UNIT-III
Cloud Computing Technology

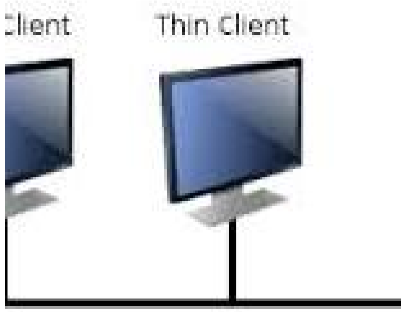
Reference book:

Toby Velte, Anthony Velte, Robert Elsenpeter, ”Cloud Computing, A practical Approach”, Wiley India, 2011, ISBN: 0071626948

Hardware and Infrastructure

Clients:

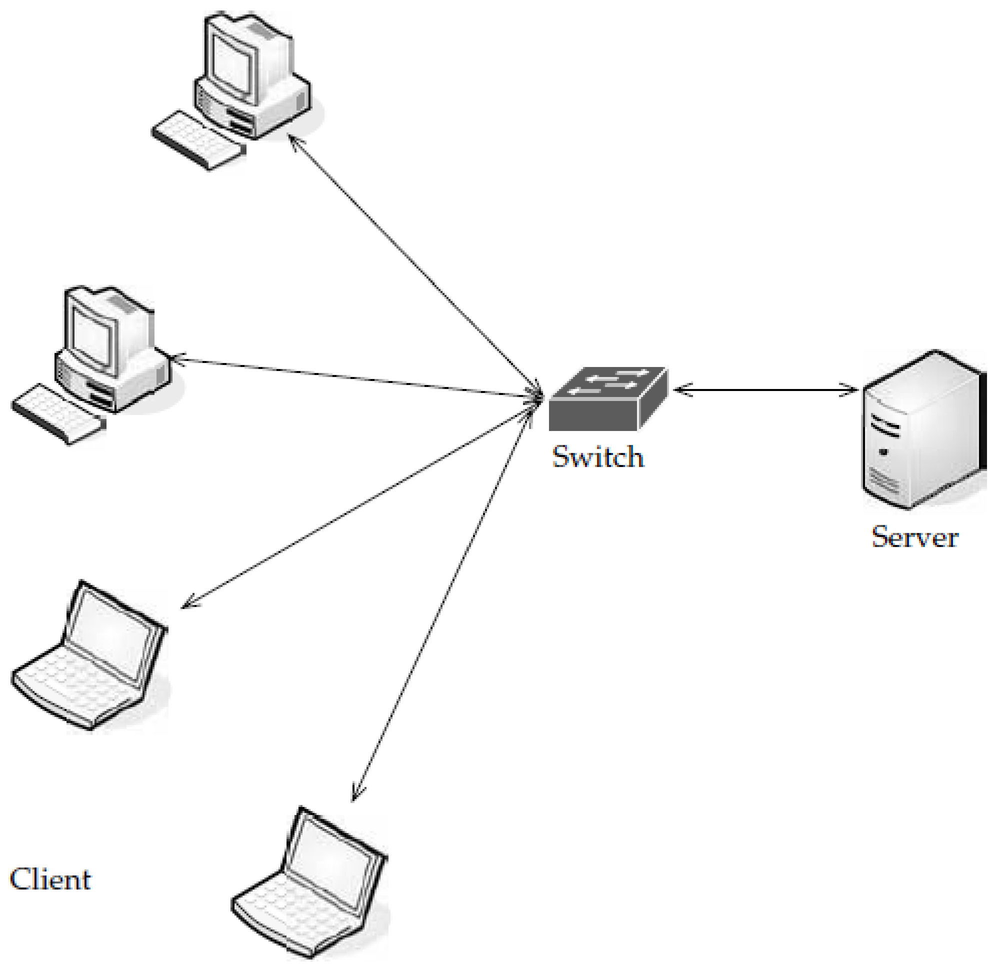
- End users configured to communicate with cloud.
- 3rd party involved in storing data.



Security

Data Leakage:

- The biggest benefit is the centralization of data.
- Organizations have an issue with asset protection, in no small part because of data being stored in numerous places, like laptops and the desktop.
- Thick clients are apt to download files and maintain them on the hard drive, and there are plenty of laptops out there with nonencrypted files.
- Using thin clients creates a better chance for centralized data storage. As such, there's less chance for data leakage.
- Centralization also provides the opportunity for better monitoring.
- That data is in one place makes it easier to check in on your data and see that everything is okay



Data store on local server with clients that store data has more opportunity for data leakage than clients that maintain no permanent storage.

Offloading Work:

- Another security benefit isn’t so much a technology, but the cloud provider to provide adequate security. After all, can your organization afford 24/7 IT security staffing? The fact of the matter is that your cloud provider might offer more security features than you had before.
- The fact that so many clients are paying allows cloud providers to have beefier security, simply because of the economy of scale involved.
- That is, there are many paying clients so the provider is able to do more, because there is more money in the pot.
- Plus it’s to the provider’s benefit to offer more, because they want to get a good reputation.

Logging:

- Logging is also improved. It’s something that, in-house, usually gets the short end of the stick. But in the virtualized world of cloud computing, providers can add as much memory as they need to extend logging.

Forensics:

- If there is a breach, the cloud provider can respond to the incident with less downtime than if you had to investigate the breach locally. It is easy to build a forensic server online, and it costs almost nothing until it comes into use. If there is a problem, the virtual machine can be cloned for easy offline analysis. Further, many companies don’t have a dedicated in-house incident response team. If there is a problem, IT staff have to quickly figure out their new job of taking the server down, quickly investigating, and getting it back online for minimal production downtime.

Development:

- Security vendors also have a unique opportunity in the cloud. Since it’s new ground, there are new opportunities for the vendors who are open-minded enough to imagine them.

Auditing:

- As an IT professional, you already know the headache of securing your own local network. But when you send your data to the cloud, a whole new set of issues arise. This is largely because your data is being stored on someone else’s equipment.

Compliance:

- The same security issues that your organization deals with are the sorts of issues that SaaS providers face—securing the network, hardware issues, applications, and data.
- But compliance adds another level of headache. Regulations like Sarbanes-Oxley (SOX), Gramm-Leach-Bliley (GLBA), and HIPAA, and industry standards like the Payment Card Industry Data Security Standard (PCI DSS) make things particularly challenging.
- Prior to SaaS, compliance could be managed by a few tasks:
    - Identify users and access privileges
    - Identify sensitive data
    - Identify where it's located
    - Identify how it is encrypted
    - Document this for auditors and regulators
- SaaS makes these steps even more complicated. If you store compliance-sensitive data with an SaaS provider, it is difficult to know where the data is being stored.
- It could be on the provider's equipment, or it could even be on the equipment of one of the provider's partners.
- SaaS brings with it a number of regulations, including PCI DSS. Within PCI DSS are regulations for service providers.
- Requirement 12.8 of PCI mandates that service providers be compliant and contractually acknowledge their responsibility for protecting credit card data.
- The PCI DSS Appendix A goes into even more depth laying out rules and regulations.

Network

- In order for the cloud to deliver its best resources, there are differing levels of connectivity needed.
- Gartner notes that different organizations require different things from the cloud, and as such they will have to connect in different ways.

1. Basic Public Internet:

- This is the type of access that you buy from an Internet service provider (ISP) and connect with via broadband or dial-up, based on your location.

This model has the following advantages:

• There's a large audience. Anyone with Internet access can use this solution.

• It's highly fault tolerant.

• Many provider options are available.

• Secure Sockets Layer (SSL)–based, Hypertext Transport Protocol Over Secure Sockets Layer (HTTPS), encrypted access provides confidentiality.

• It's cost-effective.

It also has the following disadvantages:

• Lack of end-to-end quality of service (QoS), thus making end-to-end service-level agreements (SLAs) difficult to reach.

• Probability of poor response over high-latency connections. This is worsened by protocol inefficiencies in TCP, HTTP, and web services.

• Downtime that might be out of your control (cable cuts, problems at the ISP, and so forth).

2. The Accelerated Internet

- SSL termination and TCP connection management remove a significant amount of processing from the front-line servers.
- Additionally, dynamic caching, compression, and prefetching results in better than a 50 percent performance increase for end users.
- Some providers offering this service include:
  - **AT&T Hosting**
  - **Citrix NetScaler**
  - **F5's WebAccelerator**
- This method is mostly oriented toward the cloud service provider, but in the end it benefits the end user.
- Organizations opting for this method of connectivity should look at SLAs and monthly bandwidth charges, rather than worry about what acceleration methods the service provider is adding.

3. Optimized Internet Overlay

An optimized Internet overlay approach allows customers to access the cloud via the public Internet, but enhancement occurs on the provider's cloud.

Enhancements at these points of presence (POP) include:

- Optimized real-time routing. This helps avoid slowdowns, helping to make SLAs easier to attain.
- An SSL session can be stopped so that protocols and payload can be optimized and re-encrypted.
- Some of the application logic can reside on the POP. This allows for better scalability, fault tolerance, and response time, usually in excess of 80 percent.
- Content that is frequently accessed can be delivered from local caches.

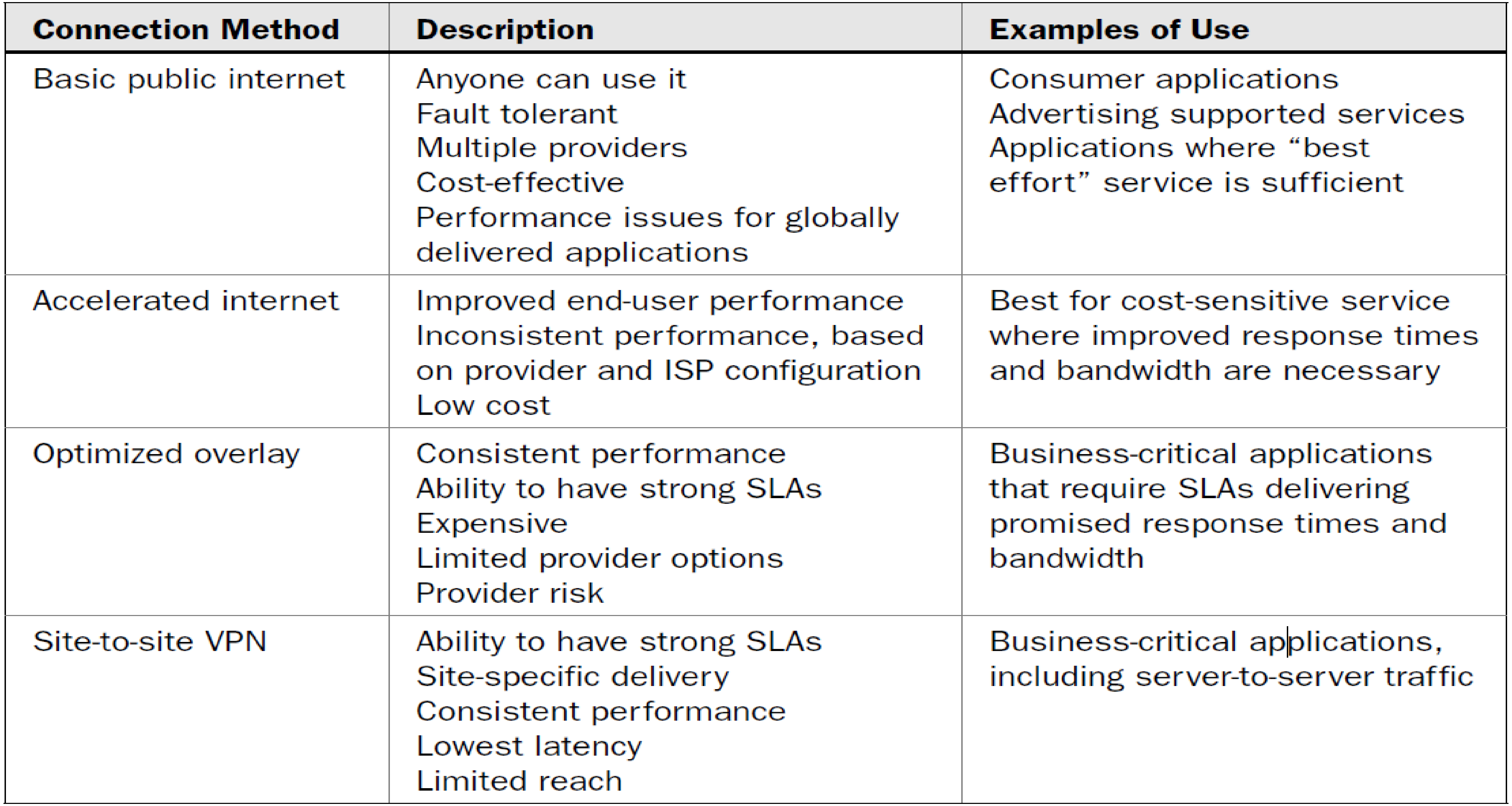
Disadvantages of this method include:

- It is costlier than public Internet connectivity, sometimes as much as four times as much.
- There is a strong vendor lock-in if the application is distributed into the carrier's network.

4. Site-to-Site VPN

- The fourth option is to connect to the service provider directly using a private wide area network (WAN) (normally an MPLS/VPN connection).
- This setup allows confidentiality, guaranteed bandwidth, and SLAs for availability, latency, and packet loss.
- MPLS can also scale to meet changing bandwidth needs, and QoS can also be written into the SLAs.
- On the downside, private WANs are not normally more reliable than Internet connections, especially redundant connections to multiple ISPs.
- Table 5-1 compares all four connections.

Comparison of all 4 levels



Connection Method	Description	Examples of Use
Basic public internet	Anyone can use it Fault tolerant Multiple providers Cost-effective Performance issues for globally delivered applications	Consumer applications Advertising supported services Applications where "best effort" service is sufficient
Accelerated internet	Improved end-user performance Inconsistent performance, based on provider and ISP configuration Low cost	Best for cost-sensitive service where improved response times and bandwidth are necessary
Optimized overlay	Consistent performance Ability to have strong SLAs Expensive Limited provider options Provider risk	Business-critical applications that require SLAs delivering promised response times and bandwidth
Site-to-site VPN	Ability to have strong SLAs Site-specific delivery Consistent performance Lowest latency Limited reach	Business-critical applications, including server-to-server traffic

Cloud Providers

- Cloud providers that use services dispersed across the cloud need a robust connection method.
- Private tunnels make sure that bandwidth, latency, and loss aren’t as likely to affect performance.
- Plus, encryption and strong authentication offer another benefit.
- Cloud providers that are growing might face big costs as network bandwidth charges increase.
- This traffic is from traffic both to and from clients as well as traffic among provider sites.
- Big providers, like Google, are able to sidestep these charges by building their own WANs with multiple peering points with major ISPs.
- Smaller providers can use WAN optimization controllers (WOCs) to reduce bandwidth requirements by up to 80 percent.

Cloud Consumers

- Large companies can build their own scalable distributed IT infrastructure in which datacenters are connected with their own private fiber optic connections.
- This depends on distance, bandwidth requirements, and—of course—their budgets.
- This infrastructure starts to look like a cloud computing service.
- Clients located at major sites normally access applications over the corporate WAN.
- For smaller offices or mobile workers, VPN connections across optimized and accelerated Internet services provide a more robust solution.
- VPN tunnels across the Internet are best as a primary link only when high performance is not crucial.

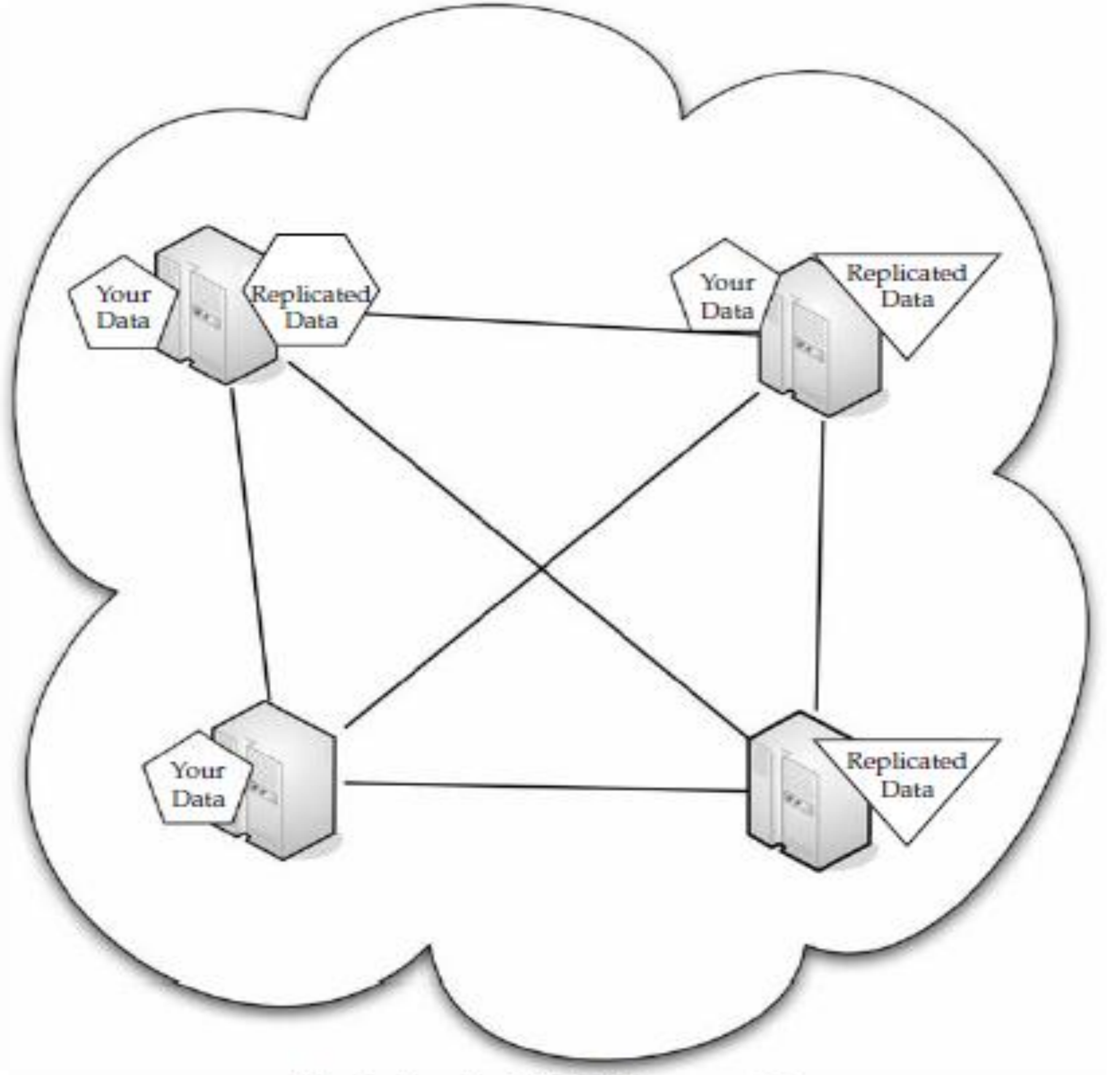
Pipe Size

- Bandwidth is, simply put, the transmission speed or throughput of your connection to the Internet.
- But, measuring bandwidth can be difficult, since the lowest point of bandwidth between your computer and the site you're looking at is what your speed is at that moment.
- There are three factors that are simply out of your control when it comes to how much bandwidth you need:
  - The Internet bandwidth between your organization and the cloud
  - The round-trip time between your organization and the cloud
  - The response time of the cloud

Upstream/Downstream

- Another factor to consider is whether it is okay for the transfers to be symmetric or asymmetric.
- If your connection with the cloud is symmetric, then that means you are sending and receiving data at the same rate.
- If your connection is asymmetric, then data is sent from your organization at a slower rate than you're receiving it.

Redundancy



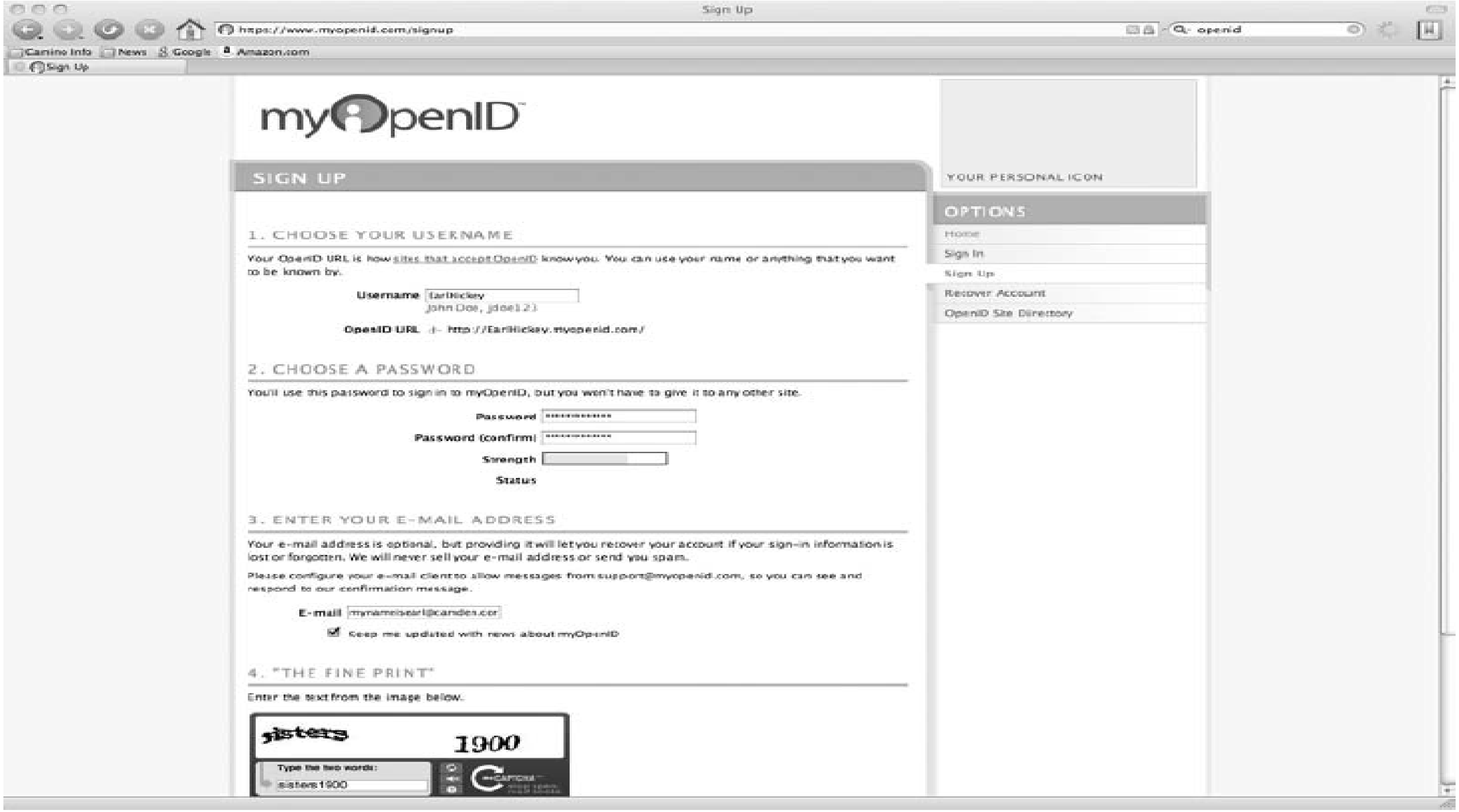
The cloud vendor is likely to have your data
and its redundant clone in geographically dispersed locations.

- When formulating your cloud infrastructure, be sure to consider the issue of reliability and uptime and ask your service provider to configure your computing infrastructure for redundancy and failover.
- In your LAN, redundancy used to mean that another server or two were added to the datacenter in case there was a problem.
- These days with virtualization, redundancy might mean a virtual server being cloned onto the same device, or all the virtual servers of one machine being cloned onto a second physical server.

Services

1. Identity:

- No matter where an application runs—in-house or on the cloud—it needs to know about its users.
- To accomplish this, the application asks for a digital identity—a set of bytes—to describe the user.
- Based on this information, the application can determine who the user is and what he or she is allowed to do.
- In-house applications rely on services like Active Directory to provide this information.
- Clouds, however, have to use their own identity services. For instance, if you sign on to Amazon cloud services, you have to sign on using an Amazon-defined identity.
- Google's App Engine requires a Google account, and Windows uses Windows Live ID for use with Microsoft's cloud applications.
- Identity services need not be proprietary. OpenID is an open, decentralized, single signon standard that allows users to log in to many services using the same digital identity.
- An OpenID is in the form of a uniform resource locator (URL) and does not rely on a central authority to authenticate a user's identity.
- Since a specific type of authentication is not required, nonstandard forms of authentication may be used, including smart cards, biometric, or passwords. An OpenID registration is shown in Figure 5-2.
- OpenID authentication is used by many organizations, including:
  - Google
  - IBM
  - Microsoft
  - Yahoo!



SIGN UP

1. CHOOSE YOUR USERNAME

Your OpenID URL is how sites that accept OpenID know you. You can use your name or anything that you want to be known by.

Username EarlHickey
John Doe, jdoe123

OpenID URL http://EarlHickey.myopenid.com/

2. CHOOSE A PASSWORD

You'll use this password to sign in to myOpenID, but you won't have to give it to any other site.

Password *************
Password (confirm) *************
Strength
Status

3. ENTER YOUR E-MAIL ADDRESS

Your e-mail address is optional, but providing it will let you recover your account if your sign-in information is lost or forgotten. We will never sell your e-mail address or send you spam.

Please configure your e-mail client to allow messages from support@myopenid.com, so you can see and respond to our confirmation message.

E-mail mynameisearl@camden.cor
☑ Keep me updated with news about myOpenID

4. "THE FINE PRINT"

Enter the text from the image below.

sisters 1900
Type the two words:
sisters1900

Figure 5-2 OpenID is a means to keep login information consistent across several sites.

2. Integration:

- Vendors come up with all sorts of on-premises infrastructure services to accomplish it.
- These range from technologies like message queues to complex integration servers.
- For example, Amazon's Simple Queue Service (SQS) provides a way for applications to exchange messages via queues in the cloud.
- Another example of cloud-based integration is BizTalk Services. Instead of using queuing, BizTalk Services utilizes a relay service in the cloud, allowing applications to communicate through firewalls.

- **3. Mapping**
- Maps are becoming more and more popular in web applications.
- For instance, hotel and restaurant web sites show their locations on their web sites and allow visitors to enter their addresses to get customized directions.
- But the guy who developed the web site likely didn’t have the time or money (not to mention the interest) to make his own mapping database.
- Enough organizations want this functionality, however, so it is offered as a cloud application.
- Such services as Google Maps and Microsoft’s Virtual Earth provide this cloud-based function, allowing developers to embed maps in web pages.

4. Payments

- Another cloud service that you might want to plan for and configure your hardware appropriately for is payments.
- Depending on your organization, you may or may not want to accept online payments from customers.
- You can simply sign up with a service to accept credit cards, or you can go the route of PayPal. With an online payment service, customers can send money directly to your organization.

5. Search

- The ability to embed search options in a web site is certainly nothing new, but it is a rich feature that you might want to employ in your own web or application development.
- Microsoft's Live Search allows on-site and cloud applications to submit searches and then get the results back.

Accessing the Cloud

Platforms:

- A platform is how a cloud computing environment is delivered to you.

Web Application Framework:

- A web application framework is used to support the development of dynamic web sites, web applications, and web services.
- The point of a framework is to reduce the overhead that comes with common activities in web development.
- For instance, frameworks provideWeb Application Framework A web application framework is used to support the development of dynamic web sites, web applications, and web services.
- The point of a framework is to reduce the overhead that comes with common activities in web development.
- For instance, frameworks provide libraries that are already written so the developer doesn't have to reinvent the wheel every time a web site is developed.
- Early in the Web's life, hypertext was mostly hand-coded Hypertext Markup Language (HTML) that was published on Web servers.
- If a published page needed to be changed, it had to be done by the page's author.
- As the Web grew up, it became more dynamic with the addition of the Common Gateway Interface (CGI). This allowed external applications to interface with web servers.

AJAX:

- Asynchronous JavaScript and XML (AJAX) is a group of web development techniques used for creating interactive web applications.
- By using AJAX, web applications can retrieve data from the server asynchronously.
- Because it is being done in the background, it won’t interfere with the display and behavior of the current page.
- Technologies AJAX is a term that represents a wide range of web technologies that can be used to help web applications communicate with a server, but without interfering with the current state of that page.

AJAX refers to these technologies:

- Extensible Hypertext Markup Language (XHTML) and Cascading Style Sheets (CSS) for presentation
- The Document Object Model for dynamic display of and interaction with data (DOM)
- XML and Extensible Style Sheet Language Transformations (XSLT) for the interchange and manipulation of data, respectively
- The `XMLHttpRequest` object for asynchronous communication
- JavaScript to bring these technologies together.
- JavaScript Object Notation (JSON) is a widely used alternative to XML.

Pros and Cons AJAX :

- Often, multiple pages on a web site contain the same information. If those pages were coded by hand, the same content would have to be written into each and every page. AJAX allows a web application to simply retrieve new information and adjust how the content is presented. This is very efficient and reduces the amount of bandwidth consumed and reduces load times.
- Using asynchronous requests allows the client's web browser to be more interactive and respond quickly to user inputs. The user may even perceive the application to be faster.
- Connections to the server are reduced, because scripts and style sheets need only be downloaded once.

Disadvantages to AJAX include:

• Dynamically created web pages do not show up in the browser's history engine, so clicking on the Back button would not re-create the last seen page.

• It is difficult to bookmark a dynamically created web page.

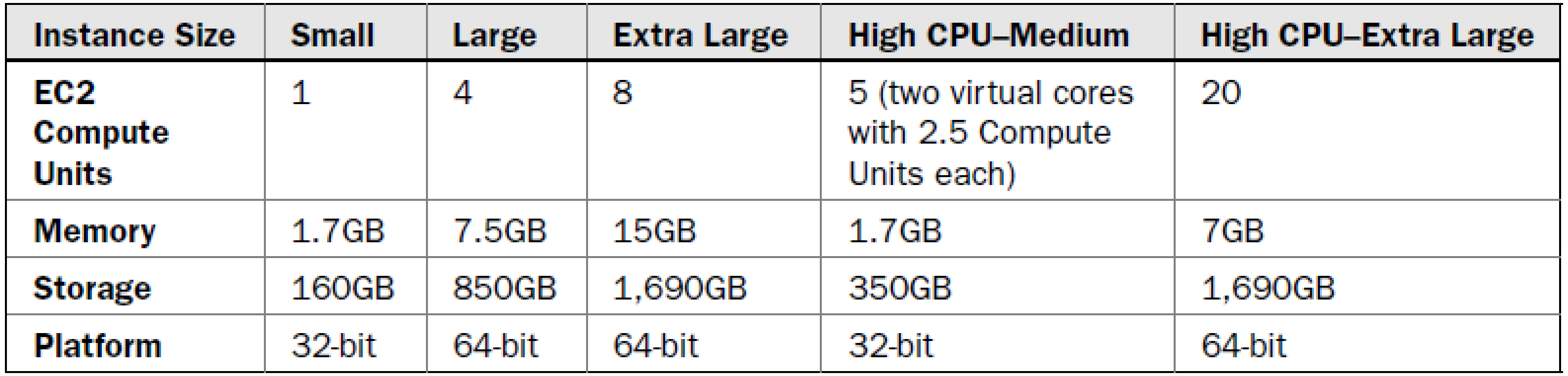
• If a browser does not support AJAX or if JavaScript is disabled, AJAX functionality cannot be used.

• There is no standards body behind AJAX, so there is no widely adopted best practice to test AJAX applications.

- Python Django
- Django is an open-source web application framework written in Python.
- Originally it was created to manage news sites for The World Company and released publicly under a BSD license in July 2005.
- In June 2008 it was announced that the Django Software Foundation will be the authority for Django.

Web Hosting Service

- Amazon Elastic Compute Cloud
- Mosso- RackSpace



Instance Size	Small	Large	Extra Large	High CPU–Medium	High CPU–Extra Large
EC2 Compute Units	1	4	8	5 (two virtual cores with 2.5 Compute Units each)	20
Memory	1.7GB	7.5GB	15GB	1.7GB	7GB
Storage	160GB	850GB	1,690GB	350GB	1,690GB
Platform	32-bit	64-bit	64-bit	32-bit	64-bit

TABLE 6-1 Amazon Elastic Compute Cloud Sizes

- Azure
- Salesforce-Force.com
- Visualforce

Web Applications

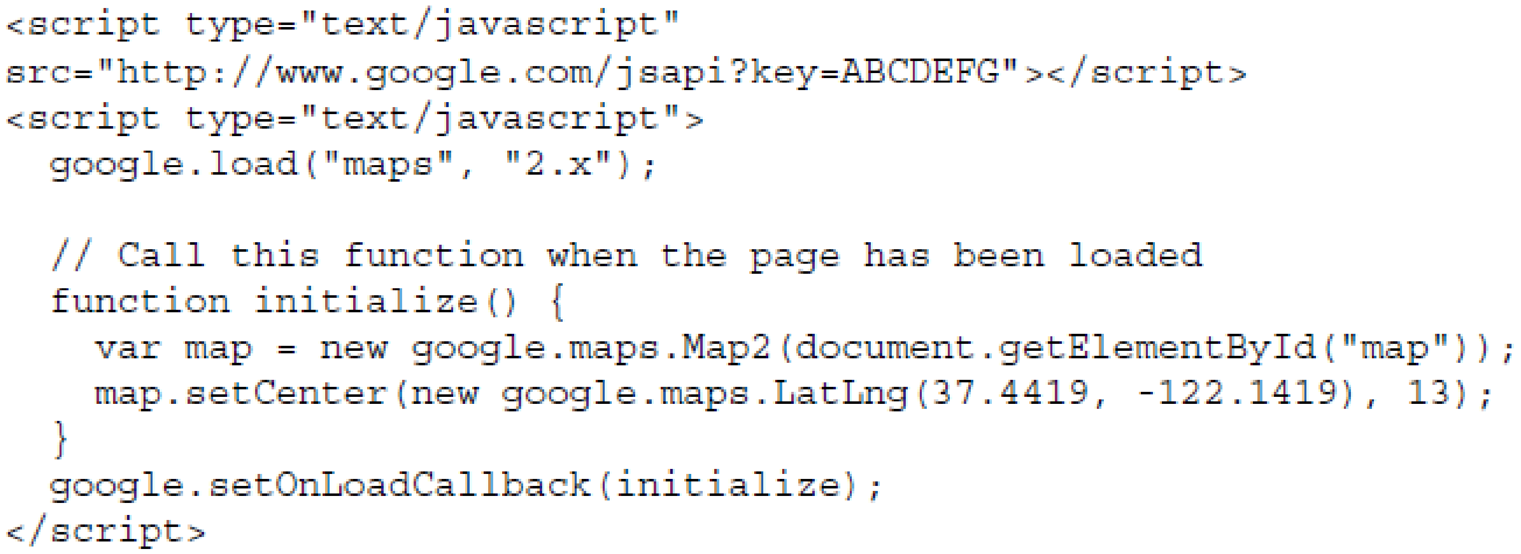
- Gmail webmail services
- Google Calendar shared calendaring
- Google Talk instant messaging and Voice Over IP
- Start Page for creating a customizable home page on a specific domain.
- Google Docs and Spreadsheets
- Google-hosted applications are available in many languages, such as French, Italian,
- German, Spanish, Chinese, Japanese, and Korean. You can find more information at
- http://www.google.com/.

Web APIs

- An application programming interface (API) is a set of programming instructions and standards for accessing a web-based program.
- Software companies release their APIs to the public so that other software developers can design products that are powered by its service.
- For example, Amazon released its own API so that web site developers could more easily access information maintained at the Amazon web site.
- By using Amazon’s API, a third-party web site can directly link to products on the Amazon site.
- APIs allow one program to speak with another. They are not user interfaces.
- Using APIs, programs can speak to each other without the user having to be involved.
- For instance, when you buy something at Amazon and enter your credit card information, Amazon uses an API to send your credit card information to a remote application that verifies whether your information is correct.
- As a user, all you saw was the place to enter your credit card information, but behind the scenes, APIs were getting the job done.
- An API is similar to Software as a Service (SaaS), because software developers don’t have to start from scratch every time they write a program.
- Rather than build one program that does everything (email, billing tracking, and so forth), the application can farm out those duties to other applications that do it better

How APIs Work???

- An API is (as the acronym says) an interface that defines the way in which two things will communicate.
- With APIs, the calls back and forth are managed by web services.
- Web services are a collection of standards including XML, the programming language that allows applications to communicate over the Internet.
- XML is a general-purpose markup language. It describes structured data in a way that both humans and computers can
- The API is a piece of software code written as a series of XML messages, like the one for the Google Maps API shown here: read and write.



```
<script type="text/javascript"
src="http://www.google.com/jsapi?key=ABCDEFG"></script>
<script type="text/javascript">
  google.load("maps", "2.x");

  // Call this function when the page has been loaded
  function initialize() {
    var map = new google.maps.Map2(document.getElementById("map"));
    map.setCenter(new google.maps.LatLng(37.4419, -122.1419), 13);
  }
  google.setOnLoadCallback(initialize);
</script>
```

Other stds for API to work

- **SOAP (Simple Object Access Protocol)** SOAP encodes XML messages so that
- they can be received and understood by any operating system over any type of
- network protocol.
- • **UDDI (Universal Description, Discovery, and Integration)** UDDI is an XMLbased
- directory that allows businesses to list themselves, find each other, and
- collaborate using web services.
- • **WSDL (Web Services Description Language)** WSDL is the SOAP of UDDI. WSDL
- is the XML-based language that businesses use to describe their services in the UDDI.

Google Data APIs

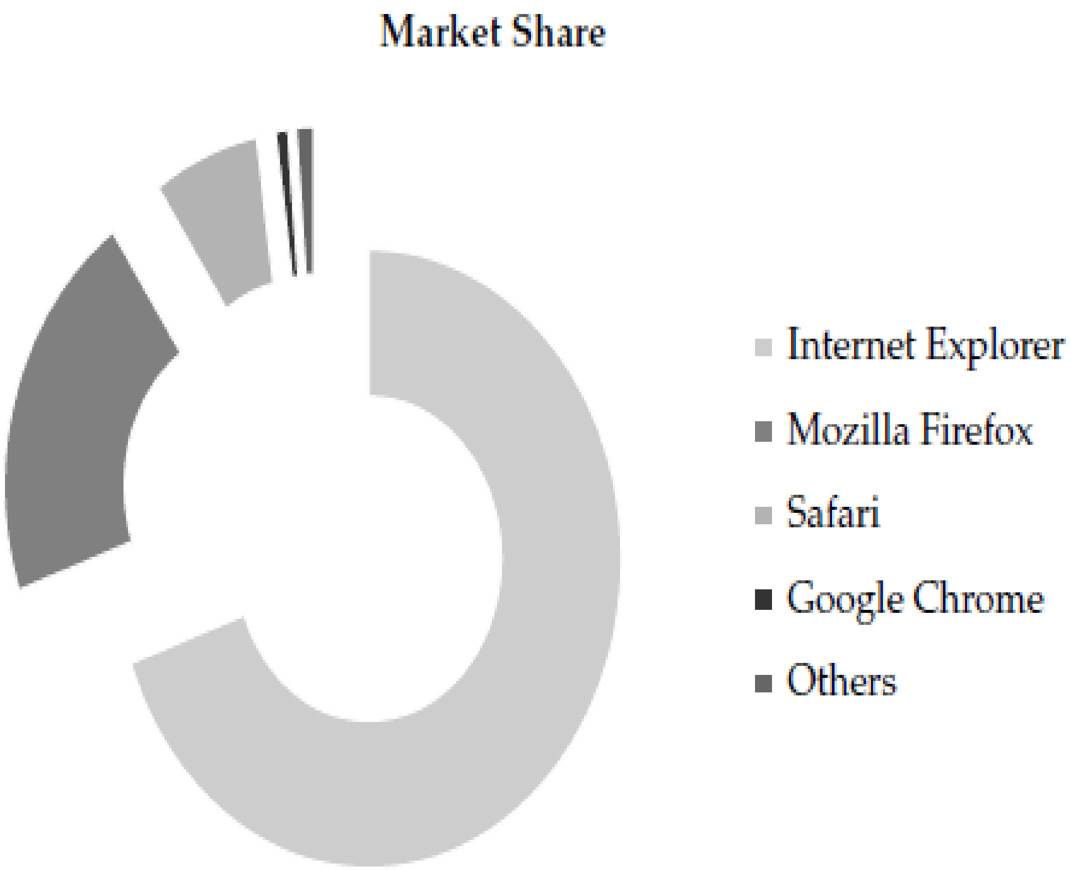
These REST-style APIs are based on the Atom Publishing Protocol (AtomPub), and use the Atom syndication format to represent data and HTTP to handle communication.

- The Google Data APIs include:
- Google Apps APIs
- Google Base Data API
- Blogger Data API
- Google Book Search Data API
- Google Calendar Data API
- Google Code Search Data API
- Google Contacts Data API
- Google Documents List Data API
- Google Finance Portfolio Data API
- Google Health Data API
- Google Notebook Data API
- Picasa Web Albums Data API
- Google Spreadsheets Data API
- Webmaster Tools Data API
- YouTube Data API

GoGrid:

- GoGrid’s API is a web service that allows developers to control their interaction with GoGrid’s
- cloud hosting infrastructure. The GoGrid API provides two-way communication for controlling
- GoGrid’s control panel functionality. Typical uses for the API include
  - Auto-scaling network servers
  - Listing assigned public and private IP addresses
  - Deleting servers
  - Listing billing details
- GoGrid’s REST-like API Query interface is designed for individuals who want to
- programmatically control their cloud hosting infrastructure over the Internet.
- The GoGrid API requires you to be a GoGrid customer and to have technical knowledge
- and programming skills. The GoGrid API supports these languages:
  - Java
  - PHP
  - Python
  - Ruby

Web Browsers



Eg: Chrome Cloud

There's a lot of buzz around Chrome being a great tool for cloud computing. It extends the cloud into your organization's computer, and vice versa. This is mainly because of the power of the V8 JavaScript engine and built-in Google Gear.

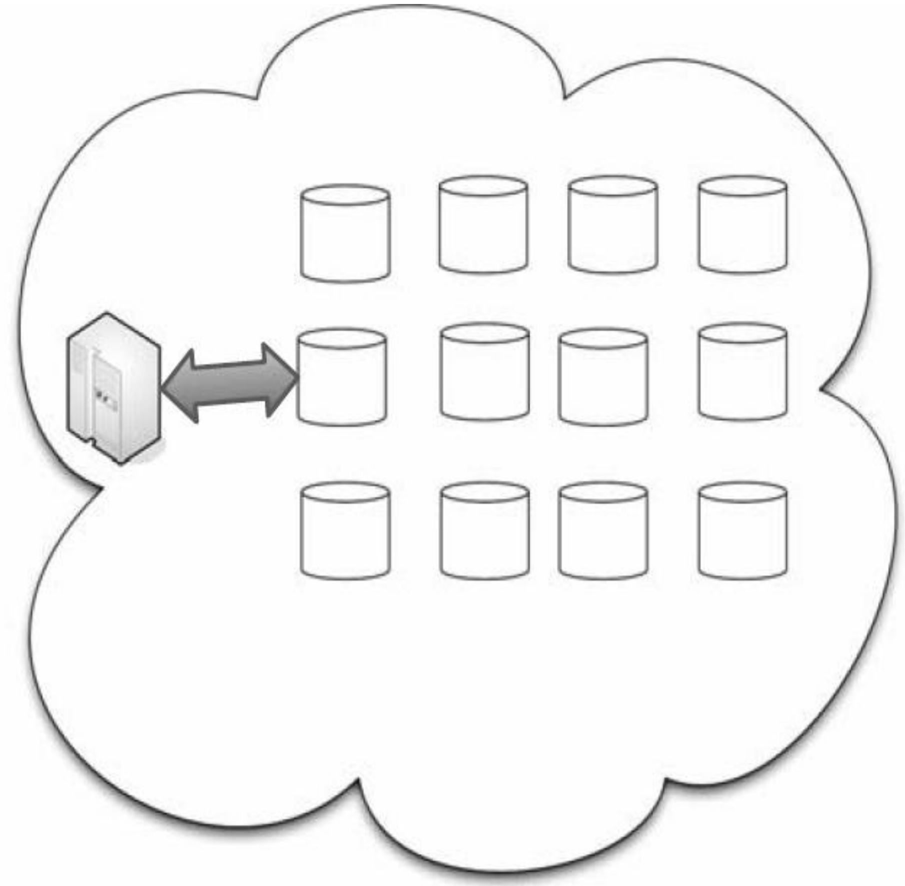
Google Gears are also open source, and they enable powerful web applications by adding new features to the web browser.

Major API components to Gears include:

- A database module that can store data locally
- A WorkerPool module that provides parallel execution of JavaScript code
- A LocalServer module that caches and serves application resources (like HTML, JavaScript, images, and so on)
- A Desktop module that lets web applications interact more naturally with the desktop
- A Geolocation module that lets web applications detect the geographical location of their users

Cloud Storage

- Cloud storage has a number of advantages over traditional data storage.
- If you store your data on a cloud, you can get at it from any location that has Internet access.
- Also, if your organization has branch offices; they can all access the data from the cloud provider.
- Many cloud storage systems like niche-oriented and store just email or digital pictures, while others store any type of data.
- Some providers are small, while others are huge and fill an entire warehouse.
- Because servers require maintenance or repair, it is necessary to store the saved data on multiple machines, providing redundancy.
- Without that redundancy, cloud storage systems couldn't assure clients that they could access their information at any given time.
- Most systems store the same data on servers using different power supplies.
- That way, clients can still access their data even if a power supply fails.



A cloud service provider can simply add more commodity hard drives to increase the organization's capacity.

Storage as a Service

- The term Storage as a Service (another Software as a Service, or SaaS, acronym) means that a third-party provider rents space on their storage to end users who lack the budget or capital budget to pay for it on their own.
- It is also ideal when technical personnel are not available or have inadequate knowledge to implement and maintain that storage infrastructure.
- Storage service providers are nothing new, but given the complexity of current backup, replication, and disaster recovery needs, the service has become popular, especially among small and medium-sized businesses.
- The biggest advantage to SaaS is cost savings. Storage is rented from the provider using a **cost-per-gigabyte-stored or cost-per-data-transferred model.**
- The end user doesn’t have to pay for infrastructure; they simply pay for how much they transfer and save on the provider’s servers.
- A customer uses client software to specify the backup set and then transfers data across a WAN. When data loss occurs, the customer can retrieve the lost data from the service provider

Providers

- Google Docs allows users to upload documents, spreadsheets, and presentations to Google’s data servers. Those files can then be edited using aGoogle application.
- Web email providers like Gmail, Hotmail, and Yahoo! Mail store email messages on their own servers. Users can access their email from computers and other devices connected to the Internet.
- Flickr and Picasa host millions of digital photographs. Users can create their own online photo albums.
- YouTube hosts millions of user-uploaded video files.
- Hostmonster and GoDaddy store files and data for many client web sites.
- Facebook and MySpace are social networking sites and allow members to post pictures and other

  content. That content is stored on the company’s servers.
- MediaMax and Strongspace offer storage space for any kind of digital data.

Security

To secure data, most systems use a combination of techniques:

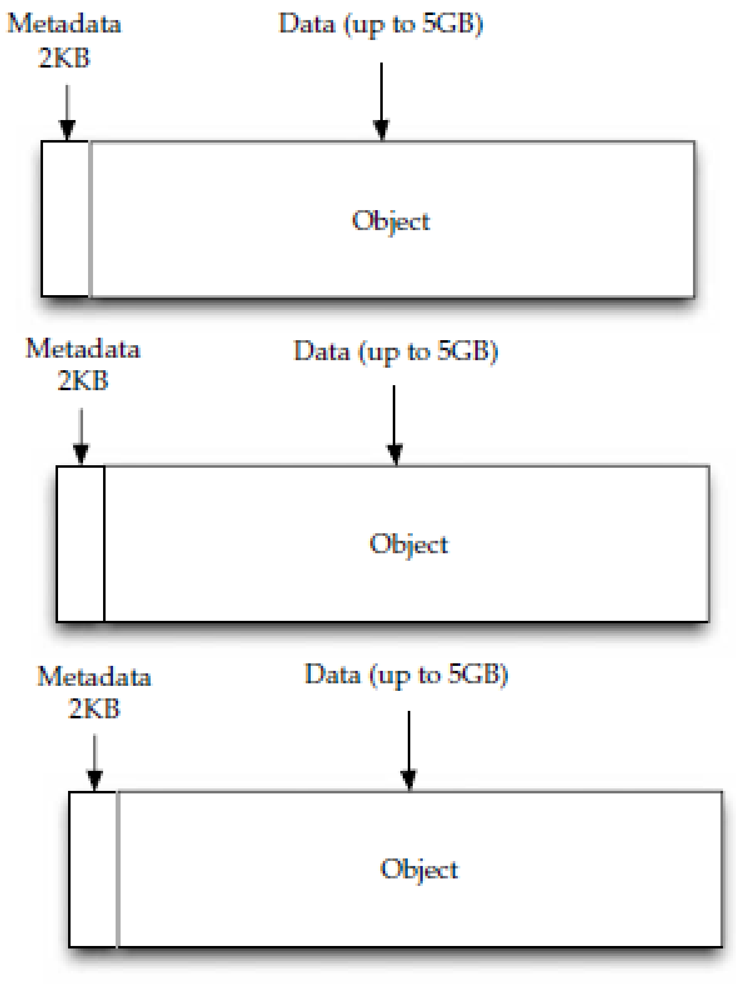
- **Encryption**
- **Authentication processes**
- **Authorization practices**

Advantages:

- Cloud storage is becoming an increasingly attractive solution for organizations.
- Reason is with cloud storage, data resides on the Web, located across storage systems rather than at a designated corporate hosting site.
- Cloud storage providers balance server loads and move data among various datacenters, ensuring that information is stored close—and thereby available quickly—to where it is used.
- Storing data on the cloud is advantageous, because it allows you to protect your data in case there’s a disaster.

Cloud Storage Providers

- **Amazon Simple Storage Service (S3)**



Multiple objects are stored in buckets in Amazon S3.

Nirvanix

- Nirvanix uses custom-developed software and file system technologies running on Intel storage servers at six locations on both coasts of the United States.
- They continue to grow, and expect to add dozens more server locations.

The benefits of cloud network attached storage (CloudNAS) include:

• Cost savings of 80–90 percent over managing traditional storage solutions

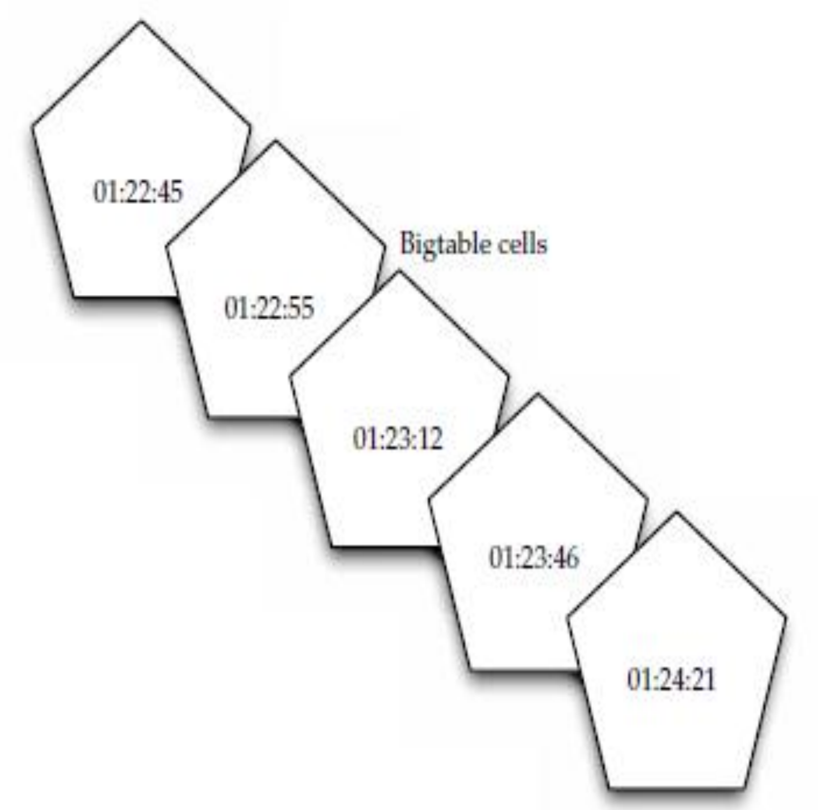
• Elimination of large capital expenditures while enabling 100 percent storage
utilization

• Encrypted offsite storage that integrates into existing archive and backup processes

• Built-in data disaster recovery and automated data replication on up to three
geographically dispersed storage nodes for a 100 percent SLA

• Immediate availability to data in seconds, versus hours or days on offline tape

- Google Bigtable Datastore



In Google Bigtable, multiple copies of a cell exist, each with a different timestamp.

- Because the tables are so large, Bigtable splits them at row boundaries and saves them as tablets.
- Each tablet is about 200MB, and each server houses 100 tablets.
- Given this, data from a database is likely to be stored in many different servers—maybe not even in the same geographic location.
- This architecture also allows for load balancing

MobileMe:

- MobileMe is Apple's solution that delivers push email, push contacts, and push calendars from the MobileMe service in the cloud to native applications on iPhone, iPod touch, Macs, and PCs.
- MobileMe also provides a suite of ad-free web applications that deliver a desktoplike experience through any modern browser.
- MobileMe applications (www.me.com) include Mail, Contacts, and Calendar, as well as Gallery for viewing and sharing photos and iDisk for storing and exchanging documents online.

Live Mesh:

- Live Mesh is Microsoft's "software-plus-services" platform and experience that enables PCs and other devices to be aware of each other through the Internet, enabling individuals and organizations to manage, access, and share their files and applications seamlessly on the Web and across their world of devices.

Standards

- Standards make the World Wide Web go around, and by extension, they are important to cloud computing.

Application:

- A cloud application is the software architecture that the cloud uses to eliminate the need to install and run on the client computer.

1. **Communication:** HTTP, XMPP

2. **Security:** SSl, OPenID, PCI DSS

3. **Client:** HTML, DHTML, XHTML, JS,

Infrastructure

Virtualization: VMware, AMD, BEA Systems, BMC Software, Broadcom, Cisco, Computer Associates International, Dell, Emulex, HP, IBM, Intel, Mellanox, Novell, QLogic, and Red Hat all worked together to advance open virtualization standards.

Open Hypervisor Standards:

- Hypervisors are the foundational component of virtual infrastructure and enable computer system partitioning.
- An open-standard hypervisor framework can benefit customers by enabling innovation across an ecosystem of interoperable virtualization vendors and solutions.

- Collaboration around open hypervisor standards is expected to focus on the following areas of interoperability and performance optimization for virtualized environments:
  - Cross-platform frameworks that govern the standardized operation and management of stand-alone virtual machine environments as well as highly dynamic, data center-scale deployment of virtualized systems
  - Cooperative virtualization APIs between hypervisors and guest operating systems
  - Virtual machine formats that enable virtual machine migration and recovery across platforms

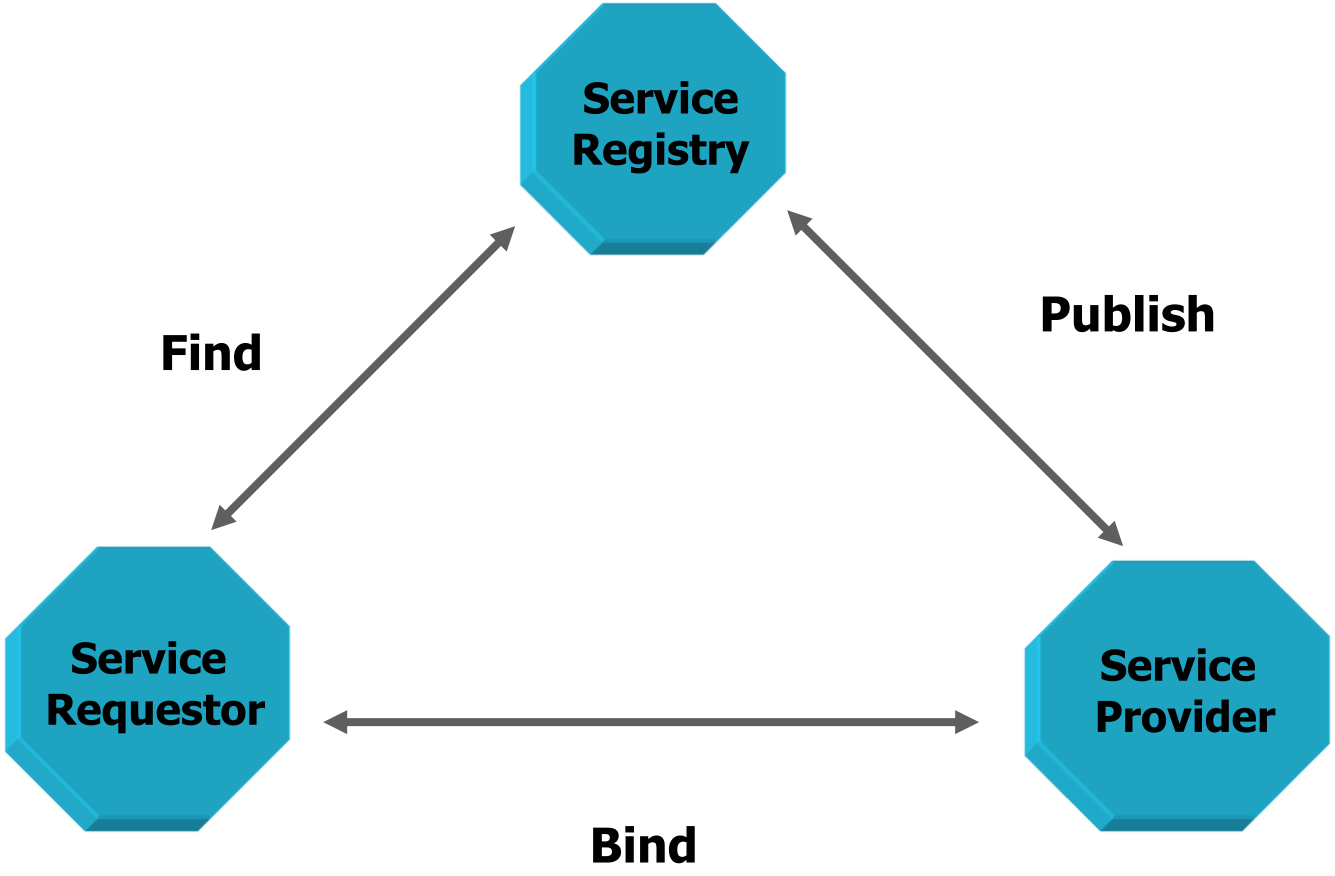
Service

- A *web service*, as defined by the World Wide Web Consortium (W3C), “is a software system designed to support interoperable machine-to-machine interaction over a network” that may be accessed by other cloud computing components.
- Web services are often web APIs that can be accessed over a network, like the Internet, and executed on a remote system that hosts the requested services.
- Popular Web services are:
- **Data**
- Data can be stirred and served up with a number of mechanisms; two of the most popular are JSON and XML.
- Both are based on leading industry standards—HTML and JavaScript—to help deliver and present data.
- **JSON**
- JSON is short for JavaScript Object Notation and is a lightweight computer data interchange format.
- It is used for transmitting structured data over a network connection in a process called *serialization*.
- It is often used as an alternative to XML.
- **JSON Basics**
- JSON is based on a subset of JavaScript and is normally used with that language.
- However, JSON is considered to be a language-independent format, and code forparsing and generating JSON data is available for several programming languages.
- This makes it a good replacement for XML when JavaScript is involved with the exchange of data, like AJAX.

What is a web service?

- **W3C Definition**:
  - A Web service is a software application identified by a URI, whose interfaces and binding are capable of being defined, described and discovered by XML artefacts and supports direct interactions with other software applications using XML based messages via internet-based protocols
- **Other definitions**
  - “Web services” is an effort to build a distributed computing platform for the Web.
  - enabling systematic application-to-application interaction on the Web.

Service Oriented Arquitecture

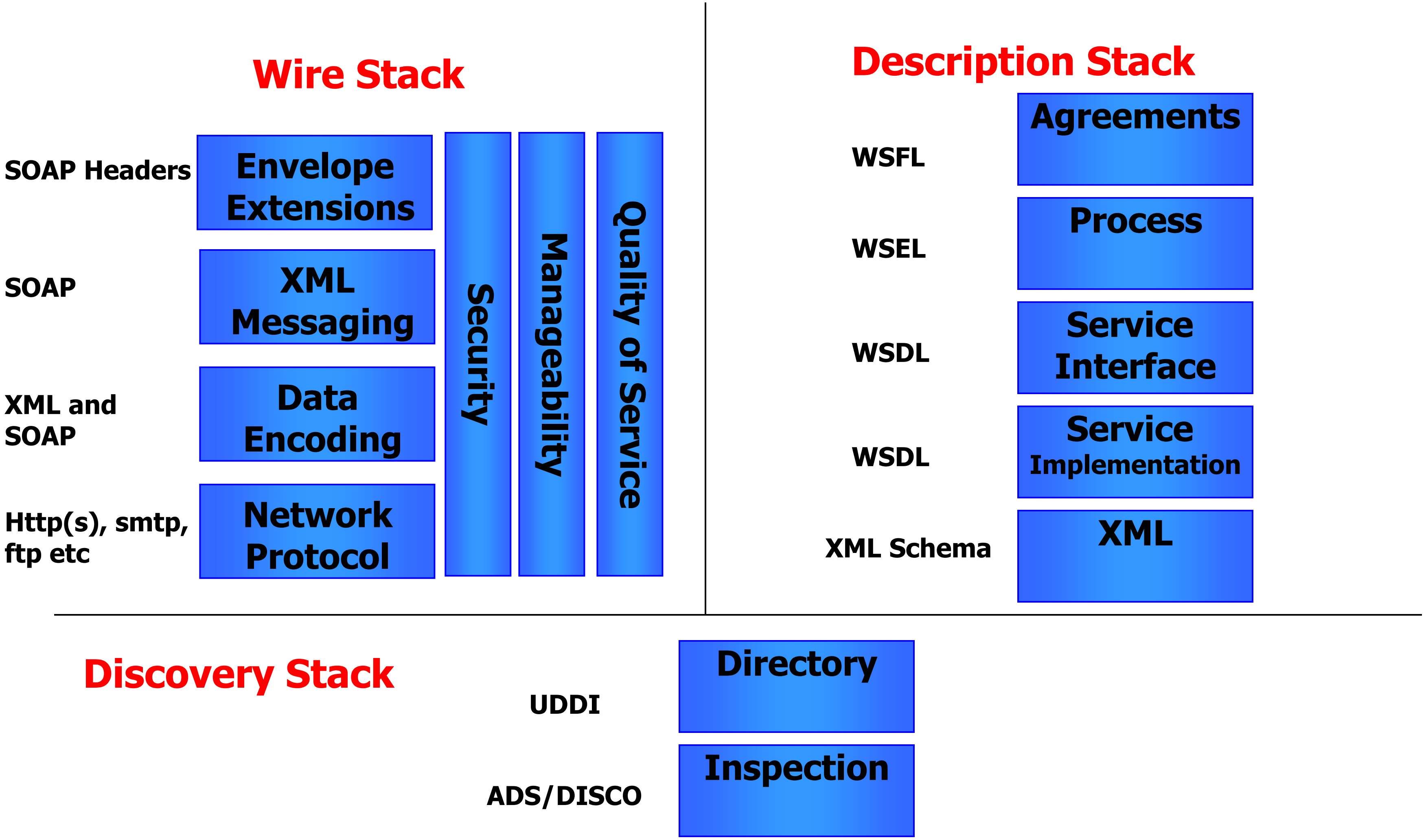


According to the SOA

A Web service is:

- An interface that describes a collection of network accessible operations
- Described using a service description language
- Published by making this service description available to users
- Found by sending queries to a registry matching service descriptions
- Bound-Invoked by using the information contained in the service description
- Composed with other services to create new services (service orchestration)

Web services interoperability stack



Wire Stack

SOAP Headers	Envelope Extensions
SOAP	XML Messaging
XML and SOAP	Data Encoding
Http(s), smtp, ftp etc	Network Protocol

Security

Manageability

Quality of Service

Description Stack

WSFL	Agreements
WSEL	Process
WSDL	Service Interface
WSDL	Service Implementation
XML Schema	XML

Discovery Stack

UDDI	Directory
ADS/DISCO	Inspection

WSDL – Web Services description Language
UDDI – Universal Description, Discovery and Interaction

eXtensible Markup Language

- All the technologies in Web Services are XML based
    - Messaging
    - Description
    - Registry

Are all in XML

- Why?
    - XML is pure text with no binary data
    - Applications read the XML
    - Applications share data using XML . Any application can talk to any other application using XML (unlike binary) irrespective of the platform
    - XML is a method for putting structured data in a text file

XML Document

```
<?XML version=“1.0” encoding=“UTF-8” standalone=“no”?>
<!-- this is an XML comment -->
<books xmlns=“somename"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation=“somename M:\XML\Schemas\docbook.xsd">
   <book year=“2000” book-title=“XML in Depth”>
        <author>
        <title>Mr.</title> John Doe
        </author>
        <publisher> &pub </publisher>
   </book>
</books>
```

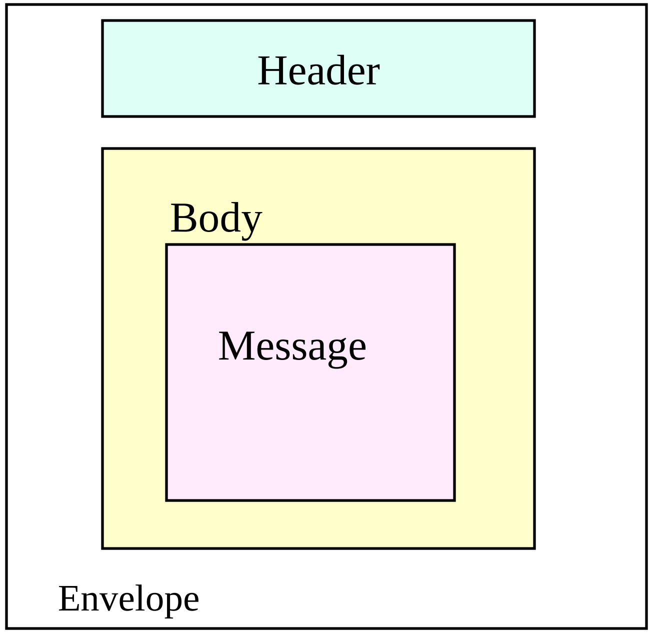
XML instance   schema   *comment*   root   attributes   element

XML Rules

- Well formed
  - Syntax is correct (all tags opened and closed)
- Valid
  - All the elements matches the definitions written in the schema
- XML Documents (.xml) == XML Instances of the Schema (.xsd)
- DTD
  - Document Type Definitions – Validates XML data against it
- XML Schema
  - Alternative to DTD with added functionality. It supports other data types not supported by DTD
    - Predefined Simple Types (integers,booleans,dateTime…)
    - User-defined datatypes (Complex Types)
    - Validations Restrictions to types
  - XML schema itself is an XML document !
- XML Processing
  - Read the XML documents XML processors (Parsers)
    - SAX (based on events)
    - DOM (reads the xml document and loads it in memory)
    - Python implements this interfaces in a package PyXML

SOAP: Simple Object Access Protocol

- **An Internet standard specification, the goal of which is to define a platform and vendor-neutral WIRE PROTOCOL based on Internet standard protocols [HTTP & XML] to access Web Services**
- **How do we access a service???**
  - **With a SOAP message: Is a XML stream which is used to transmit messages via HTTP**
- **SOAP Structure**
  - **Envelope:contains the entire SOAP message**
  - **Header**
  - **Boby**
    - **Message**



SOAP Example

Soap Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
        <SOAP-ENV:Body>
                <m:calculateCarPayment xmlns:m="http://www.exadel.com/services/CarPayment.xsd">
                        <loanAmount xsi:type="xsd:string">5000</loanAmount>
                        <loanTerm xsi:type="xsd:string">12</loanTerm>
                        <loanRate xsi:type="xsd:string">8</loanRate>
                </m:calculateCarPayment>
        </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

This example sends a request for a web service method called calculateCarPayment with three different arguments

You can try that with XML SPY (v 4.4)

Soap Example II

Soap Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
          <SOAP-ENV:Body>
                    <ns1:calculateCarPaymentResponse
xmlns:ns1="http://www.exadel.com/services/CarPayment.xsd" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
                              <Result xsi:type="xsd:double">434.94</Result>
                    </ns1:calculateCarPaymentResponse>
          </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The response could include Fault elements to describe any error that occurred invoking the service

Invoking a web service

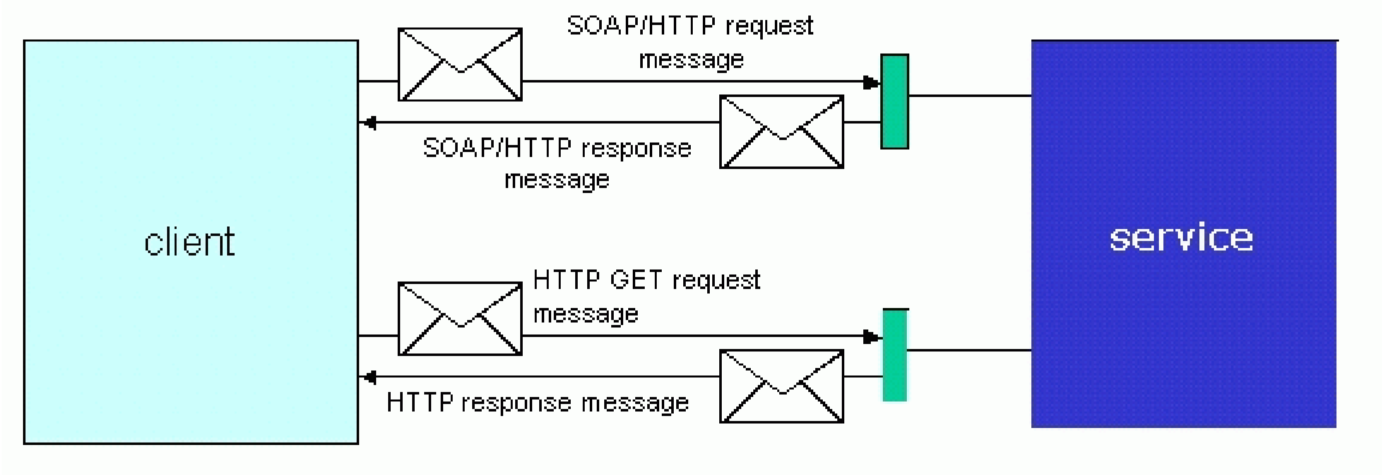


Figure 1. A client invoking a Web service

Soap libraries-engines

- To run SOAP :
    - XML Spy (first try)
- Soap Engine
    - Apache AXIS: Is an application you include inside Tom Cat Has Java API's that talk to SOAP
- Python SOAP Libraries
    - Python SOAP Libraries
        - 4Suite SOAP, administered by Fourthought
        - SOAPy, administered by Adam Elman
        - SOAP.py, a project of the Web services for Python project
            - **SOAP.py 0.9.5 download from http://sourceforge.net/projects/pywebsvcs**
        - soaplib, by Secret Labs
        - Orchard, by Ken MacLeod
        - PySOAP, administered by Dave Warner
- The good part of this is that SOAP is completely transparent to developers when trying to access web services
- You need PyXML to install python soap libraries

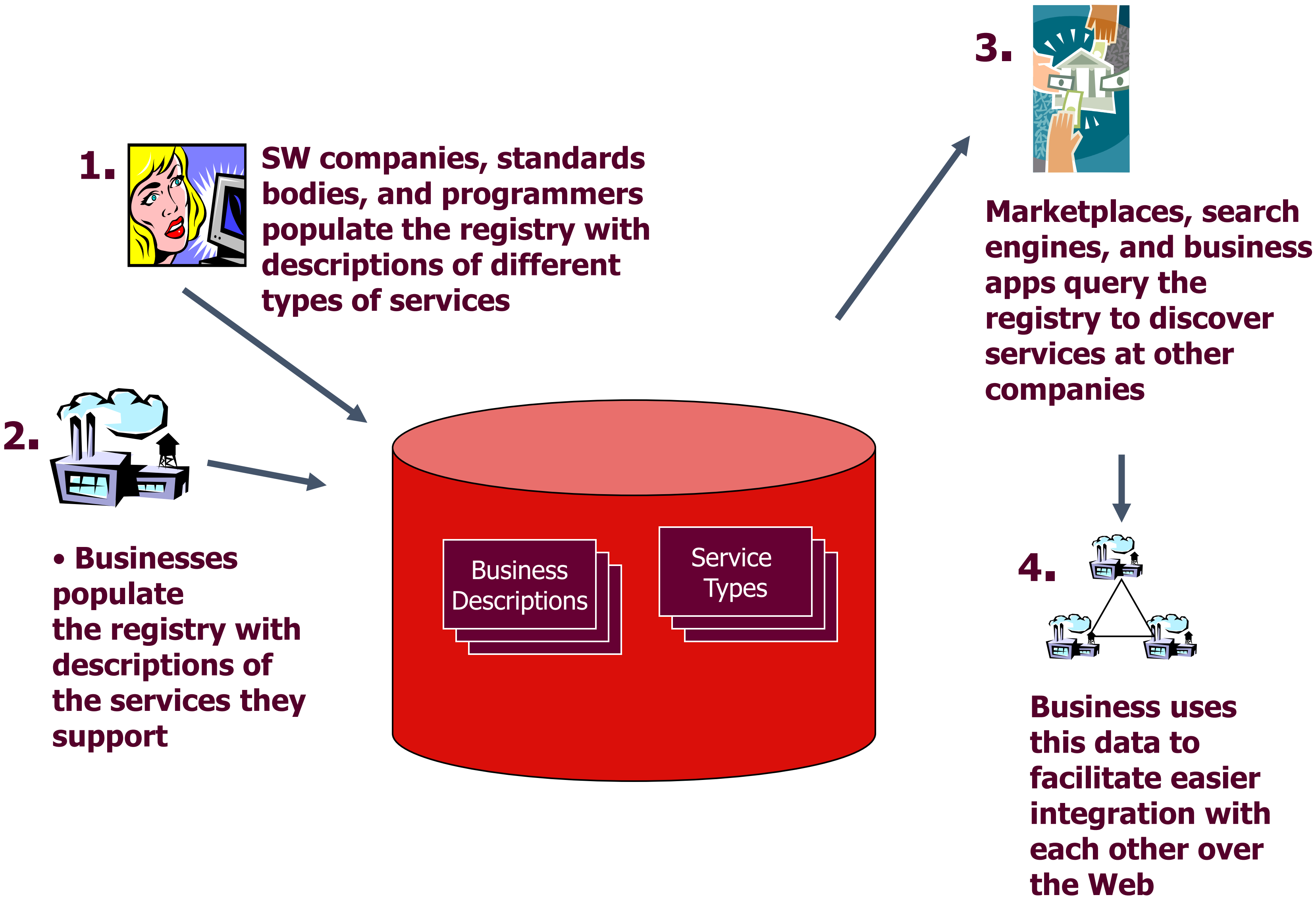
WSDL Web Service Definition Language

- WSDL is an XML-based language used to define Web Services and describe how to access them.
- It is the external interface for a client (IDL)
- WSDL includes information about
  - Data types it uses
  - Parameters it requires and returns
  - Groupings of functionality
  - The protocol to be used to access the service
  - The location or address of the service

WSDL Structure

- <definition> - Root element
- <types> - Provides data type definitions
- <message> - Represents the abstract definition of the data being transmitted
- <portType> - Defines a set of abstract operations
- <binding> - Specifies concrete protocol and data format specifications for the operations and messages defined by a particular portType
- <port> - Specifies an address for a binding
- **<**service> - Used to aggregate a set of related ports
- <serviceType> - Mechanism to aggregate portTypes

UDDI Universal Description Discovery and Integration



1. **SW companies, standards bodies, and programmers populate the registry with descriptions of different types of services**

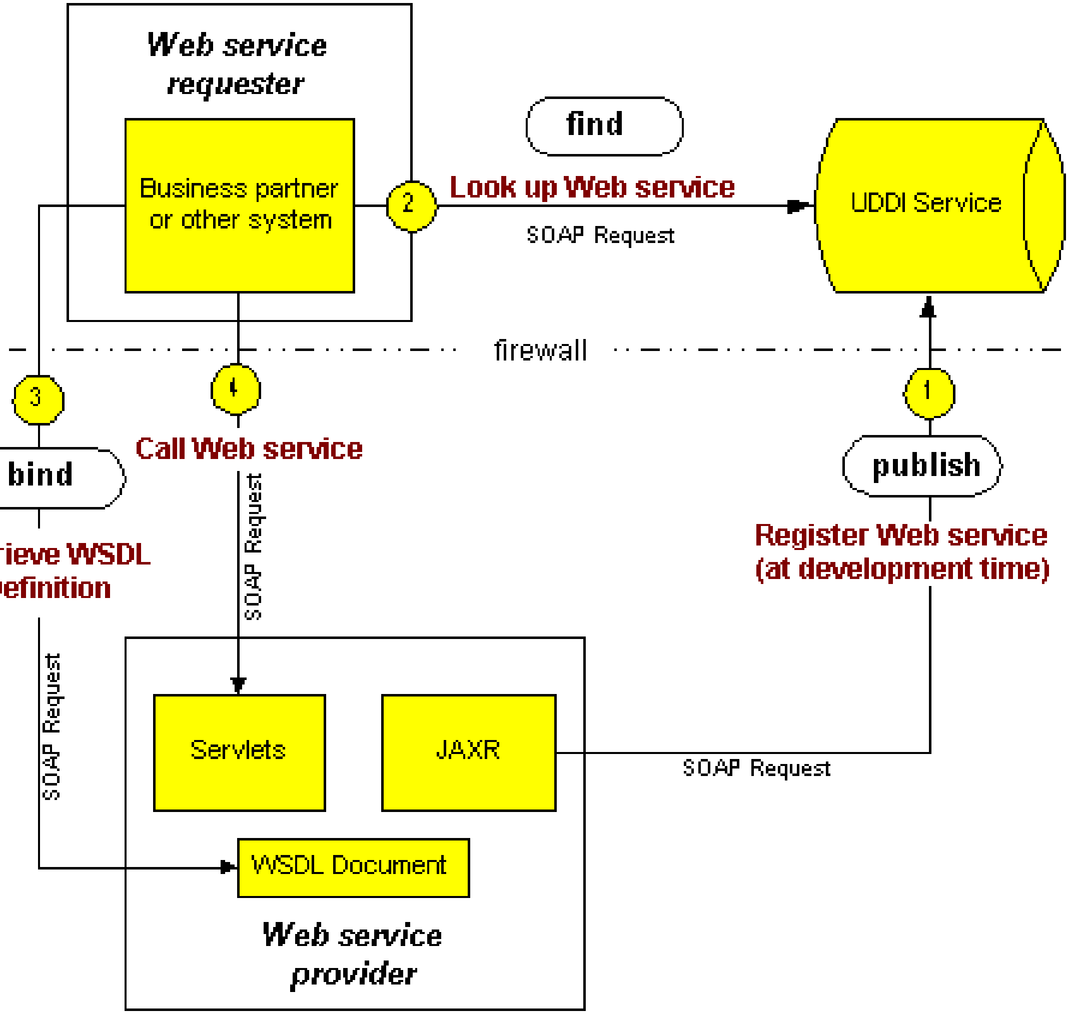
2.

- **Businesses populate the registry with descriptions of the services they support**

3. **Marketplaces, search engines, and business apps query the registry to discover services at other companies**

4. **Business uses this data to facilitate easier integration with each other over the Web**

How it works all together



Anatomy of a SOAP message

```
<?xml version=‘1.0’ encoding=‘UTF-8’?>

<SOAP-ENV:Envelope xmlns:SOAP_ENV=“http://schemas.xmlsoap.org/soap/envelope/”
        xmlns:xsi=“http://www.w3c.org/1999/XMLSchema-instance”
        xmlns:xsd=“http://www.w3c.org/1999/XMLSchema”>
        <SOAP-ENV:Header>

        </SOAP-ENV:Header


        <SOAP_ENV:Body>


        </SOAP-ENV:Body>


</SOAP-ENV:Envelope>
```

SOAP protocol binding

```
SOAPAction = “urn:soaphttpclient-action-uri”

Host = localhost

Content-Type = text/xml; charset=utf-8

Content-Length = 701
```

```
<SOAP-ENV:Envelope xmlns:SOAP_ENV=“http://schemas.xmlsoap.org/soap/envelope/”
        xmlns:xsi=“http://www.w3c.org/1999/XMLSchema-instance”
        xmlns:xsd=“http://www.w3c.org/1999/XMLSchema”>




</SOAP-ENV:Envelope>
```

A SOAP fault

```
<?xml version=‘1.0’ encoding=‘UTF-8’?>
<SOAP-ENV:Envelope xmlns:SOAP_ENV=“http://schemas.xmlsoap.org/soap/envelope/”
   xmlns:xsi=“http://www.w3c.org/1999/XMLSchema-instance”
   xmlns:xsd=“http://www.w3c.org/1999/XMLSchema”>
   <SOAP_ENV:Body>        <SOAP-ENV:Fault>
                               <faultcode>SOAP-ENV:Server</faultcode>
                               <faultstring>Test fault</faultstring>
                               <faultactor>/soap/servlet/rpcrouter</faultactor>
                               <detail>
                                            ..
                               </detail>
                          </SOAP-ENV:Fault>



   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Attaching a file to a SOAP message

- To add a file to a SOAP message a tag is added within the body of the message.

```
<?xml version=‘1.0’ encoding=‘UTF-8’?>
<SOAP-ENV:Envelope xmlns:SOAP_ENV=“http://schemas.xmlsoap.org/soap/envelope/”
          xmlns:xsi=“http://www.w3c.org/1999/XMLSchema-instance”
          xmlns:xsd=“http://www.w3c.org/1999/XMLSchema”>
          <SOAP_ENV:Body>


                    <attachment href=“{URL}”/>


          </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

REST

- Representational state transfer (REST) is a way of getting information content from a web site by reading a designated web page that contains an XML file that describes and includes the desired content.
- For instance, REST could be used by your cloud provider to provide updated subscription information.
- Every so often, the provider could prepare a web page that includes content and XML statements that are described in the code.
- Subscribers only need to know the uniform resource locator (URL) for the page where the XML file is located, read it with a web browser, understand the content using XML information, and display it appropriately.

Benefits REST offers the following benefits

- It gives better response time and reduced server load due to its support for the caching of representations.
- Server scalability is improved by reducing the need to maintain session state.
- A single browser can access any application and any resource, so less client-side software needs to be written.
- A separate resource discovery mechanism is not needed, due to the use of hyperlinks in representations.
- Better long-term compatibility and evolvability characteristics exist than in RPC.

This is due to:

- The ability of documents, like HTML, to evolve with both forward- and backward-compatibility.
- Resources can add support for new content types as they are defined, without eliminating support for older content types.