be part of a network operating system or may be implemented using a set of middleware products.

An ESB creates a virtual environment layered on top of an enterprise messaging system where services are advertised and accessed.

Typical features found in ESBs:

Monitoring services aid in managing events.

Process management services manage message transactions.

Data repositories or registries store business logic and aid in governance of business processes.

Data services pass messages between clients and services.

Data abstraction services translate messages from one format to another, as required.

Governance is a service that monitors compliance of your operations with governmental regulation, which can vary from state to state and from country to country.

Security services validate clients and services and allow messages to pass from one to the other.

The difference between a repository and a registry in the context of a Service Oriented Architecture is subtle. Repositories and registries are both data stores, but a repository stores references to the components of the SOA, their source code, and linking information that are used to provide SOA services. An SOA registry contains references to rules, descriptions, and definitions of the services—that is, the metadata of the components.

Service Catalogs

Finding any particular service and locating the service's requirement in a large SOA implementation can involve a large amount of network system overhead.

To aid in locating services, SOA infrastructure often includes a catalog service.

This service stores information on the following, among other things:

- What services are available.
- How to use a service
- Which applications are related to a particular service (dependencies)
- How services relate to one another
- Who owns the service and how a service is modified
- The event history of a service, including service levels, outages, and so on
- The nature of service contracts

Service catalogs are dynamic and under constant modification. Catalog servers have these features:

- *Standalone Catalog Servers* serving a single site.
- *Global Catalog Service* where two or more catalog servers are merged to include several sites.
- *Federated Catalog Service* where two or more global catalog servers have access to one another's information through a trusted query relationship.

Defining SOA Communications

Message passing in SOA requires the use of two different protocol types:

- the data interchange format
- the network protocol that carries the message.

A client connected to an ESB communicates to a component (or service) over a network protocol such as HTTP, REST or JMS.

Messages are most often in the form of the XML.

REST-Representational State Transfer
JMS-Java Message Service
XML-eXtensible Markup Language
SOAP-Simple Object Access Protocol

SOAP is a messaging format that use XML as the message format while relying on Application layer protocols such as HTTP and RPC for message negotiation and transmission.

An ESB may require a variety of combinations in order to support communications between a service consumer and a service provider.

WSDL is one of the most commonly used XML protocols for messaging in Web services.

The most common transport for WSDL is SOAP, and the WSDL file usually contains both XML data and an XML schema.

RPC-Remote Procedure Calls
WSDL-Web Service Description Language

Contained within WSDL are essential objects to support message transfer, including these:

- ✓ The *service* object, a container where the service resides.
- ✓ The *port or endpoint*, which is the unique address of the service.
- ✓ The *binding*, which is the description of the interface (e.g. RPC) and the transport (e.g. SOAP).
- ✓ The *portType*, or interface that defines the capabilities of the Web service, and what operations are to be performed, and the messages that must be sent to support the operation.

- ✓ The *operation* that is to be performed on the message.
- ✓ The *message* content, which is the data and metadata that the service operation is performed on.
- ✓ The *types* used to describe the data, usually as part of the XML schema that accompanies the WSDL.

Messages must have some of the following information contained inside them:

- ✓ *Header*: The header contains the name of the service, service version, owner of the service, and perhaps a responsibility assignment.

- ✓ *Service Type:* Examples of service types include data, business, integration, presentation, and process types.
- ✓ *Functional Specification:* Includes the functional requirements, what service operations or actions and methods must be performed, and the manner in which a service is invoked or initiated.
- ✓ *Transaction Attributes:* A transaction may need to be managed or tracked or include another transaction operated at a specific Quality of Service and under a specific SLA. Security parameters also are part of a transaction's attributes.

Business Process Execution Language

If a message represents an atomic transaction in a Service Oriented Architecture, A level of abstraction up in SOA is the grouping and managing of sets of transactions to form useful work and to execute a business process.

Eg: Business Process Execution Language (BPEL), Web Service Business Process Execution Language(WS-BPEL).

BPEL (Business Process Execution Language) is an XML-based language that allows Web services in a service-oriented architecture (SOA) to interconnect and share data.

BPEL is used to compose, orchestrate, and coordinate business processes with Web services in the SOA model, and it has commands to manage asynchronous communications.

BPEL is an XML-based language that supports the Web services technology stack, including SOAP, WSDL etc.

(WS-BPEL), a language standard for Web service interactions. The standard is maintained by the Organization for the Advancement of Structured Information Standards (OASIS).

Oracle BPEL Process Manager, IBM WebSphere Business Integration Server Foundation,AquaLogic

BPEL is a meta-language(whose statements refer to statements in another language) comprised of two functions:

- Executable commands for Web services and clients,
- Internal or abstract code for executing the internal business logic that processes require.

It includes techniques for error handling and scopes transactions. It uses Web services for standards and to assemble and decompose processes.

Business Process Modelling

SOA was created by the industry to solve a problem: how to make disparate, diverse, and distributed services talk to disparate and diverse clients.

Various modelling tools have been developed to support SOA development and optimization, system and process management, change and life-cycle management.

Commonly encountered system models include the following:

- ✓ *Unified Modelling Language (UML):* UML creates graphical representations of software systems in the form of a set of diagram types.
- ✓ *XML Metadata Interchange (XMI):* Used to exchange metadata using the Extensible Markup Language (XML).

- ✓ *Systems Modelling Language (SysML):* Is an open-source extension of the part of the UML system . Smaller, focused, and easier to learn and work with.

- ✓ *Business Process Modelling Notation (BPMN):* It is a methodology for representing business processes as a set of connected visual objects that illustrate workflow in a Business Process Diagram (BPD).

- ✓ *Service-Oriented Modelling Framework (SOMF):* It combines a modelling language with a graphical display of the various SOA components so the system can be viewed as a map of objects and associated relationships.

Service-Oriented Modelling Framework (SOMF)

Eg: IBM's Service-Oriented Modelling and Architecture (SOMA), introduced in 2004.

SOMA reduces services to a set of service objects and breaks down relationships into three components:

❖ the services themselves

❖the service components that make use of those services

❖ the information flows required to interact between them. Flows consist both of processes as well as their internal composition

In SOMA, domains and functional groups are identified, variables that affect processes are analyzed, and component development and object oriented analysis are used to model specific cases.

SOMA is meant to provide information on service and service boundaries, service granularity, and asset analysis.

In SOMF, the modelling is based on the elements of the service life cycle:

- ✓ *Conceptualization*: Defining the processes that will be supported
- ✓ *Discovery*: Determining how components are exposed

- ✓ Analysis: Identifying the relationships services have to clients
- ✓ Integration: Implementing the messaging infrastructure that connects clients to components
- ✓ Logical design: Building the system and process logic and determining how orchestration or choreography will be performed
- ✓ Architecture design: Creating the system component architecture and reducing design to software components with specified interfaces
- ✓ System implementation: Building and testing the system components

Four modelling topologies for message passing are used in SOMF:

Circular Topology

Message passing is carried out in a circular fashion.

There is no orchestrator in this system

Each component providing a service is responsible for knowing which message to act on and where to send a message next.

The choreography of the system is maintained at the component.

Hierarchical Topology

Services are arranged in a tree pattern with parent/child relationships.

Messages from one service to another must traverse up the branch of the tree and down another branch from the root until the matching service is found.

Advantage: A well-defined set of relationships, a central location (the root) where logic may reside.

Disadvantage: The overhead of passing a message from one service to another isn't optimized.

Network Topology

A network topology has a many-to-many relationship between services and their clients.

Advantage: The overhead associated with message passing has been minimized.

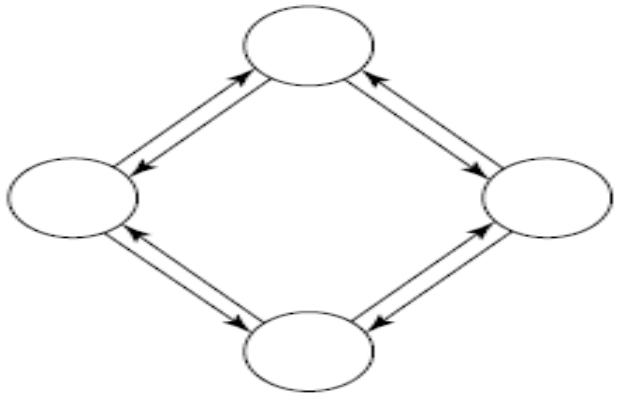
Disadvantage: Considerable overhead built into the system in order to maintain the many links needed.

Star Topology

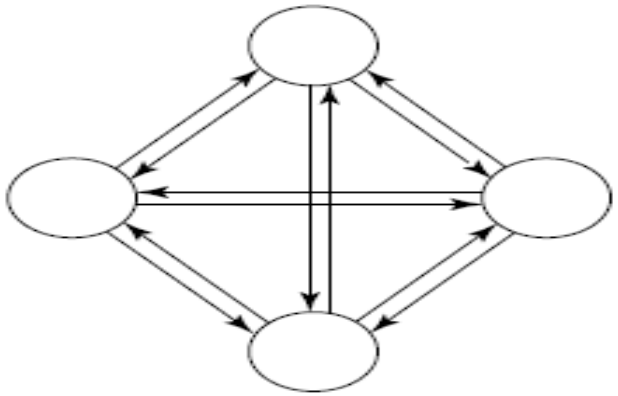
Services are designed to connect through a central service.

Favoured orchestration processes.

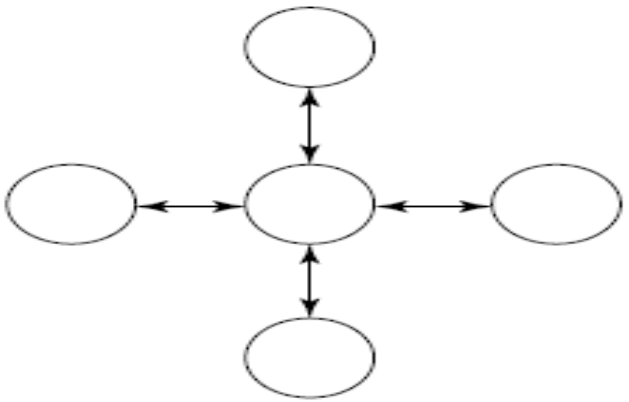
Useful for services that use broadcasting or multicasting services, publish and subscribe, and other related systems.



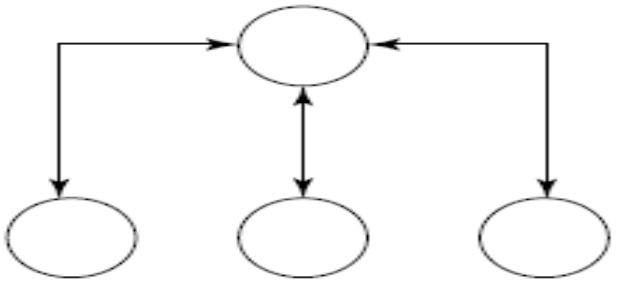
Circular topology



Network topology



Star topology



Hierarchical topology

A service in SOMF that is granular and very narrow in scope is referred to as an *atomic service*.

An atomic service cannot be decomposed into smaller services that provide a useful function.

A collection of services that work together is referred to as a *composite service*.

A composite service is usually organized as a hierarchical topology and is multifunctional.

A collection of composite services that would form a process module is referred to as a *service cluster*.

Service clusters may be composed of both atomic services and composite services.

The SOMF modelling notation has a symbol for each analysis that relates one service to another.

As you build a business process, you add services to the model and connect them in ways that make sense for your workflow.

When the model is complete and optimized, it is reduced to a conceptualized service that relates the business process to the specific implementation chosen.

Managing and Monitoring SOA

Software for monitoring and managing an SOA infrastructure plays an important role in large SOA deployments.

Tools for managing SOAs tend to be multifaceted and run constantly.

Eg:Oracle BPEL Process Manager

IBM WebSphere

HP Software and Solutions OpenView SOA Manager

SOA Management Tools

IBM Tivoli Framework Composite Application Manager

- Specialized in change management and SOA lifecycle development, and it integrates with a WebSphere and other Tivoli systems.

Oracle BPEL Process Manager

IBM WebSphere

HP Software and Solutions OpenView SOA Manager

HP Software and Solutions OpenView SOA Manager

Provides dynamic mapping, monitoring, and optimization of SOA services such as Web services, software assets, and virtual services.

These framework products create a central console with a variety of management views.

Oracle BPEL Process Manager and WebSphere

These are process managers for creating an Enterprise Service Bus.

Configuration and change management present a particular challenge in the area of SOA.

Elements of an SOA infrastructure can be highly distributed and therefore require good discovery mechanisms.

These environments also are highly virtualized.

SOA Security

SOA is subject to attack in all the traditional ways that network traffic is hijacked, spoofed, redirected, or blocked.

Because SOA eliminates the use of application boundaries, implementing just application level security will not be effective.

SOA security tools: Cisco's Application Oriented Networking, Citrix's NetScaler 9.0

To address SOA security, a set of OASIS standards was created, which includes the following:

- Security Assertion Markup Language (SAML)
- WS-Security (WSS)
- WS-SecureConversion
- WS-SecurityPolicy
- WS-Trust

The Open Cloud Consortium

It is an organization that supports the development of standards for cloud computing and for interoperating with the various frameworks.

OCC working groups perform these functions:

- ✓ They develop benchmarks for measuring cloud computing performance. Their benchmark and data generator for measuring large data clouds is called MalStone.

- ✓ They provide testbeds that vendors can use to test their applications, including the Open Cloud Testbed and the Intercloud Testbed.
- ✓ They support the development of open-source reference implementations for cloud computing. MapReduce is Google's patented software framework that supports large distributed data sets organized by the Google File System (GFS) accessed by clusters of computers.
- ✓ They support the management of cloud computing infrastructure for scientific research.

2. Moving Applications to the Cloud

Moving some applications fully or partly from a local or on-premises installation benefit from cloud deployment, and the cloud enhances some features.

The process for determining whether, what, and when to move your applications to the cloud involves an analysis of what critical features of the application need to be supported.

A particular cloud service provider is finalized based on our application's critical features they support.

Application porting methods:

- Physical hardware is eliminated by moving the entire application to the cloud.
- A system is essentially cloned to the cloud.

Factors such as access to data, latencies, data security etc limit application porting abilities.

When you move an application to the cloud, you must use the APIs of your particular cloud service provider.

There are APIs for each of the types of cloud services: infrastructure, software services, and applications.

These APIs are generally not interoperable.

Applications in the Clouds

Applications in the cloud, must account for system abstraction and redirection, scalability, a whole new set of application and system APIs, LAN/WAN latencies, and other factors that are specific to one cloud platform or another.

Any application can run either completely or partially in the cloud.

A developer should analyze whether his application's function is best served by cloud or local deployment

It depends upon the application's attributes that has to preserve or enhance, and how locating those services in the cloud impacts those attributes.

The location of an application or service plays a fundamental role in how the application must be written.

An application or process that runs on a desktop or server is executed coherently, as a unit, under the control of an integrated program.

An action triggers a program call, code executes, and a result is returned and may be acted upon. Taken as a unit, "Request => Process => Response" is an atomic transaction.

ACID Principle: The properties necessary to guarantee a reliable transaction in databases and other applications and the technologies necessary to achieve them. The acronym stands for:

- *Atomicity*: Defines a transaction as something that cannot be subdivided and must be completed or abandoned as a unit.
- *Consistency*: States that the system must go from one known state to another and that the system integrity must be maintained.
- *Isolation*: States that the system cannot have other transactions operate on data that is currently being processed by a transaction.
- *Durability*: States that the system must have a mechanism to recover from committed transactions when required.

An application that runs as a service on the Internet has a client portion that makes a request and a server portion that responds to that request.

The request has been decoupled from the response because the transaction is executing in two or more places.

In order to create a stateful system in a distributed architecture, a transaction manager or broker must be added.

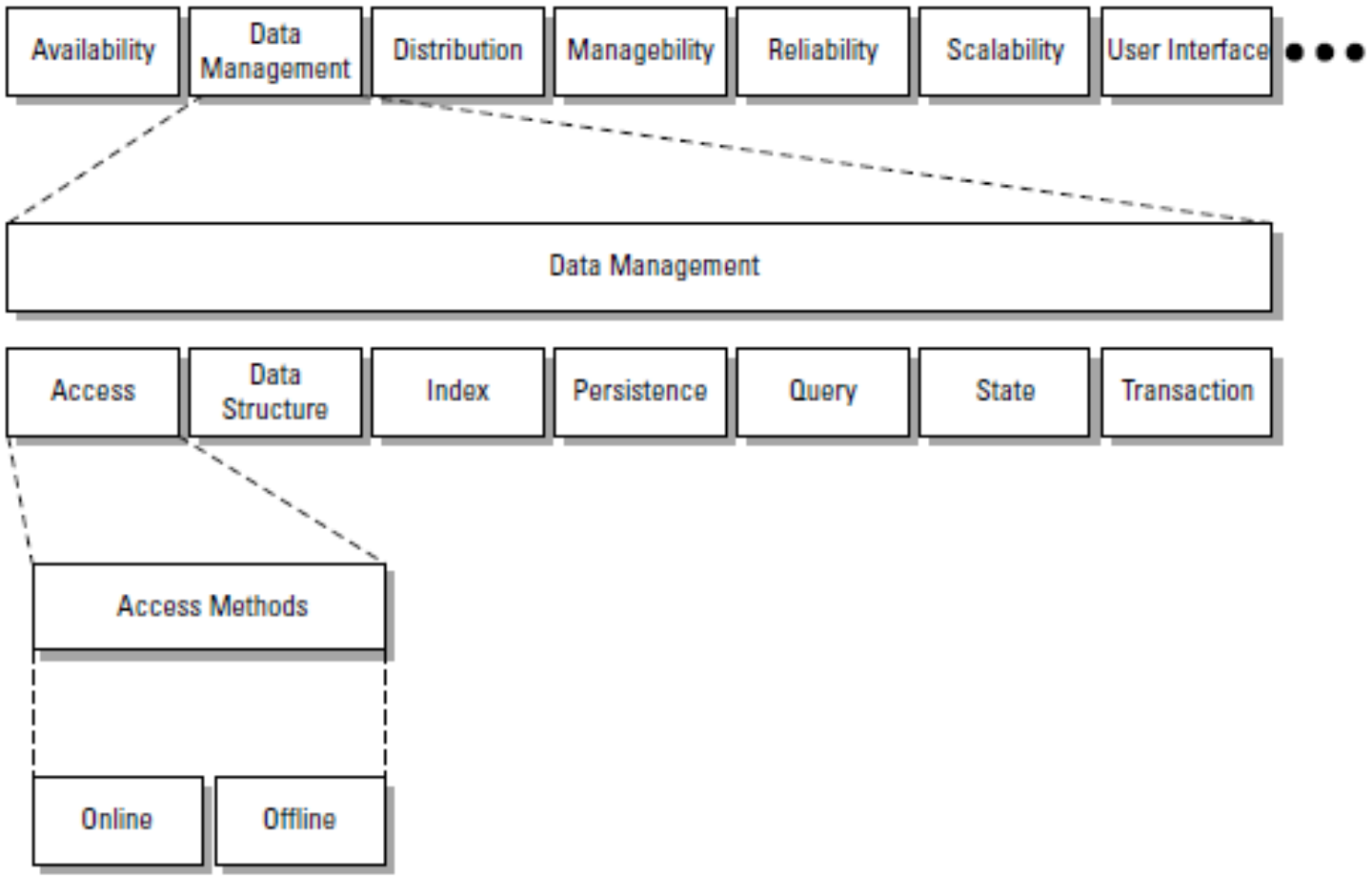
When applications get moved to cloud physical systems become virtualised. The place where the program execution occurs is different every time.

Functionality Mapping

During Cloud deployment, some applications can be successfully ported to the cloud, but some fails.

Understanding whether an application can benefit from cloud deployment requires to deconstruct your application's functionality into its basic components and identify which functions are critical and can be supported by the cloud.

An attribute map is created to expose critical functionality.



Situation Analysis

Consider the functionality mapping:

Transaction system-> Data Management-> Access-> Access Methods-> Online or offline?

Online or offline access methods determine the nature of your application's interaction.

Allowing both online and offline data access requires to create a hybrid application with a cloud component and a local component.

If only online access required, then a cloud-based data store is what required. The application could be entirely in the cloud and browser-based.

If only offline access required, provisions for local data stores is mandatory. Still, client-side support is needed.

Application Attributes

Various levels of application attributes needs to be considered in the analysis of an application's suitability that has to be ported to the cloud.

Examples of first-level application attributes are: Application, Availability, Costs, Data Management, Maintenance, Scalability, Security, User Interface

Each cloud platform also has its own set of attributes that need to be mapped.

Application Attributes

First Level	Second Level	First Level	Second Level
Application	Abstraction		Implementation
	Architecture		Language/locale
	Configuration		Monitoring
	Interoperability		Operations
	Modularity		Staffing
	Object model		Startup/recovery
	Reusability		Tools
Availability	Caching	Scalability	Caching
	Fault management		Expertise
	Geographic location		Licensing
	Pooling		Lifecycle management
	Resource access		Load balancing
	Reliability		Replication
	Uptime		Scale up or out
Costs	Development		
	Resources		Staging

The key drivers for applications that benefit from deployment to the cloud are those that meet these criteria:

- Are not mission critical
- Are not core business functions
- Do not have sensitive data to protect
- Tolerate high network latencies or low network bandwidth
- Are legacy applications with no particular competitive advantage
- Are based on industry standard technologies
- Do not need to be customized
- Are mature enough and understood well enough to be successfully ported to the cloud

Cloud Service Attributes

We want to match up application attributes to these key cloud service attributes:

- Applications
- Core services
- Infrastructure
- Platform features
- Storage

Each provider has a unique solution, uses its own APIs, and provides unique services.

Therefore, each cloud provider needs separate developer skills, and integration between clouds would be a major a routine task.

System Abstraction

The cloud turns physical systems into virtual systems.

Organizations choose to deploy systems to the cloud entirely by recreating the essential parts of their process and thus eliminating infrastructure.

The manual and local processing can be replaced by cloud deployment. It helps to reduce the use of local infrastructure.

This enables fast and efficient global data access, optimized load balancing, scalability and proper resource management.

Cloud Bursting

Many cloud deployments are hybrid applications: a part of the application is on a local system, and another part is in the cloud. This is because the cloud can serve excess capacity at times of high volume. This type of hybrid has been called *Cloud Bursting*.

Eg: Transaction processing systems such as reservations systems. There is a certain low level of background transactions occurring at any time. But at certain times, events trigger high demand.

Most systems built to perform cloud bursting have a simple underlying design: clone the local system in the cloud.

Mostly, the cloud portion of the system stay idle, but when the activity grows, the system copy in the cloud picks up the extra activity and, when necessary, provisions extra resources.

Applications relying on dynamic data driven content must have a transaction manager in the system to perform synchronization between your on-premises and cloud-based Web servers.

Applications and Cloud APIs

The nature of a cloud provider's Cloud API will impact the movement of an application to the cloud and affect how its features operate.

Cloud APIs are the Application Programming Interface to functions that exchange information in and with the cloud, request supported operations, and provide management and monitoring functions for applications running in the cloud.

Each cloud vendor has its own specific API; relying on REST or SOAP, or on both.

Each API serves specific purpose required by that vendor's infrastructure and service.

The cloud API contains the authentication and authorization mechanisms needed to access cloud services.

Each layer of a cloud application has its own specific API as well.

The decision to move an application to the cloud requires an analysis of various vendor lock-in.

Depending upon the nature of applications, it might be very easy or nearly impossible to port to any other cloud technology.

3. Working with Cloud-Based Storage

Cloud Storage-Definition

The characteristics of cloud storage include network access (most often through a browser), on-demand provisioning, user control, and most often adherence to open standards so that cloud storage may be OS-neutral and file-system-neutral.

Storage devices may be broadly categorized as either *block storage devices or file storage devices*.

Cloud storage devices can be either block or file storage devices.

Block Storage Device

A block storage device exposes its storage to clients as Raw storage that can be partitioned to create volumes.

It is up to the OS to create and manage the file system. Data is transferred in blocks.

Block storage devices offer faster data transfers, but impose additional overhead on clients.

File Storage Devices

File server storage is most often found in the form of a Network Attached Storage (NAS) device.

NAS exposes its storage to clients in the form of files, maintaining its own file system.

File-oriented storage devices are generally slower, but require less overhead from attached clients.

Provisioning Cloud Storage

Cloud storage may be broadly categorized into two major classes of storage: *unmanaged and managed storage.*

Unmanaged Cloud Storage

The most basic service that online storage can serve is to provide disk space on demand.

In unmanaged storage, the storage service provider makes storage capacity available to users, but defines the nature of the storage, how it may be used, and by what applications.

The storage is preconfigured for you, you can’t format as you like, nor can you install your own file system, or change drive properties such as compression or encryption.

The user doesn't have much options to manage.

It is presented to a user as if it is a ready-to-use disk drive.

Storage was offered as fixed online volumes. Disk space is made available to users as a sized partition.

It is reliable, relatively cheap to use, and particularly easy to work with.

Applications using unmanaged cloud storage are Software as a Service (SaaS) Web services

Most user-oriented software such as file-sharing and backup consume unmanaged cloud storage.

Applications using unmanaged cloud storage are Software as a Service (SaaS) Web services.

The simplest of these unmanaged cloud storage services falls into the category of a file transfer utility.

You can upload files to the service where that file is stored and made available to you for downloading from another location.

Eg:Dropbox, 4Shared, MediaFire, RapidShare

Managed Cloud Storage

It is mainly meant for developers and to support applications built using Web services.

This type of storage is provisioned and provided as a raw disk.

The user has to partition and format the disk, attach or mount the disk, and make the storage assets available to applications and other users.

In a managed cloud storage system, the user provisions storage on demand and pays for the storage using a pay-as-you-go model.

This type of system is meant to support virtual cloud computing as the virtualized storage component of that system.

Applications using managed cloud storage are Infrastructure as a Service (IaaS) Web services.

Eg: Amazon’s Simple Storage System (S3), Rackspace Cloud, Google Storage for Developers

Creating Cloud Storage Systems

The Internet was designed to be a fault-tolerant network.

Paths between endpoints are redundant, message transfer is packetized, and dropped or lost packets can be retransmitted and travel different paths.

Networks are redundant, name servers are redundant, and overall the system is highly fault tolerant.

These features help make cloud-based storage systems highly reliable, particularly when multiple copies of data are stored on multiple servers and in multiple locations.

Failover can involve a system simply changing the pointers to the stored object's location.

Availability Zones are created with sets of systems that are isolated from one another with the aim that instances in different availability zones shouldn't fail at the same time.

Virtual Storage Containers

Delivering effective cloud storage solutions requires the use of a virtual storage container, which allows a tenant to perform storage, consistent with the capabilities of the storage system used.

It provides easy, on-the-go storage assignment and the high disk utilization rates that are required in a multi-tenancy storage system to create high-performance cloud storage systems. LUNs, files, and other objects are then created within the virtual storage container.

When a tenant is granted access to a virtual storage container, he performs standard disk operations such as partitioning, formatting, file system modifications, and CRUD (Create, Read, Update, and Delete) operations as desired.

Data stored in a virtual storage container may be stored in chunks or buckets, or it may be stored in containers that are in a hierarchical relationship.

To make cloud storage data discoverable on a TCP/IP network requires the objects be assigned a URI (Uniform Resource Identifier) and the relationship between objects and their metadata are specified.

To secure virtual storage containers, the objects must carry a set of security attributes that protect a tenant's data from snooping, denial of service attacks, spoofing, inappropriate deletion, or unauthorized discovery.

The main mechanism for securing one user's virtual storage container from another is to assign an IP address to it and then bind it to a separate VLAN connecting storage to the tenant (host).

Traffic flowing over the VLAN is encrypted, and the tenant is carefully authenticated by the system.

Usually, data sent over the VLAN is compressed to improve data throughput over a WAN connection.

Important considerations while evaluating cloud storage solutions are:

- Client self-service
- Strong management capabilities
- Performance characteristics such as throughput
- Appropriate block-based or file-based storage protocol to support your systems
- Seamless maintenance and upgrades

Exploring Cloud Backup Solutions

Backup Types

Backups may be categorized as belonging to one of the following types:

Full system or image backups: Creates a complete copy of a volume, including all system files, the boot record, and any other data contained on the disk.

Point-in-time (PIT) backups or snapshots: The data is backed up, and then every so often changes are amended to the backup creating an incremental backup. Lets you restore your data to a point in time.

Differential and incremental backups: During an incremental backup, any changed files since last change are copied to the backup media. In a differential backup, all of the changed files since the last full backup are copied by the backup software.

Open file backup: creates a complete file backup after all the transactions have been processed.

Data archival: The migration of data that is no longer in use to secondary or tertiary long-term data storage for retention. Useful for legal compliance or to provide a long-term historical record.

Cloud Backup Features

- ✓ Logon authentication.
- ✓ High end-to-end encryption of data transfers.
- ✓ Lossless data compression to improve throughput.
- ✓ Automated, scheduled backups.
- ✓ Fast backup (snapshots).
- ✓ Data versioning with the ability to retrieve historical versions of files from different backups.
- ✓ Multiplatform support (Windows, Macintosh, Linux/Unix).
- ✓ Bare file/folder restore.

- ✓ Adequate bandwidth and perhaps scalable bandwidth options.
- ✓ Web-based management console with ease-of-use features.
- ✓ 24x7 technical support.
- ✓ Backed up data set validation.
- ✓ Logging and reporting of operations.
- ✓ Open file backups of mission-critical transactional systems.
- ✓ Multisite storage or replication, enabling data failover

Cloud Attached Backup

Some hardware-based solutions are available for backing up our systems to cloud-based storage.

CTERA's Cloud Attached Storage is meant for the Small and Medium Business (SMB) market, branch offices, and the Small Office Home Office (SOHO) market.

It has the attributes of a NAS (Network Attached Storage)

Once the system settings, user accounts, and backup options are configured, the system runs automated backup copying and synchronizing of your data with cloud storage. Backed up data may be shared between users.

CTERA cloud backup provides a solution that optimizes the backup based on bandwidth availability.

It performs incremental backups from the server, compressing and encrypting the data that is transmitted.

The CTERA server performs the backups of clients without requiring any client-based software.

Clients have browser-based access to the backups or can locally access files using CTERA's "Virtual Cloud Drive" network drive.

A scaled-down version of the CTERA server called the CTERA CloudPlug converts a USB/eSATA drive and your Ethernet network and turns the hard drive into a NAS server.

Cloud Storage Interoperability

Cloud Data Management Interface (CDMI)

An example of an open cloud storage management standard is the Storage Networking Industry Association's Cloud Data Management Interface (CDMI).

CDMI works for interoperation between different cloud systems, whether on public, private, or hybrid cloud systems.

CDMI includes commands that allow applications to access cloud storage and create, retrieve, update, and delete data objects

In CDMI, the storage space is partitioned into units called containers

It supports data object discovery, enables storage data systems to communicate with one another, and provides for security using standard storage protocols, monitoring and billing, and authentication methods.

CDMI uses the same authorization and authentication mechanism as NFS (Network File System) does.

It provides an interface through which applications can gain access to the storage objects in a container over the Web.

CDMI features includes access controls, usage accounting, and the ability to advertise containers so that applications see these containers as if they are volumes.

Open Cloud Computing Interface (OCCI)

An open standard API for cloud computing infrastructure systems.

Created jointly by SNIA and the Open Grid Forum.

The OCCI interface standard is based on the Resource Oriented Architecture (ROA) and uses the URI definition

OCCI specifies a service life cycle. In a service life cycle, a client (service requestor) instantiates or invokes a new application and through OCCI commands provisions its storage resources, manages the application's use, and then manages the application's destruction and the release of its cloud storage.

The ability to provide storage on demand from a storage pool is referred to as thin provisioning.

- CDMI and OCCI are meant to interoperate, and CDMI containers can be accessed through a data path and over other protocols.
- A CDMI container can be exported and then used as a virtual disk by Virtual Machines in the cloud.
- The cloud infrastructure management console can be used to attach exported CDMI containers to the VM that is desired.
- CDMI exports containers so the information that is obtained from the OCCI interface is part of the exported container.
- OCCI also can create containers that are interoperable with CDMI containers.
- These export operations can be initiated from either the OCCI or CDMI interfaces, with similar results.

4. Working with Productivity Software

Productivity Software

Anything you do on the computer that is faster and more productive than you could do any other way can be termed productivity software.

A cloud-based productivity software is user-facing software that creates work product in the form of documents or files using resources accessed over a WAN.

Major players that have office suits:

- Acrobat.com
- Glide Digital
- Google Docs
- Microsoft Office Web Apps
- ThinkFree Office
- Zoho Office Suite
- HubSpot Free Sales Tools
- Todoist

Characteristics of productivity software

- ✓ User-centric and user-facing
- ✓ Static features
- ✓ Ease of use
- ✓ Standards for data interchange
- ✓ Modular interactivity
- ✓ Inter Applications Communications (IAC)

Online Office Systems

Benefits of using a cloud computing office suite:

- ✓ Generally lower costs for users
- ✓ Platform independence because the software runs inside a browser and is universally available
- ✓ Reduced maintenance costs because of fewer hardware requirements
- ✓ Centralized software patches and updates, whereby users have access to the latest version of the software
- ✓ Easier document sharing due to standard formats, and often easier document sharing for group collaboration

Disadvantages:

- ✓ An Internet connection is required to access your documents.
- ✓ Performance can be slow, particularly when you have a slow Internet connection.
- ✓ Online productivity applications generally have fewer features than their shrink-wrapped competitors.
- ✓ Documents stored in the cloud may not be secure and are certainly not under a user's full control.
- ✓ Data in the cloud can be lost and must be managed and backed up.
- ✓ Documents created with desktop applications may not be fully compatible with cloud-based applications.

5. Using Webmail Services

Exploring the Cloud Mail Services

Browser-based hosted e-mail or “Webmail” is a Software as a Service (SaaS) application.

Webmail was one of the first cloud computing applications to emerge.

The first of the free hosted Webmail services to emerge was Hotmail.

The number of active use of accounts may be obtained by examining the number of visits (hits) that the different

Web sites get. A hit can be measured relatively accurately by looking at the DNS server logs.

Major players are:

- Google Gmail
- Mail2Web
- Windows Live Hotmail
- Yahoo! Mail

Working with Syndication Services

A syndication service is a way for people to send messages to a group of people; it's a form of published e-mail.

To receive syndicated content, you must opt into the system and subscribe to the "feed" from one of the many content management system services.

You can read RSS(Really Simple Syndication) and Atom formatted content inside special applications called newsreaders, or "readers," as well as in many Webmail applications.

After you subscribe to a feed, the reader uses the link provided to download content from a site.

The RSS and Atom Protocols

Two technologies are behind most of the syndicated content being used on the Internet: RSS and Atom.

RSS, stands for Really Simple Syndication.

A typical RSS document or feed contains text and metadata that can be used to indicate publication dates, authors, keywords, and more.

RSS uses an XML file format and the concept of an RSS world or module.

Several modules exist that are XML namespaces, including Ecommerce RSS 2.0, Media RSS 2.0, and OpenSearch RSS 2.0 modules.

An alternative version of XML syndication called the Atom Publishing Protocol was developed by the IETF.

Atom has some structural differences with RSS, but is similar in approach and technology.

Most of the major browsers support RSS and Atom.

When you view a syndication content management application, the aggregators tend to list feeds by content.

This is possible because a feed contains keywords in its metadata.

Newsreaders

Most major Web sites support Web feeds, either as a consumer(browser or the reader applications) or a provider(a Web site or aggregator service).

A browser that is a consumer of RSS feeds shows you an RSS icon in the browser.

Clicking the icon opens a dialog box asking if you want to subscribe to the feed.

By subscribing, you add your favorites and is updated regularly.

There is a class of applications whose entire purpose is displaying RSS content.

Best-known newsreaders are: Google Reader, Bloglines, Newsgator Online

News Aggregators

A news aggregation Web site is one that relies on collecting the syndicated content from other Web sites in a form that can be viewed together.

Most are theme based and display syndicated content with short descriptions and stories mixed in.

Web syndication allows a Web site to receive content from other sites, which makes the site score higher with search engines.

Aggregation also can be done within software creating a custom Web page. Eg: iGoogle, My Yahoo! account.

6. Communicating with the Cloud

Exploring Instant Messaging

Instant messaging builds on the concept of Internet chat to add a more immediate response to text messages between individuals and groups.

Online chat is aimed at exchanging information between people in a forum or multiuser environment, instant messaging is most often a peer-to-peer communication.

A good description of IM is that it is “near-real-time” communications.

Instant Messaging Clients

Use of instant messaging applications are very popular: Operating systems, Webmail services, video games, enterprise messaging systems, and many other applications install IM clients.

Eg: AOL Instant Messenger, Yahoo! Instant Messenger, Windows Live Instant Messenger, Gmail's embedded IM

Instant messaging is as popular in business as many organizations use it as a second kind of e-mail.

This category of software is referred to as Enterprise Instant Messaging (EIM).

Servers of this type support IM, e-mail, document stores, and other features.

Instant messaging applications have the following forms:

- ✓ Discrete instant messaging desktop clients
- ✓ Enterprise messaging services with IM capabilities
- ✓ Browser-based IM clients (often in conjunction with Webmail)
- ✓ Mobile device clients, either embedded and client specific or mobile browser-based

IM adds the following features:

- ✓ Directed messages
- ✓ Rich contact management
- ✓ Message logging and archiving
- ✓ File transfers
- ✓ E-mail integration
- ✓ Voice and video feeds
- ✓ Connections to social media

Instant Messaging Interoperability

Some IM clients use peer-to-peer messaging.

IM software aimed at businesses use a client/server architecture based on their message server products.

Interoperability between different instant messaging clients has been made possible with several instant messaging protocols.

Micro-blogs or Short Message Services

The form of text messaging most widely used today is SMS, which stands for Short Message Service.

When a service collects your messages in a conversation, it is called a micro-blog.

SMS limits the size of messages sent to around 150 characters.

SMS was originally developed as part of GSM networks, but it's now in use on CDMA, 3G, and a variety of other proprietary networks.

Eg:Twitter: an SMS service organized into a social network and blog. It is referred to as a form of Internet Relay Chat (IRC).

Exploring Collaboration Technologies

Collaborative softwares are software that enables real-time or near-real-time communication which enables conferencing and workgroup support.

Eg: Lotus Notes, Citrix GoToMeeting, Zimbra Collaboration Suite, Dimdim.

Using Social Networks

Social Networking sites let you create your own small personal Web site, which you give people access to view.

Information on these pages is of the nature of personal profile.

There are tools for leaving comments, chats, alerts when new information is listed, and many other features.

Eg: Facebook, Twitter, LinkedIn

Features of Social Networking sites:

These are the central elements of social networking sites:

- ✓ Personal profile
- ✓ Friends, buddies, or connections
- ✓ Groups
- ✓ Discussions
- ✓ Blogs
- ✓ Widgets

Privacy and Security

- Issues regarding hacking, privacy rights, exchange of the data over a long time period.
- Profile’s security settings needs to be properly configured.

Interaction and Interoperability

Aggregators has appeared that collects information from other sites into a consolidated data store.

Some of these sites also allow you to create a universal personal profile that can be shared with several social networks.

In order to perform aggregation, these applications have to tap into the social network's API, provided that the service is given consent by the user in the form of supplying his username and password for the social networking service.

Aggregator sites may offer some of the following features:

- ✓ Search across multiple sites
- ✓ Tools for managing messages
- ✓ RSS readers for social networking feeds
- ✓ Tools to manage bookmark from multiple sites
- ✓ Tools to track friends, buddies, or connections over time

7. Using Media and Streaming

Understanding the Streaming Process

Streaming media are files that are sent in pieces by a service and played back by a client as the delivery continues.

Streamed material can be live or on-demand.

Live streaming is called progressive streaming or progressive download, while on-demand streaming is from material that is already stored to disk.

Unicast is a stream sent from a service to an individual user.

To stream the contents successfully, the system requires adequate network bandwidth to transfer enough material to support user playback.

The services sending content should be able to deliver the material in sequence or nearly in sequence to the client.

As material is streamed to a client, the material can be buffered to memory and the system can use the sequence number in the packets received to structure playback.

Protocols in Use

This difference between streamed media and transferred media is fundamental in deciding which transfer protocol to use.

For transferred media, the entire file must be transmitted with exactness, thus TCP (Transmission Control Protocol) is the transmission protocol.

In a streamed media scenario, fidelity isn’t a prerequisite, thus UDP (User Datagram Protocol) is the transmission protocol.

Prominent network control protocols are Real-time Transport Control Protocol (RTCP), Real-time Streaming Protocol (RTSP), and Real-time Transfer Protocol (RTP).

All the streaming protocols take raw files and compress them in some way.

For protection, the files may be encrypted prior to transmission and decrypted when they arrive.

The company Encoding.com uses a cloud infrastructure to convert files that you send them to the formats you require.

The conversions are done in the cloud on your uploaded file, and the results are then downloaded to your desktop when complete.

The Cloud Computing Advantages

Cloud computing has some unique features that make it a very suitable platform for large scale audio and video streaming to customers on demand:

- ✓ Access to large scale storage, which enable the storage of large media files and on-demand media libraries.
- ✓ Access to scalable compute engines and network storage that can serve as the streaming server to large audiences.
- ✓ Access to a scalable compute engines that can be useful when you want to perform encoding/decoding or transcoding on media files.
- ✓ Access to content delivery networks or edge systems that can push content out to users based on geographical location.

Audio Streaming

Audio streaming makes much lower demands on network bandwidth.

Players like Real Networks' Real Player, Windows Media Player and Apple QuickTime play video formats as well as audio formats, and all are available as stand-alone players or as browser plug-ins.

Working with VoIP Applications

Voice over IP or VoIP is a set of communication protocols for delivering voice over the Internet.

Some of these services have been migrated to the cloud.

Major VoIP Applications are Skype, Google Voice and Google Talk

VoIP uses additional protocols and standards other than audio streaming.

Most commonly used VoIP standards:

- H.323
- IP Multimedia Subsystem (IMS)
- Media Gateway Control Protocol (MGCP)
- Session Initiation Protocol (SIP)
- Real-time Transport Protocol (RTP)
- Session Description Protocol (SDP)

Video Streaming

Facilities like broadband networks, high-capacity commodity disk drives, low-cost computing power and cloud computing helps video streaming over the Internet a reality.

Eg of video streaming services: YouTube, Flickr, Metacafe

Television in the cloud

Uses cloud to store TV shows and programs and deliver content to viewers on demand.

Eg: Netflix, Apple TV