

Cloud Computing

(PECO8013T)

Unit-IV

Software as a Service (SaaS)

(SaaS)

By:- Dr. D.R.Patil

Cloud Computing (PECO8013T)

Teaching Scheme

Lectures : 03 Hrs./week

Credits : 03

Examination Scheme

Term Test : 15 Marks

Teacher Assessment : 20 Marks

End Sem Exam : 65 Marks

Total Marks : 100 Marks

Prerequisite: Information Security, Distributed Computing, Web programming

Course Objectives:

- To capture the state-of-the-art in Cloud Computing technologies and applications.
- To cover a series of current cloud computing technologies, including technologies for Virtualization, Infrastructure as a Service, Platform as a Service and Software as a Service.

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Understand the fundamental concepts of cloud computing.	L2	Understand
CO2	Explore the virtualization at various layers of cloud infrastructure.	L6	Create
CO3	Analyse various cloud security concerns and mechanisms.	L3	Apply
CO4	Assess the need and then migrate to cloud.	L5	Evaluate
CO5	Explain Hadoop File System and role of HDFS in cloud.	L2	Understand

Syllabus

Course Contents

Unit-I **04 Hrs.**

Introduction to Cloud Computing

What is cloud computing?, Properties & Characteristics, Service models, Deployment models.

Unit-II **08 Hrs.**

Infrastructure as a Service (IaaS)

Introduction to IaaS, Resource Virtualization (Server, Storage, Network).

Unit-III **06 Hrs.**

Platform as a Service (PaaS)

Introduction to PaaS, Cloud platforms & Management (Computation and Storage), Case studies.

Unit-IV **10 Hrs.**

Software as a Service (SaaS)

Introduction to SaaS, Web services, Web 2.0, Web OS.

Syllabus

Unit-V **10 Hrs.**

Hadoop

Hadoop distributed file system, distributed computations with MapReduce, Hadoop's data and I/O building blocks. Hadoop in the cloud.

Unit-VI **05 Hrs.**

Cloud Security

Cloud Security reference model, governance and enterprise risk management, compliance and audit management, information management and data security.

Unit-VII **02 Hrs.**

Migration to Cloud

Cloud models suitable for different categories of users, Considerations for choosing applications suitable for cloud, Different phases to adopt the cloud.

Books

Text Books:

1. Raj Buyya, Christian Vecchiola, S. Selvi, "Mastering Cloud Computing", TMH, 2013.
2. RajkumarBuyya, James Broberg, Andrzej Goscinski, "Cloud Computing: Principles and Paradigms", Wiley India, 2013.

Reference Books:

1. Tom white, "Hadoop: The Definitive Guide", Ed. O'Reilly, 2012.
2. Chuck Lam, "Hadoop in action", Dreamtech Press, 2011.
3. Dr. Kumar Saurabh, "Cloud Computing: Insights into New-Era Infrastructure", 1st Edition, Wiley India, 2011.
4. Anthony T. Velte, "Cloud Computing: A Practical approach", TMH, 2009.
5. Halper Fern, Kaufman Marcia, Bloor Robin, Hurwit Judith, "Cloud Computing For Dummies", Wiley India, 2009.

Scheme

Evaluation Scheme:

Theory:

Continuous Assessment (A):

Subject teacher will declare Teacher Assessment criteria at the start of semester.

Continuous Assessment (B):

Conduction of Term Test

The two Term Tests of 15 marks will be conducted under Continuous Assessment(CA) out of which best performance among the two Term Tests will be considered.

Term Test (TT) (for 15 Marks)

Best of Two (TT-1/TT-2)

End Semester Examination (C):

1. Question paper based on the entire syllabus, summing up to 65 marks.
2. Total duration allotted for writing the paper is 3 hrs.

Executive Summary

- In this presentation, we go through the **Software as a Service Methodology**, examine its benefits and drawbacks and talk about two state-of-art SaaS systems—**Amazon Web Service** and **Google App Engine**
- We also look into **Service Oriented Architecture** powering SaaS applications and its impact on **modern web 2.0 applications**
- Finally, we examine **hybrids** of traditional and SaaS applications

Overview

- What is Software as a Service (SaaS)
- Background
 - Brief history
 - Concept
 - Big picture
 - Related terms
- Computing Today
 - SaaS is everywhere
 - The SaaS Market
- Benefits of SaaS
- Drawbacks of SaaS
 - Robustness
 - Privacy
 - Security
 - Reliability
- Service Oriented Architectures (SOA)
 - Guiding principles of SOA
- Case studies
 - Amazon Web Services (AWS)
 - Google App Engine
- Influence of SOA on Web 2.0 development
 - Zend Framework
- Hybrids of Traditional and SaaS applications
 - Dropbox
 - Microsoft Office
- Summary
- References

What is SaaS?

- **Definition:** Software as a Service (SaaS), a.k.a. on-demand software, is a software delivery model in which software and its associated data are hosted centrally and accessed using a thin-client, usually a web browser over the internet.—Wikipedia
- Simply put, SaaS is a method for delivering software that provides remote access to software as a web-based service. The software service can be purchased with a monthly fee and pay as you go.

What is SaaS?

- Where does the term SaaS come from?
 - The SAAS acronym allegedly first appeared in an article called "Strategic Backgrounder: Software As A Service", internally published in February 2001 by the Software & Information Industry's eBusiness Division
- Multi-tenant architecture
 - Virtualization as a alternative
- Pricing model
 - pay as we go, relatively low cost for user provisioning
- Configuration and customization
 - Easy for application customization
- Accelerated feature delivery
 - It means a much shorter release cycle
- Open integration protocols
 - Typically based on HTTP, JSON, REST, SOAP

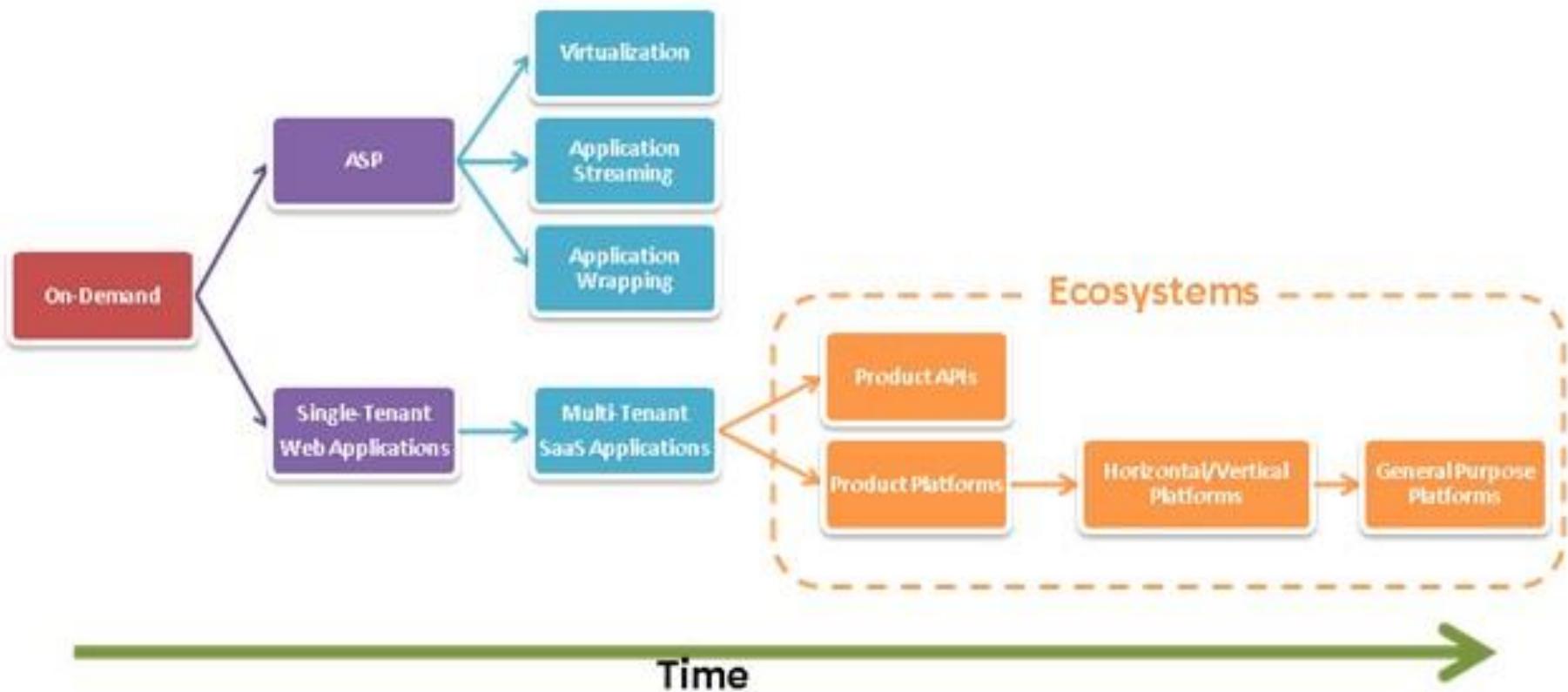
An example

- **Imagine** you are the founder of a start-up company and you need to deal with tons of new customers
- Buying a full version Customer Relationship Management (CRM) Software is expensive
- With SaaS, you can buy a web-based CRM software that is pay as you go and scales to demand!
- **Benefits:** Save money on software license, cut cost on maintenance, and hardware purchase. Combined with lower start-up cost and a faster return on investment!

A Brief History

- In 1960s, IBM and other mainframe providers conducted time-sharing or utility computing services, offering computer power and database to banks and large organization
- In 1990s with the expansion of Internet, Application Service Providers (ASP) appeared. They provided small businesses with the service of hosting and managing specialized business application
- Starting from 2003, the true SaaS became popular due to the increased speed of internet connections. Ultimately, all software will be web-based and pay-as-go

Timeline

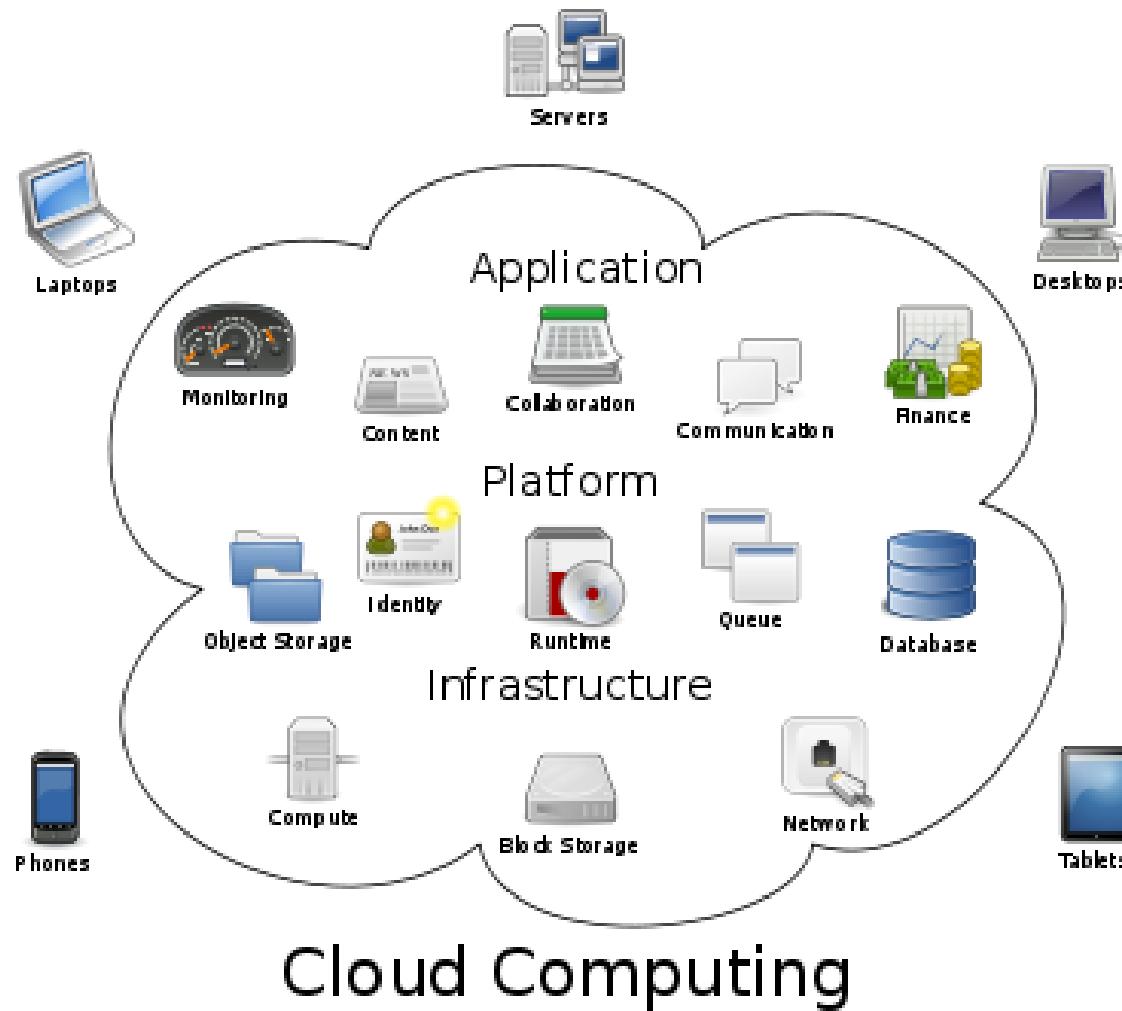


This diagram shows the evolution of the Software as a Service and its ecosystem

Concept

- The idea of using software as a service first popped up in the late 1990s in order to allow sharing end-user licenses in a way that reduced cost and also shifted infrastructure demands from the company to the software provider.
- Does it merely save on the license cost?
- And more: upgrading, maintenance, hardware...

The Big Picture



Software as a Service is located in the application level of the stack

Related Terminology

- **Cloud computing**
 - Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a utility over a network
- **Platform as a Service**
 - Platform as a service (PaaS) is a category of cloud computing services that provide a computing platform and a solution stack as a service.
- **Infrastructure as a Service**
 - Infrastructure as a Service is a provision model in which an organization outsources the equipment used to support operations, including storage, hardware, servers and networking components. The service provider owns the equipment and is responsible for housing, running and maintaining it.
- **Multi-Tenancy**
 - Multi-Tenancy refers to a principle in software architecture where a single instance of the software runs on a server, serving multiple client organizations
- **Application Service Provider**
 - provided businesses with the service of hosting and managing specialized business applications, with the goal of reducing costs through central administration and through the solution provider's specialization in a particular business application

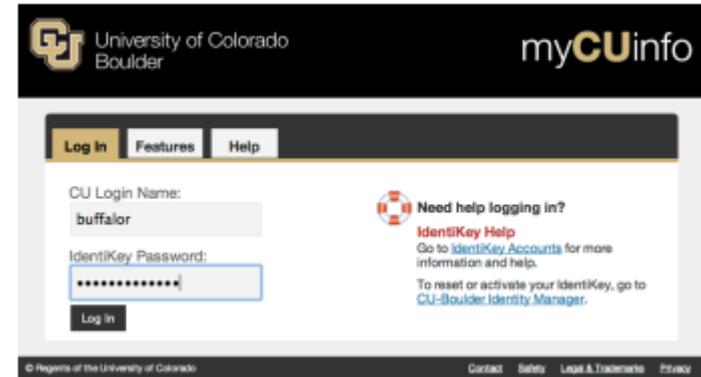
Some key differences

- Clouding computing is the big application context (umbrella) covering SaaS and other related terms
- “... as a Service” are the buzz words used to specify various application scenarios.
 - E.g. Storage as a Service is an umbrella for SaaS applications that provide cloud storage.
- PaaS deals with whole computing platforms provided as a service such as operating system enviromnets
 - E.g. Google Chrome OS running on Google ChromeBooks
- IaaS aims to provide the whole computing power (computer clusters) for the application domain so we don’t have to worry about the physical machines and how they are deployed

Computing Today

- We are in the era of Cloud Computing!
- Cloud computing stack
 - Infrastructure as a Service (IaaS)
 - Platform as a Service (PaaS)
 - Software as a Service (SaaS)
 - Monitoring as a Service (MaaS) - emerging
 - Communication as a Service (CaaS)
 - Anything as a Service – emerging
- So many buzz terms...but SaaS is one of the most widely used service model

SaaS is everywhere



The SaaS market

- SaaS sales in 2010 reached \$10 billion
- In 2011, its sales is up 20.7% from 2010
- SaaS revenue will be more than double its 2010 numbers by 2015 and reach a projected \$21.3
- Business SaaS is the major market – Customer Relationship Management (CRM) is the largest market with 18.8% annually growth worldwide

Business's perspective

SaaS has a lot of appeal to businesses. Here are a few reasons why:

- Multi-tenant software architecture
- Low cost, fast investment, shared license
- High manageability
- Free of deployment and support
- Cost-effective: pay as we go
- Customization is easy
- Can scale well – commercialization

Advantages of SaaS

- Easy to use – Most SaaS applications do not require more than a web browser to run
- Cheap- The pay as you go pricing model of SaaS makes it affordable to small businesses and individuals.
- Scalability: SaaS application can be easily scaled up or down to meet consumer demand. Consumers do not need to worry about additional computing infrastructure to scale up.
- Applications are less prone to data loss since data is being stored in the cloud.
- Compared to traditional applications, SaaS applications are less clunky. They do not require users to install/uninstall binary code on their machines
- Due to the delivery nature of SaaS through the internet, SaaS applications are able to run on a wide variety of devices.
- Allows for better collaboration between teams since the data is stored in a central location.
- Velocity of change in SaaS applications is much faster.
- SaaS favors a Agile development life cycle.
 - Software changes are frequent and on-demand. Most SaaS services are updated about every 2 weeks and users are most time unaware of these changes.

Drawbacks of SaaS

- Robustness:
 - SaaS software may not be as robust (functionality wise) as traditional software applications due to browser limitations. Consider Google Doc & Microsoft Office.
- Privacy
 - Having all of a user's data sit in the cloud raises security & privacy concerns. SaaS providers are usually the target of hack exploits e.g. Google servers have been the target of exploits purportedly from China in the last several years
- Security
 - Attack detection, malicious code detection
- Reliability:
 - In the rare event of a SaaS provider going down, a wide range of dependent clients could be affected. For example, when Amazon EC2 service went down in April 2011, it took down FourSquare, Reddit, Quora and other well known applications that run on it.
 - We shall discuss each of these issues in more details in the next section

Robustness

- SaaS applications may not be able to provide the same level of functionality as traditional applications. This is partly due to current limitations of the web browser. Consider Google doc and Microsoft Office
- Most SaaS applications are intolerant to slow internet connections and this can lead to erratic behavior
 - Google doc may not be synchronized well between teams in a low internet connection

Privacy

- Lots of issues arise with sensitive data stored in the cloud. Common privacy questions include:
 - Who has the access to the data? How to distribute the rights?
 - What type of data can be saved on the cloud, and locally? What about the confidential data?
 - Don't we really have to worry about data sharing? Who is viewing our data, modifying the data, and re-distributing our data? With or without permission?
 - Data sharing between private and public clouds

Security

- SaaS applications are prone to attack because everything is sent over the internet
- Data encryption and decryption
- Communication protocols
- Virtualization versus Multi-tenant architecture: which one is better in terms of the security?
- Transaction processing, networking issues

Reliability

- Although most SaaS applications are highly reliable, down time is still inevitable and can be very expensive – commercial SaaS software
- The application, data, backups, everything are in the cloud, thus making it hard to recover from the server down time.
 - You don't physically own the code, they are in the cloud

Web Services

- A **web service** is a software system designed to support interoperable machine-to-machine interaction over a network.
- The Internet is the worldwide connectivity of hundreds of thousands of computers of various types that belong to multiple networks.
- On the World Wide Web, a web service is a standardized method for propagating messages between client and server applications.
- A **web service is a software module that is intended to carry out a specific set of functions.**
- Web services in cloud computing can be found and invoked over the network.

Web Services

- A **web service is a set of open protocols** and standards that allow data to be exchanged between different applications or systems.
- Web services can be used by software programs written in a variety of programming languages and running on a variety of platforms to exchange data via computer networks such as the Internet in a similar way to inter-process communication on a single computer.
- Any **software, application, or cloud technology that uses standardized web protocols (HTTP or HTTPS) to connect, interoperate, and exchange data messages – commonly XML (Extensible Markup Language) – across the internet is considered a web service.**

Functions of Web Services

- It's possible to access it via the internet or intranet networks.
- XML messaging protocol that is standardized.
- Operating system or programming language independent.
- Using the XML standard, it is self-describing.
- A simple location approach can be used to locate it.

Components of Web Service

- XML and HTTP is the most fundamental web services platform.
- The following components are used by all typical web services:
 - SOAP (Simple Object Access Protocol)
 - UDDI (Universal Description, Discovery, and Integration)
 - WSDL (Web Services Description Language)

SOAP (Simple Object Access Protocol)

- **SOAP stands for “Simple Object Access Protocol.”**
- It is a transport-independent messaging protocol.
- SOAP is built on sending XML data in the form of SOAP Messages.
- A document known as an XML document is attached to each message.
- Only the structure of the XML document, not the content, follows a pattern.
- The best thing about Web services and SOAP is that everything is sent through HTTP, the standard web protocol.

SOAP (Simple Object Access Protocol)

- A root element known as the envelope is required in every SOAP document.
- In an XML document, the root element is the first element.
- The “envelope” is separated into two halves.
- The header comes first, followed by the body.
- The routing data, or information that directs the XML document to which client it should be sent to, is contained in the header.
- The real message will be in the body.

UDDI (Universal Description, Discovery, and Integration)

- UDDI is a standard for specifying, publishing and discovering a service provider's online services.
- It provides a specification that aids in the hosting of data via web services.
- UDDI provides a repository where WSDL files can be hosted so that a client application can discover a WSDL file to learn about the various actions that a web service offers.
- As a result, the client application will have full access to the UDDI, which serves as a database for all WSDL files.

UDDI (Universal Description, Discovery, and Integration)

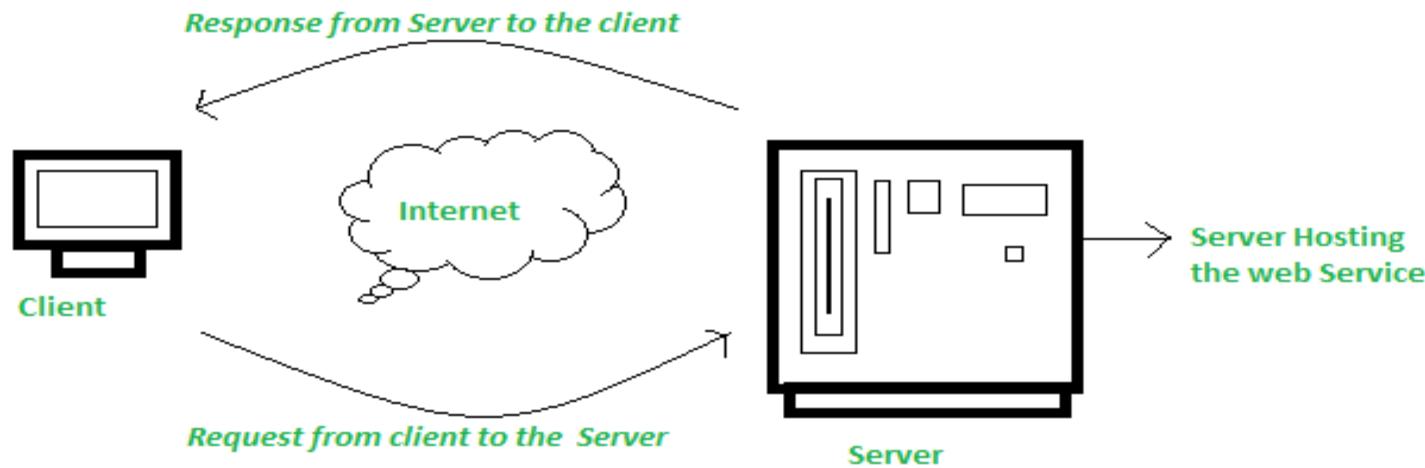
- The UDDI registry will hold the required information for the online service, just like a telephone directory has the name, address, and phone number of a certain individual.
- So that a client application may figure out where it is.

WSDL (Web Services Description Language)

- If a web service can't be found, it can't be used.
- The client invoking the web service should be aware of the location of the web service.
- Second, the client application must understand what the web service does in order to invoke the correct web service.
- The WSDL, or Web services description language, is used to accomplish this.
- The WSDL file is another XML-based file that explains what the web service does to the client application.
- The client application will be able to understand where the web service is located and how to use it by using the WSDL document.

How Does Web Service Work?

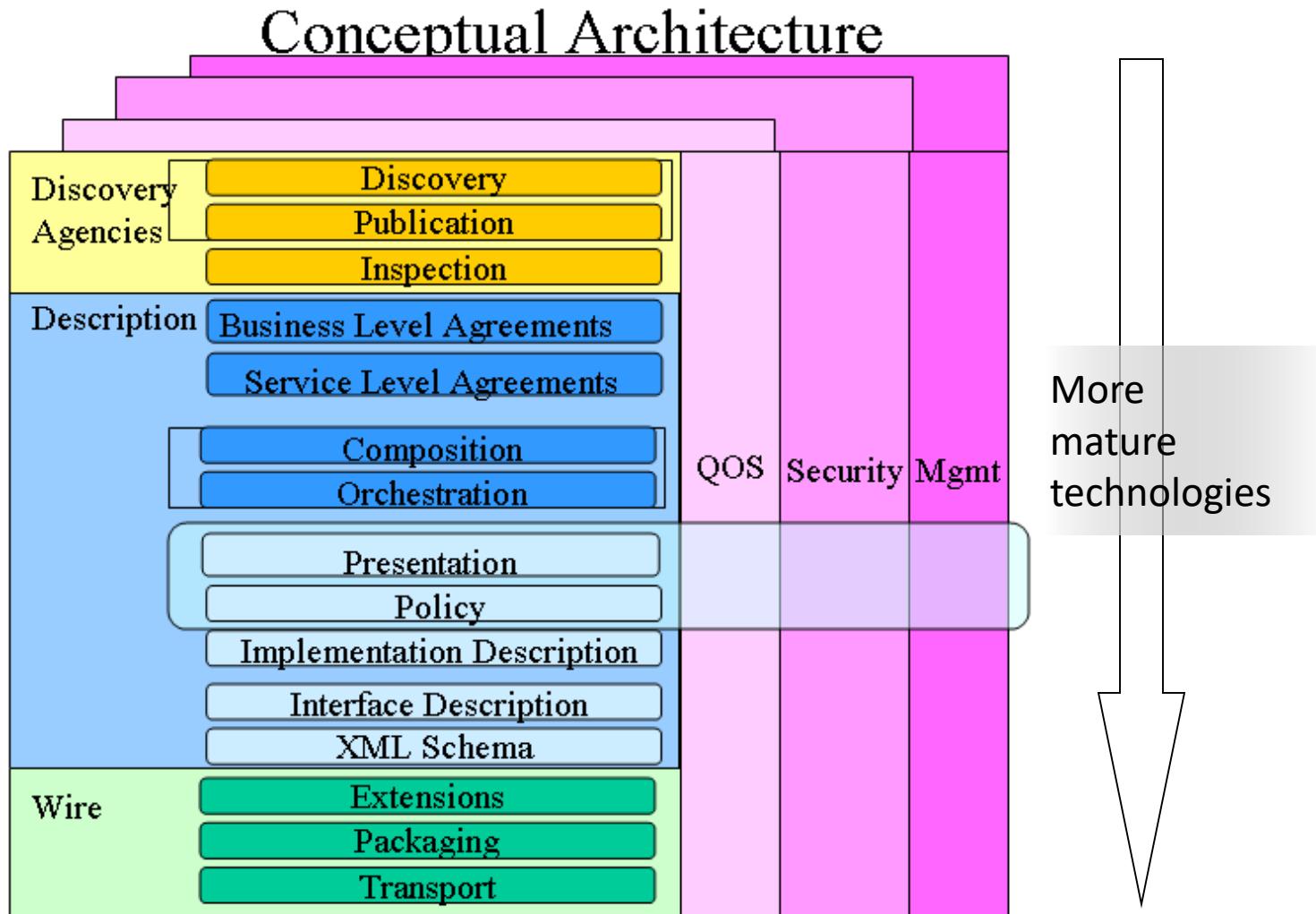
- The diagram depicts a very simplified version of how a web service would function.
- The client would use requests to send a sequence of web service calls to a server that would host the actual web service.



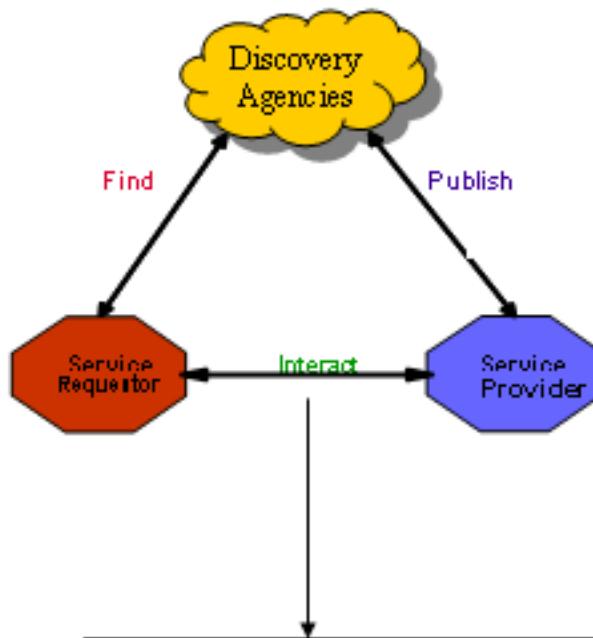
Web Services

- Incorporates additional features and functionality by extending technologies and components defined within the basic architecture, such as:
 - Asynchronous messaging
 - Attachment – typical usage : associating binary data with SOAP messages.
 - Caching
 - Message exchange pattern (MEP) - Describes a generalized pattern of message exchange between two services. e.g. : one-way, request/response, publish/subscribe, and broadcast.
 - Reliable message - implementation of Reliable Messaging one MEP is a series of requests between two nodes with an acknowledgement SOAP Module.
 - Message confidentiality – Can transmit the message via SSL or TLS, or have a SOAP Module provides for encryption and decryption.
 - Message integrity – Can have a SOAP Module use digital signature.
 - Session

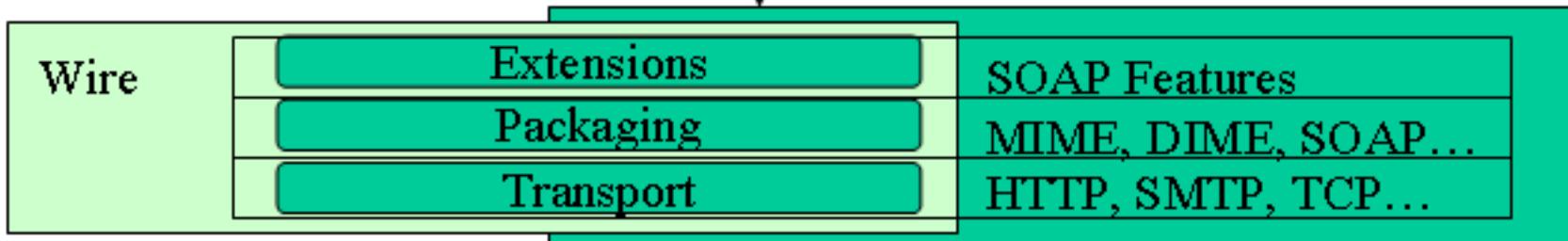
The Complete Web Services "Stack"



The Wire Stack



- Transport: HTTP is the de facto, other may be supported.
- Packaging: SOAP is the de facto standard for XML messaging.
- Extensions :
Additional information attached to web services messages.

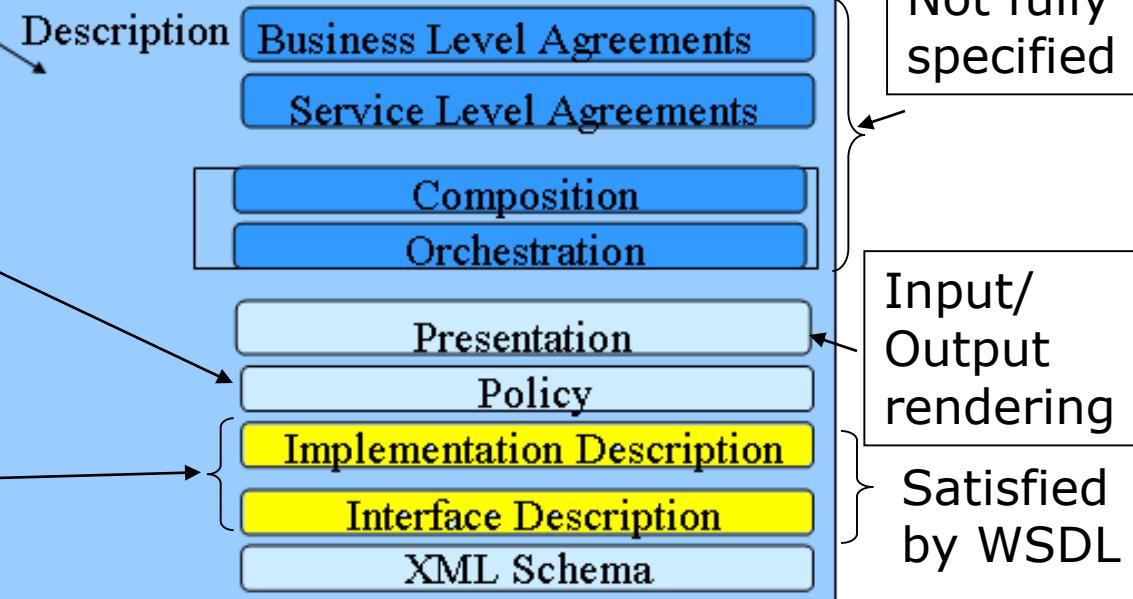


The Description Stack



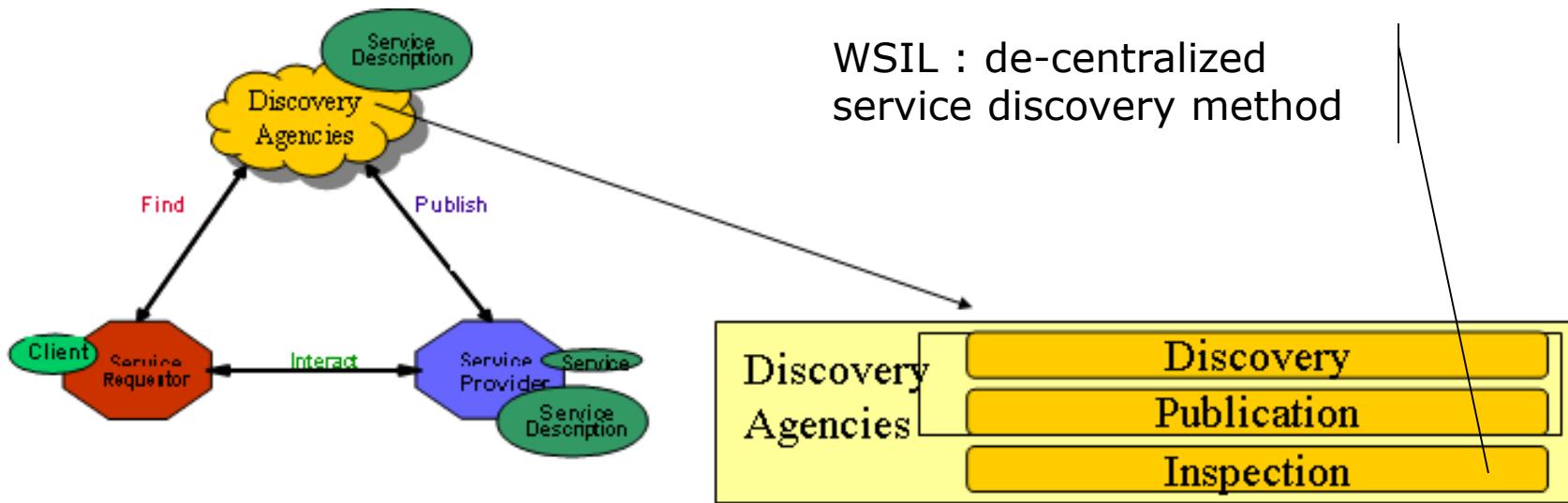
It is actually a stack of description documents defined using XML Schema.

facts, or assertions, and rules that apply to a particular Web service



minimum service description necessary to support interoperable Web services.

The Discovery Stack



- Service can be published using a variety of mechanisms:
 - Direct publish: description sent directly to requestor;
 - WSIL : HTTP GET retrieves descriptions from URL;
 - ***Universal Description, Discovery and Integration*** (UDDI) registries: a Web-based distributed directory.
- Service requestors can retrieve a service description at design time (search by interface) or runtime (by communication and QoS) from a Web page (URL), a service description repository, a simple service registry or a UDDI registry. Discovery depends on how services are published;

The technology so far

The WS technology is completely based on XML.
Therefore, both the data format and the interaction protocols are XML-based:

- customized XML -> data format
- SOAP -> communication protocol
- WSDL -> the Interface definition language
- WSIL/UDDI -> standards for services discovery

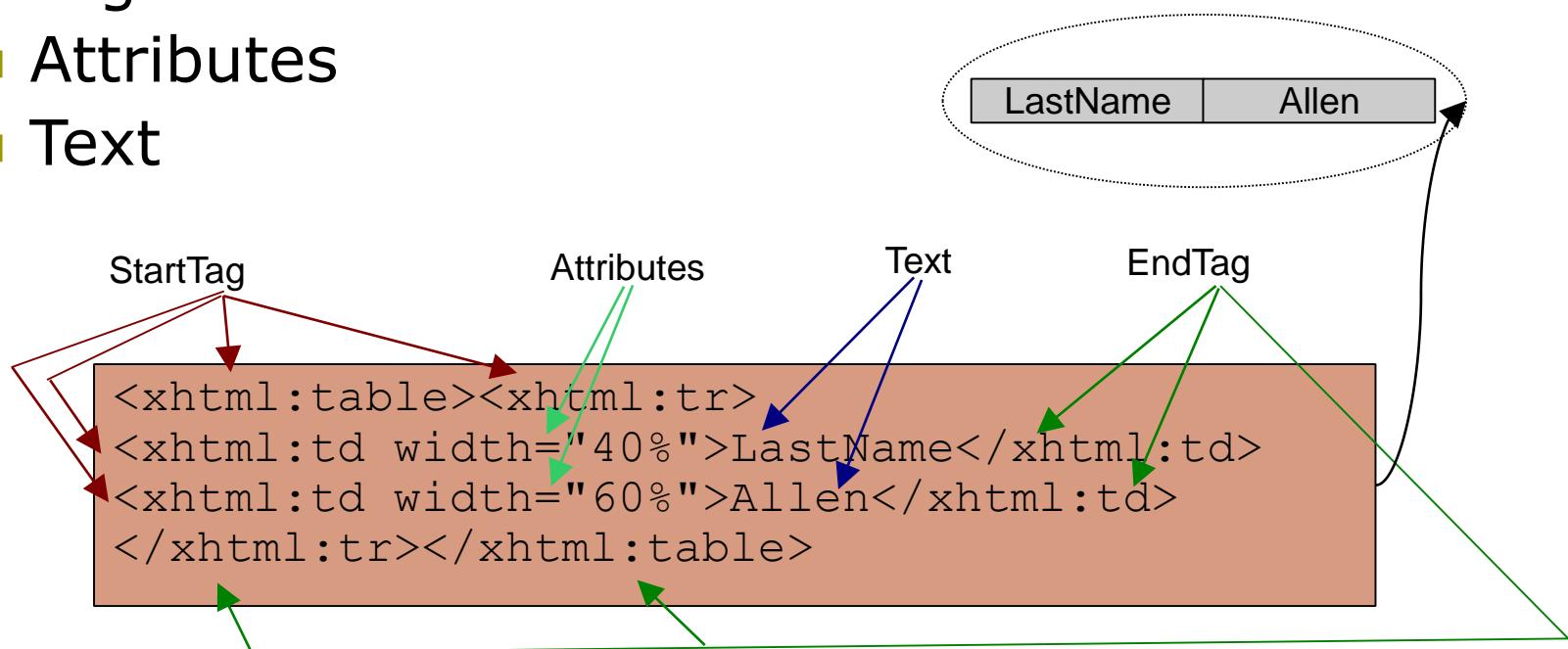
The lowest-level layers (the transport layer) should exploit some existing Internet protocols, like HTTP or SMTP

What is XML

XML is a simple tag-based language for describing information in a structured way.

Basic elements:

- Tags
- Attributes
- Text



How to work on XML

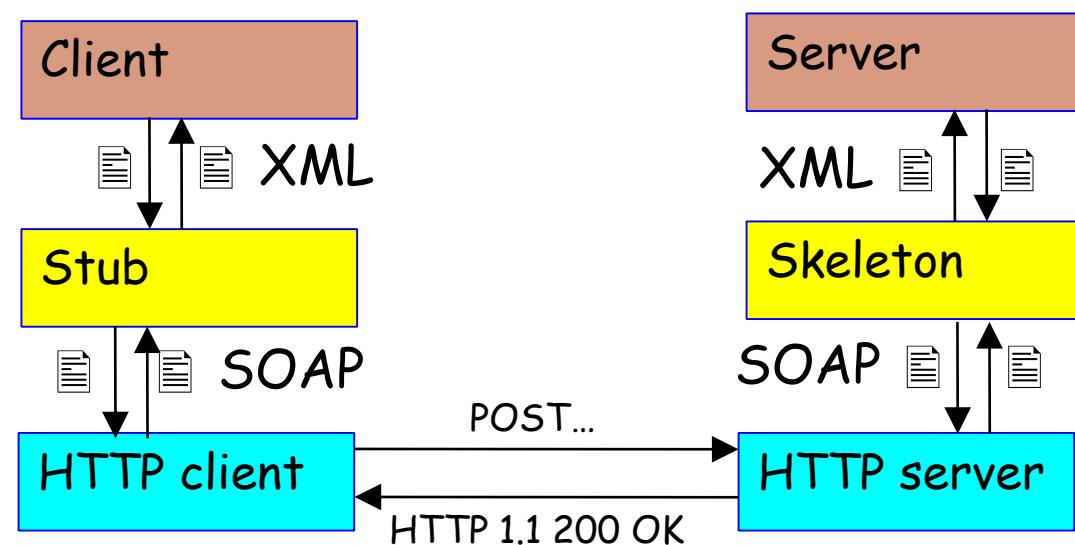
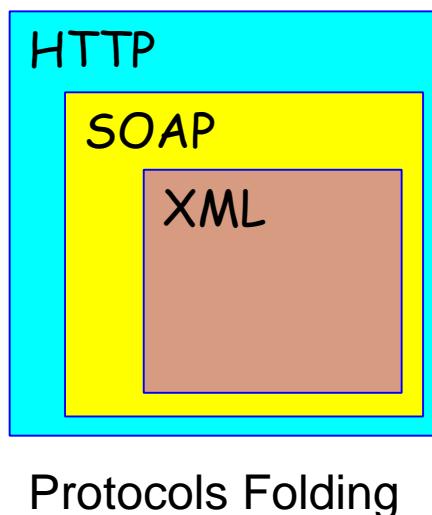
The tree-like structure of XML makes developers life hard.

In practice there is not a standard way for editing and analyzing, but the best method depends on your need. Just choose among:

- SAX -> callback-based parser
- DOM -> tree representation
- XSLT -> “XML to XML” conversion
- XPATH -> queries
- XML Binding -> transparent conversion to objects

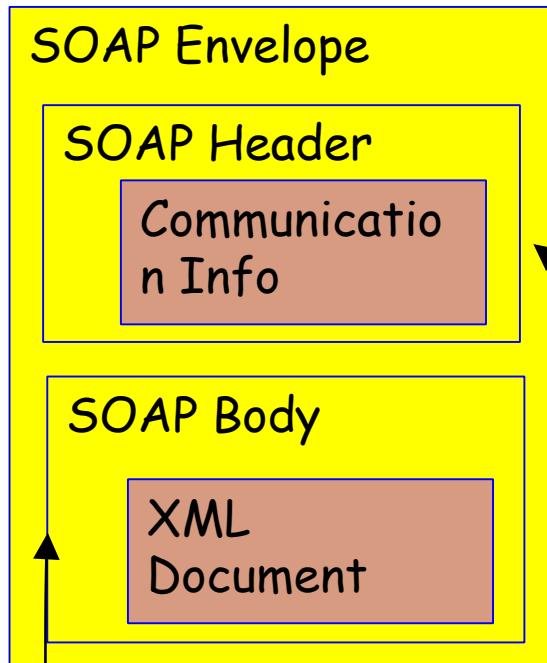
Simple Object Access Protocol

SOAP is a technology to support the exchange of XML-coded messages over a transport protocol, such as HTTP and SMTP. (*wire stack*)



Simple Object Access Protocol

A SOAP runtime engine basically adds a XML envelope to an existing XML document



Document Container

Example

```
<soap:Envelope>
  <soap:Header>
    <axis:SessionKey>
      SDHH37TYEW7R7
    </axis:SessionKey>
  </soap:Header>
  <soap:Body>
    <GetPrice>
      <Item>Apples</Item>
    </GetPrice>
  </soap:Body>
</soap:Envelope>
```

Session, Authentication,
Routing, Security

SOAP Encoding

Dealing directly with XML messages is not easy.
Therefore, SOAP provides a “RPC emulation”
technology

The result: developers work with web services like
with traditional RPC (e.g. CORBA,DCOM,DCE)

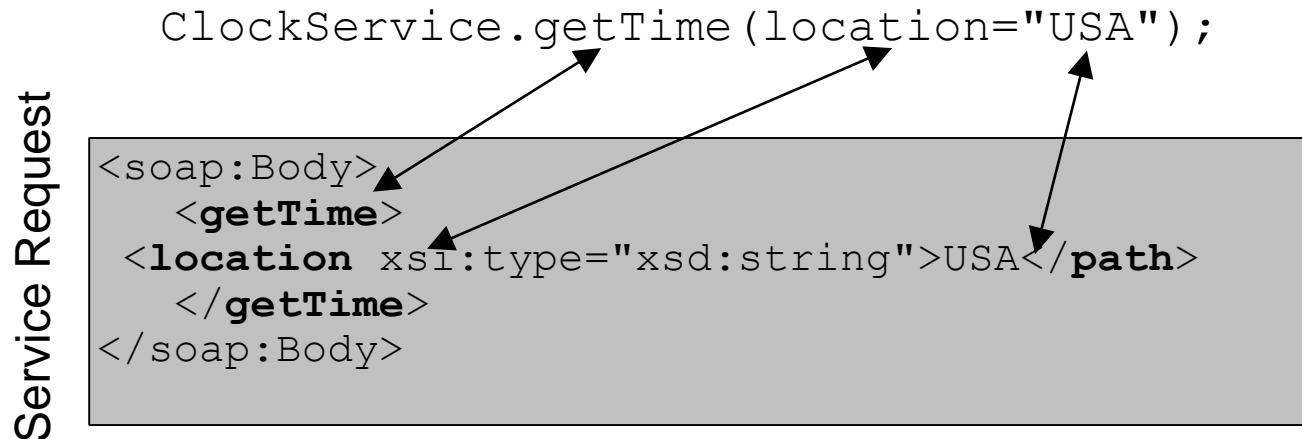
The “RPC emulation” is named SOAP encoding.

Fundamentally it is a set of rules to map a
procedure invocation to a XML document.

SOAP Encoding

The rules:

- method name -> first level element in the SOAP Body
- arguments identifiers -> second level elements
- arguments values -> third level elements
- arguments types -> attribute xsi:type



Web Service Description Language

WSDL is a standard format to describe a Web Service (*description stack*)

A WSDL document is composed by two sections:

- An abstract interface section -> like in traditional IDL, it defines the signatures of procedures (RPC-style) or messages (document-style)
- A deployment section -> it defines the service location and the supported transport protocols

Fundamentally a client uses the WSDL to create the stub or to dynamically decode messages.

Web Service Introspection Language

WSIL and UDDI are the standard way to search Web Services. (*Discovery stack*)

WSIL is the decentralized approach.

Fundamentally a WSIL document contains a directory of the Web Services deployed on a server.

It is analogous to the index.html document for web pages.

In the future, specific crawlers will browse the Internet looking for WSIL documents, like Google does today for web pages.

Web Service Introspection Language

```
<inspection>
  <abstract>Acme Industries Public Web Services</abstract>
  <service>
    <name>Store Finder Service</name>
    <abstract>
      A service to perform a geographical search of Acme stores.
    </abstract>
    <description>
      location="http://example.org/services/storefinder.wsdl"/>
    </service>
    <link location="http://example.org/services/ecommerce.wsil"/>
  </inspection>
```

The diagram illustrates the structure of the WSIL XML code with several annotations:

- An annotation labeled "Service name" points to the element `<name>Store Finder Service</name>`.
- An annotation labeled "Service location and description" points to the element `<description>`, which includes the attribute `location="http://example.org/services/storefinder.wsdl"`.
- An annotation labeled "Link to an other WSL page" points to the element `<link location="http://example.org/services/ecommerce.wsil"/>`.

- **Web 2.0**
- The field of Computer Assisted Learning (CAL) has been transformed to a great extent, since the development of PLATO system in 1960.
- The traditional audio and video materials have been augmented by multimedia applications.
- The developments in the field of Web and digital multimedia have provided new opportunities to teachers to engage the learners as active learners.
- Emergence of Web 2.0 is the most phenomenal of them.
- Traditionally the World Wide Web (WWW) has been considered as a place to retrieve information.

- **Web 2.0**
- The information provided through HTML codes and sharing of resources through FTP (File Transfer Protocol) was the main utility.
- But the drawback was that such flow of information was largely unidirectional.
- This necessitated the need for some interactivity on the web and tools were developed which would allow the users to add content to the web. Such content could be in the form of text, audio, video, slideshows etc.
- The communication became bidirectional.
- The term web 2.0 was first used in 1999 by Darcy DiNucci; but it came into prominence in 2004 when O'Reilly Media hosted the first Web 2.0 conference.
- Web 2.0 is the place where the users (teachers, students and anyone) could read and write.

- **Web 2.0**
- It has taken the educational delivery to the next level of advancement where content can be generated online through collaboration.
- It is an innovative platform where the creative minds meet and discuss or share ideas.

- **Comparison between Web 1.0 and Web 2.0**

Table 17.1: Web 1.0 Vs Web 2.0

Feature	Web 1.0	Web 2.0
Type of Interaction for user	One-way	Two-way
User Participation	Authoritarian	Democratic
Reading and writing capability	Passive	Active
Change of Content	Static	Dynamic
Nature of Web Content	Closed	Collaborative
Reaching audience	Pull Technology	Push Technology

- **Technology and Standards**
- Data is the backbone of Web.
- All the major Internet applications are based on some specialized database (O'Reilly, 2009) like: Google's web crawl, Yahoo!'s directory (and web crawl), Amazon's database of products, eBay's database of products and sellers, MapQuest's map databases, Napster's distributed song database, etc.
- Web 2.0 technologies allow users to store all kind of data.
- With ever increasing volume of data on the Internet, there is a constant need for its access, search, synchronization, movement and managing data from one repository to another and from one network to another network.
- This need has given rise to a new kind of Internet based services like.

- Internet - connectivity and regional Internet caching
- Internet - Filtering
- Application training
- Learning management system (LMS) hosted service delivery
- LMS - development
- Third-party LMS procurement and management
- Online community development and hosting
- Firewall intrusion protection
- Personal workspace, shared folders/library/ search
- Portal controlled filtering
- Unified communications (UC): email, Web mail and filtering
- UC: videoconferencing - desktop and meeting room; white boarding; application sharing
- UC: collaboration - instant messaging
- Web site hosting
- Content : hosting - external and internal
- Content: delivery - Webcasting and broadcasting
- Supporting content development

Some of the standards and technologies that are used in the Web 2.0 are as follows:

- **OpenID:** If you have ever tried to write a comment on a blog, you may have noticed that before allowing your comment to appear on the blog, it asks you to log in. There you are provided some choices on how you wish to log in to that site. You may be asked to use your email account of some service provider like Google or Yahoo, or it asks you to use your OpenID. OpenID is a form of single digital identity across the web and is already used by providers like Google, Microsoft, Yahoo, AOL, and MySpace etc. With the help of this digital identity you can easily log on to a site and interact or contribute.

- **RSS:** Let's go back to Unit 16, where you studied about RSS in details. Yes, you remembered it correctly; RSS is an acronym for "Really Simple Syndication". Maybe while surfing websites you have already noticed the orange RSS icon. It was developed by Netscape in 1999 and is used on websites to allow the publication of recurrently updated items from an external source. Feeds allow you to have new content delivered to a computer or mobile device as soon as it is published. Once you set the preference for an item for RSS feed, the information will come to you, instead of you searching the internet for that information. We discussed about RSS in detail in the previous unit.
- **OAuth:** With the help of OAuth you can publish and interact with protected data on the Internet. This may happen when an Internet based application you are using needs more information from you like your date of birth or from a person who has logged on to a site via his OpenID asking for email address. In that case if the required information is stored in a different location, retrieving that information may be difficult and time consuming by the user. In that case, OAuth asks the user about the location of that information and the related password and then retrieves that information. You must have experienced this while using social networking applications. OAuth enables the application to access data for one service to be accessed located within a subset of another service over a secure data protocol.

- **Microformat:** These are a set of simple, open data formats used to represent data in XHTML. Microformats embed these XHTML data directly into the web pages which is then accessed or viewed by the user with the help of an internet based application or service (for example through some search engine or browser plug-in). Microformats enable adding context to information so that other services can use that on automated basis.
- **Ajax** (shorthand for Asynchronous JavaScript and XML) is a group of interrelated web development techniques used on the client-side to create interactive web applications. With Ajax, web applications can retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page. The use of Ajax techniques has led to an increase in interactive or dynamic interfaces on web pages as in Web 2.0. Some other technologies that make Ajax are HTML, XML, XHTML, CCS, XSLT, XMLHttpRequest, Document Object Model and Javascript.

- **Javascript** is an object oriented scripting language that enables enhanced interactivity by providing a dynamic website. The JavaScript is used to write functions that are embedded in or included from HTML pages to interact with the Document Object Model (DOM) of the page. Some simple examples of this usage are:
 - Pop up windows in a page.
 - Validation of input values in web forms to make sure that they will be accepted before they are submitted to the server.
 - Changing images as the mouse cursor moves over them.

JavaScript code can run locally in a user's browser (rather than on a remote server), and it can respond to user actions quickly, making an application feel more responsive. Furthermore, JavaScript code can detect user actions which HTML alone cannot, such as individual keystrokes. Applications such as Gmail take advantage of this: much of the user-interface logic is written in JavaScript, and JavaScript dispatches requests for information (such as the content of an e-mail message) to the server.

- **XML** (Extensible Markup Language) is an application profile of Standard Generalized Markup Language that believes in the use of markup should describe a document's structure and other attributes, rather than specify the processing to be performed on it. It also postulates that markup should be rigorous so that programs and data bases can be used for processing documents as well.

17.2.3 Big Ideas of Web 2.0

JISC Technology and Standards Watch (TechWatch) report highlights Web 2.0 services and applications like blogs, wikis, social bookmarking and networking, RSS and video sharing. It also analyses some of the recent developments related to the field of web technologies, like social networking, aggregation, filtering and tracking content. This report indicates that there are at least six big ideas of Web 2.0 which shall change the way the people interact. These big ideas are:

Individual production and user-generated content

The User Generated Content is also known by the name of Self-publishing or Personal Publishing (Downes, 2004) and 'self expression'. The web has enabled users to contribute materials for programmes, news, documentaries or all other kind of programming wherein the users can input their content generated through their cameras, mobile phones or other devices. You must have noticed that now-a-days the news channels in India have started 'citizen journalism' asking citizen to send their news or photos or stories to be broadcast.

Harnessing the power of the crowd

This is also known as 'Wisdom of Crowds (WoC)'(which incidentally is the title of a book written by James Surowiecki) wherein he identified three types of problems (cognition, coordination and cooperation) and how they can be solved through group efforts. O'Reilly (2005) explained the concept of WoC as "harnessing the collective intelligence". One such example is Cloudmark (a collaborative spam filtering system) which reflects 'the individual decisions of email users about what is and is not spam, outperforming systems that rely on analysis of the messages themselves' (p. 2). The Cloudmark system is a representative of distributed human intelligence where in simple terms it can be said that here media content generation is outsourced to crowd logged into the Internet. Like multimedia sharing sites (Flickr and YouTube) have created a second generation of websites where user generated content can be re-used. *ShutterStock* and *Fotolia* are such other web-based stock photo or video databases where the amateur content producers and anyone who wants to use the content can collaborate.

Data on an epic scale

Since Web 2.0 allows users to generate content, therefore the volume of data on the Internet is ever-increasing. Database management and networking have developed as the core competency for the people involved to deal in with this data on an epic scale. Google, Amazon, Yahoo or Ebay like companies have voluminous data which is increasing day by day as new content is being generated and added to the pool.

Architecture of participation

The architecture of participation indicates the service where user content generation and collaborations are equally important and put into effect. Take the case of Bit Torrent where users download the data but make that data available for others to download too. Therefore both the data and bandwidth are made available as more and more people are involved in the network. This model is based on the premise of ethics of cooperation, you downloaded the data so let others also download it from your source.

Network effects

Klemperer (2006) and Liebowitz and Margolis (1994) described network effect as an increase in value to the existing users of a service when more and more people start using it, there is more interaction among themselves. Understand it with an example of Microsoft Office products. It is one of the most successful software products because more and more people started using it (because someone else was using it) and thus it became easy for people to share documents with each other. Therefore not only the user was benefited but also the software company too. There are other services too, take for example the case of Facebook. Once you add someone to your friends list, others are added from their friends list and thus network grows.

Openness

Openness deals with access, control and rights of digital content in terms of legal, political, and cultural settings. Web 2.0 services are based on open standards, using open source software, allowing use of free data, re-using data etc. Take the case of Open source browser FireFox and its extensible plug-ins where users are allowed to experiment to bring improvement to the software.

17.2.4 Key Web 2.0 Services

Web 2.0 offers many services pertaining to different areas like social networking, collaboration, content sharing in terms of photo sharing or document sharing or video sharing etc. Some of the key Web 2.0 services are:

- Blogs
- Wikis
- RSS and syndication
- Tagging and social bookmarking
- Multimedia sharing
- Audio blogging and podcasting
- Newer Web 2.0 services and applications

You will be reading about blogs and wiki in the next sections, whereas about RSS we have studied in Unit 16. Tagging is adding a keyword to a digital object (any picture or video clip or a website) to describe it. Social Bookmarking allows sharing common features and allows users to create a list of bookmarks or favourites at a central server so that these can be shared with other users. See for example, <http://delicious.com/>

Multimedia sharing allows users to store and share multimedia content like you must have seen or used YouTube (video), Flickr (photographs) and Odeo (podcasts). Social Networking, Aggregation services, Data ‘mash-ups’, and Tracking and filtering content are some of the examples of newer Web 2.0 services and applications.

17.3 BLOGS

A blog is a personal website that contains content organized like a journal or a diary. Each entry is dated, and the entries are displayed on the web page in reverse chronological order, so that the most recent entry is posted at the top. Readers catch up with blogs by starting at the top and reading down until they encounter material they're already read.

Though blogs are typically thought of as personal journals, there is no limit to what may be covered in a blog. It is common for people to write blogs to describe their work, their hobbies, their pets, social and political issues, or news and current events. And while blogs are typically the work of one individual, blogs combining contributions of several people, 'group blogs', are also popular.

Blogs are connected to each other to form what is commonly known as the 'blogosphere'. The most common form of connection is for blogs to link to each other. Blog authors may also post a list of blogs they frequently read; this list is known as a 'blogroll'. Blogs may also be read through special readers, known as 'RSS readers', which aggregate blog summaries produced by blog software. Readers use RSS readers to 'subscribe' to a blog. Popular web-based RSS readers include Google Reader and Bloglines. You have studied about RSS in the previous unit.

17.3.1 Using Blogs in Education

Blogs are widely popular in education, as evidenced by the 400 thousand educational blogs hosted by edublogs (www.edublogs.org). Teachers have been using them to support teaching and learning since 2005 (Downes, 2004). Through years of practice, a common understanding has formed around the benefits of the use of blogs in education (see <http://anne.teachesme.com/2007/01/17/rationale-for-educational-blogging/>).

Because blogs are connected, they can foster the development of a *learning community*. Authors can share opinions with each other and support each other with commentary and answers to questions. For example, the University of Calgary uses blogs to create learning communities.

Additionally, blogs give students ownership over their own learning and an *authentic voice*, allowing them to articulate their needs and inform their own learning. (Uniservity, 2007) Blogs have been shown to contribute to identity-formation in students (Bortree, 2005).

Further, blogging gives students a genuine and potentially *worldwide audience* for their work (Aguilar, 2009). Having such an audience can result in feedback and greatly increase student motivation to do their best work (See <http://www.big6.com/2006/06/12/motivating-middle-schoolers-grades-5-8/>). Students also have each other as their potential audience, enabling each of them to take on a leadership role at different times through the course of their learning.

17.3.2 How to Use Blogging in Learning?

- *Begin simply.* Most uses of blogs in the classroom began with the instructor using blogs to post class information such as lists of readings and assignment deadlines (Downes, 2004). This fosters in the teacher a familiarity with the technology and with students a habit of regularly checking the online resource.
- *Lead by example.* Before requiring students to blog, instructors should lead by example, creating their own blogs and adding links to interesting resources and commentary on class topics. This not only produces a useful source of supplemental information for students, it creates a pattern and sets expectations for when students begin their own blogging.
- *Read.* Students should begin their entry into blogging by reading other blogs. Teachers should use this practice not only to demonstrate how other people use blogs to support learning but also to foster critical thinking and reading skills. Teaching how to *respond* to blog posts is as important as creating blog posts.

- *Create a context.* Like the author facing a blank sheet of paper, a blogger will be perplexed unless given something specific to write about. Have students blog about a current issue, about a specific piece of writing, or some question that comes up in the course.
- *Encourage interaction.* Blogging should not be a solo activity. Encourage bloggers to read each other's works and to comment on them. Encouraging students to set up an RSS reader with each other's blogs will make reading and commenting a lot easier. Teachers, also, should subscribe to student blogs and offer comments, again setting an example of the expected practice.
- *Respect ownership.* A student blog becomes important because it is a manifestation of his or her own work. However, to have this value, a student's ownership of a blog must be genuine. While reasonable limits or codes of practice (See http://www.digitalquery.com/2005/08/hill_knowlton_o.html) need to be respected, student bloggers should have the widest latitude possible for personal expression and opinion.
- *Address issues immediately.* The most significant danger to students online is posed by other students. In particular, bullying (or ragging) is a significant problem (See <http://en.wordpress.com/tag/school-bullying/>). It is important to spot instances of bullying as soon as they occur and to take steps to prevent further incidents. Teachers should educate themselves as online bullying can be invisible and hard to address.

17.4 WIKIS

A wiki is a website which can be edited by anyone having an account on the wiki platform. Wiki is a great tool for collaboration over the Internet and a store house of information. Allowing anyone to add, delete or edit the content on the wiki pages has made it an effective tool for collaborative writing.

The term wiki has been taken from Hawaii Language, where they call it *wiki wiki* (means quick or fast). In simple terms, wiki can be taken as simplified webpages where all the previous versions of a page are also stored. This enables one to retrieve any past page. There are different tools inbuilt in a wiki system to keep track of changing information on wiki pages or uploading images, audio or video or providing links (URL) to internal pages or external websites (external links).

History of Wiki

WikiWikiWeb was the first Wiki software which was developed by Ward Cunningham in 1994. He described it as “the simplest online database that could possibly work.” [http://en.wikipedia.org/wiki/Wiki#cite_note-2#cite_note-2]. The WikiWikiWeb was installed on the Internet domain C2.com on 25 March 1995. Cunningham is said to be inspired by Apple’s HyperCard system (this system allowed users to create ‘card stacks’ by creating links among various cards). The Wikis gained popularity as collaborative software and were being used for project communication, intranets and documentation where one user can comment on and edit the text of other user. Wikis are dynamic databases for creating, sharing, updating, using and searching information on the web. It is an open space or platform to engage in sharing and learning.

Strengths

- Free, openly available to anyone (you need an internet connection to access pages).
- You can write on the topic of your interest where others can contribute to your content.
- Since others can contribute to your content, it encourages peer review of content and quality of content may improve.
- The wiki pages can be edited by any user (who is authorised to do so, in other words, who is a registered user).
- The history of all the pages created is saved and any time you can revert back to a page.
- The 'Watch' feature enables to be informed of any change of content on that page.
- It provides a collaborative platform for developing and sharing content, different people can work from different parts on same document.
- You can include online quizzes and assessment activities in your course modules.
- Other software utilities and applications can be easily integrated into wiki pages, like YouTube videos, Slideshare presentations, Google Calendar, MindMap etc.
- The wiki editing skills are easy to learn and use.
- As soon as you edit and save a page, it is published on the web, so it is instantaneous in nature.
- Licensing costs can be taken care of because there is a wide range of open source software that you can install for institutional wiki.

Limitations

- There can be incomplete information or page on a wiki platform.
- Since anyone can edit the pages, there are chances that incorrect information can be uploaded on to the pages. (But since others can read that and correct it, so this aspect can be taken care of). Also at systems level editing can be blocked if required.
- The educational institutions are yet to recognise it as a full scale mode of instruction delivery as there are questions about the validity and reliability of content.
- Since there is no formal structure of wiki, therefore the information can be disorganised if page designing is not done carefully.

17.4.2 Software for Wiki

There are many software available for creating your own wiki. The Wikimedia Foundation's MediaWiki is one of the most widely used open source wiki technology that is robust enough to also host an encyclopaedia (e.g. Wikipedia and its different versions such as wikibooks, wikiversity, etc.). you may like to host a media wiki server for which you need the following:



Figure 17.1: Webapage of Media Wiki

17.5 SOCIAL NETWORKING

Social networking is a term in common use only since 2003. The term has been defined by many and generally viewed as referring to networked tools that allow people to meet, interact and share ideas, artifacts and interests with each other. Social networking applications have been phenomenally popular with sites such as Facebook, MySpace, SecondLife and LinkedIn counting their user numbers in the tens of millions. Social networking to date has found applications primarily in the contexts of informal learning and entertainment however there is growing interest in its use in formal education in face-to-face, distance and blended modes. Social networking in the context of distance education can be defined as “networked tools that support and encourage learning through face-to-face and online interactions while retaining individual control over time, space, presence, activity and identity” (Anderson, 2006). Key to understanding both the power and the disruptive affordances of social networking is what Dalsgaard (2008) refers to as transparency – making visible and retrievable the activities, ideas, communications, artifacts and interests of others.

There are many different network learning applications. Some are generalized and multi-faceted application systems that combine social networking applications including blogs, wikis, profiles, resource tagging, documents sharing and other services. Conversely, there are specialized social networking applications focusing on particular applications such as language learning, meeting people who live near by or those who share common interest, hobbies or goals, scheduling and many other applications. The web 2.0 aggregation site <http://gotoweb20.net> currently lists over 2800 applications - most of which could be classified as social networking applications.

For e-learning applications social networking serves three broad functions (Anderson, 2009):

- *Socializing:* Many forms of distance education and their e-learning derivatives have focused on the provision of content to students and provided only limited contact between student and teacher and often no opportunity for student-student interaction. This lack of social interaction, help seeking and provision, and lack of general interpersonal communication and support opportunities has been associated with lack of social integration and resulting higher levels of attrition in both distance education and e-learning (Kember, 1995; Rovai, 2003; Tinto, 1987; Woodley, 2004). Of particular concern in modern e-learning is the inability of institutions to provide contact information to fellow students owing to restrictions on release of private student information to other students. Thus, it can easily happen that students enrolled in the same course, living in the same apartment building, have no opportunity to connect with each other for mutual support, engaging in 'study buddy' or study group type interaction, engage in cooperative

- *Sharing:* One of the most common informal and formal learning applications of network software is the capacity to store, organize and annotate network resources. These include favorite web sites, photographs, music, travel recommendations, references, books and many other electronic resources that people want to be able to quickly retrieve, annotate and share with others. If these resources are stored in accessible networked locations and tagged or identified by the user, they can be combined with other people's resources to create aggregated collections. These collections allow users to discover what others have found, to rate and comment on these resources and generally add value to the individual collection by collective aggregation (Dron & Anderson, 2007). These shareable resources need not be restricted to those created by others. Rather resources created by students and teachers such as learning diaries (blogs), student created learning resources (portals, wiki contributions, original music, multi-media art, reports and essays) can also be shared. These collections need not be bound to particular courses, cohorts or even institutions. Rather they can be used to create permanent, yet continuously growing and evergreen resources as they are used and augmented by multiple groups of learners and educators.

- **Sojourning:** To sojourn means to travel or work with others. There is ample evidence from both class room delivery and distance education at all levels of formal education that collaborative and cooperative learning increases learning effectiveness, motivation, persistence and develops interpersonal and communications skill collaborative (Fisher, Phelps & Ellis, 2000; Gokhale, 1995; Johnson & Johnson, 1994; Kaplan-Leiserson, 2003; Kaye, 1991; Kreijns, Kirschner & Jochems, 2002; Shindler, 2004; Springer, Stanne & Donovan, 1999; Stacey, 1999). However, providing collaborative learning opportunities for distance education students has, until the development of networking software, always been inconvenient, restrictive and often expensive (Paulsen, 2008). Social software allows groups of students to efficiently schedule their activities, meet online via text chat, audio, video or immersion technologies and to engage collaboratively in a variety of brainstorming, mind mapping, group games, simulations, project management, and other types of organizational, administrative and learning activities.

- **What is webOS?**
- WebOS is an LG-owned, Linux-based operating system for smart devices, particularly smart TVs.
- The OS runs on LG televisions, as well as on a limited number of other LG devices, including refrigerators, projectors and digital signage.
- In addition, several third-party vendors now license webOS from LG for their smart TVs, including RCA, Ayonz and Konka.
- The OS is also making headway into the auto industry, where the system is being used for car infotainment systems.

- **What are webOS features?**
- WebOS provides a graphical user interface (GUI) that enables TV viewers to access and control the television's features, much like an OS such as Android or iOS lets users work with their smartphones.
- The webOS GUI includes the Home Dashboard, where users can navigate apps, configure settings and connect with compatible devices.
- WebOS can also connect directly to the internet, making it possible to stream content and access other cloud resources.
- In fact, the OS comes with its own browser for accessing web resources directly.

- The webOS platform also includes several other features, although their availability depends on the type of device and model.
- For example, some of LG's smart TVs are compatible with Apple HomeKit, Amazon Alexa and Google Home.
- They might also include intelligent voice recognition, offer hands-free voice control, or even provide sports alerts.
- In addition, many of the TVs are integrated with LG's Magic Remote, which brings point-and-click control functionality to LG smart TVs.

- **Web apps supported by webOS features**
- When used for LG televisions, the webOS platform comes with a set of preinstalled apps such as Apple TV, Disney+, Hulu, Prime Video and Netflix.
- And users can download additional apps from the Content Store (LG's app store) if they're available.
- The platform supports two types of web apps:
- **Packaged web apps.** Also referred to as basic web apps, this type of app is created as a package that includes the resources necessary for the app to run on a smart TV or other device.
- When the user installs the app on the device, the app's resources are also installed. If there are any changes to the app, a new package must be downloaded and installed. This can be a challenge if the app is updated too frequently.

- **Hosted web apps.**
- With this type of app, the contents are hosted on a remote server.
- The installed app is basically a shell that points to the server where the app's contents reside.
- The resources are then downloaded from the server to the device when the app is launched.
- Updates are easier to handle than with a packaged app because the updates don't have to be pushed out to all the devices with each change.
- However, the hosted app might not perform as well as the packaged app because it's subject to the whims of network connectivity.

- The webOS core contains a set of services for running apps, managing devices, connecting to networks and carrying out other operations.
- The OS also includes the system bus, a channel that enables apps to communicate with the webOS services.
- In addition, webOS supports the Connect SDK, an open source framework that enables mobile apps to connect to multiple device platforms, providing developers with more flexibility when building their mobile apps.
- **WebOS was first developed by Palm** as a mobile operating system.
- It was released in 2009 as Palm webOS.
- Hewlett Packard acquired Palm in April 2010 and used the OS in a number of Palm and HP smartphones. HP also modified the OS for its tablet PCs, including TouchPad devices. When these devices failed to gain market share, HP made webOS open source. LG purchased webOS in February 2013 and modified it for its smart TVs.

- In 2018, LG introduced an open source edition of webOS, which is officially named WebOS Open Source Edition (OSE).
- The project is hosted on GitHub and is available under Apache License, Version 2.0.
- WebOS OSE includes the core architecture of webOS, but adds other features for supporting more diverse vertical markets, including robotics, automotive and smart homes.
- WebOS OSE is being actively developed by a community of partners, programmers and other contributors.