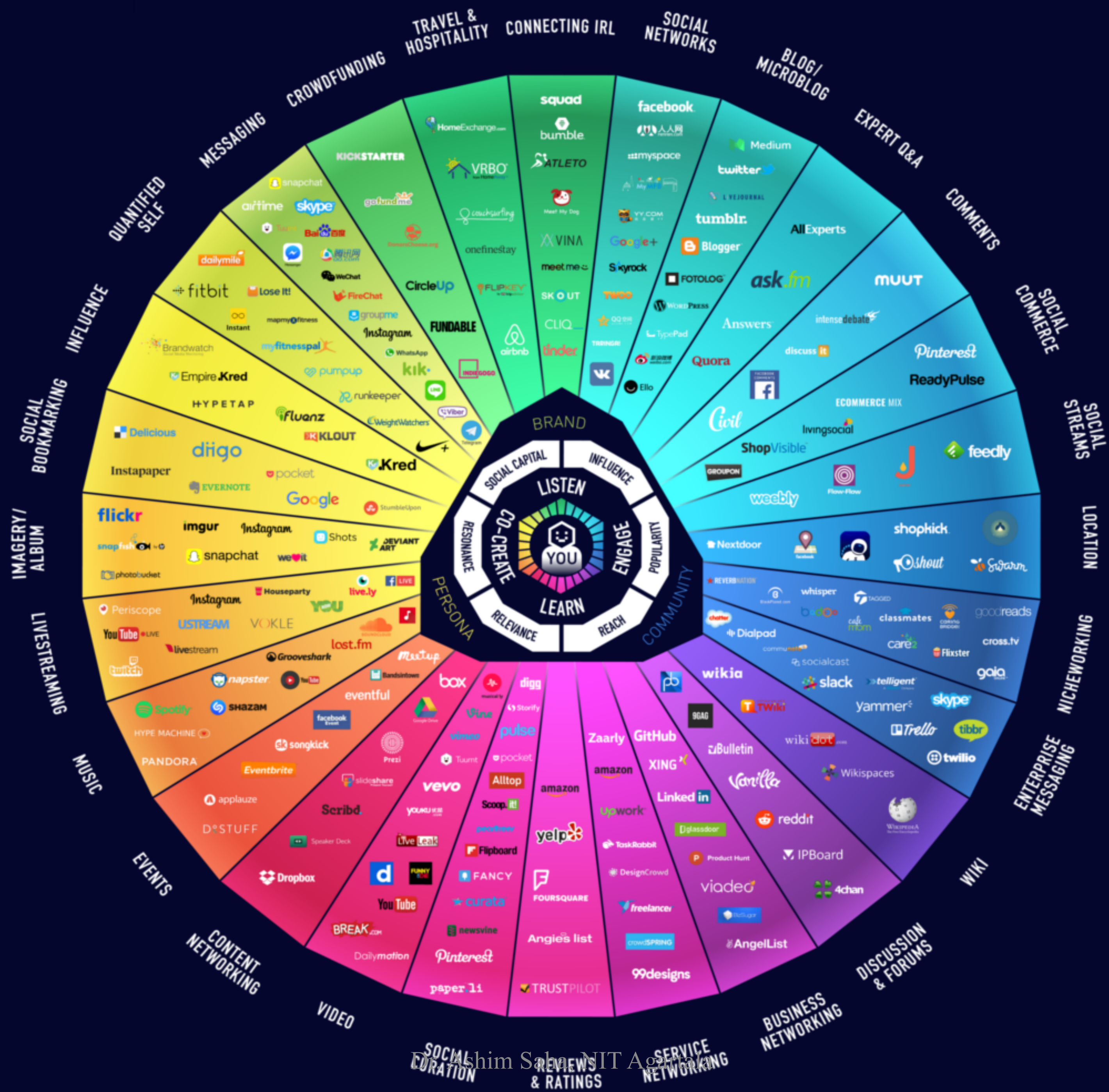
Web Technology

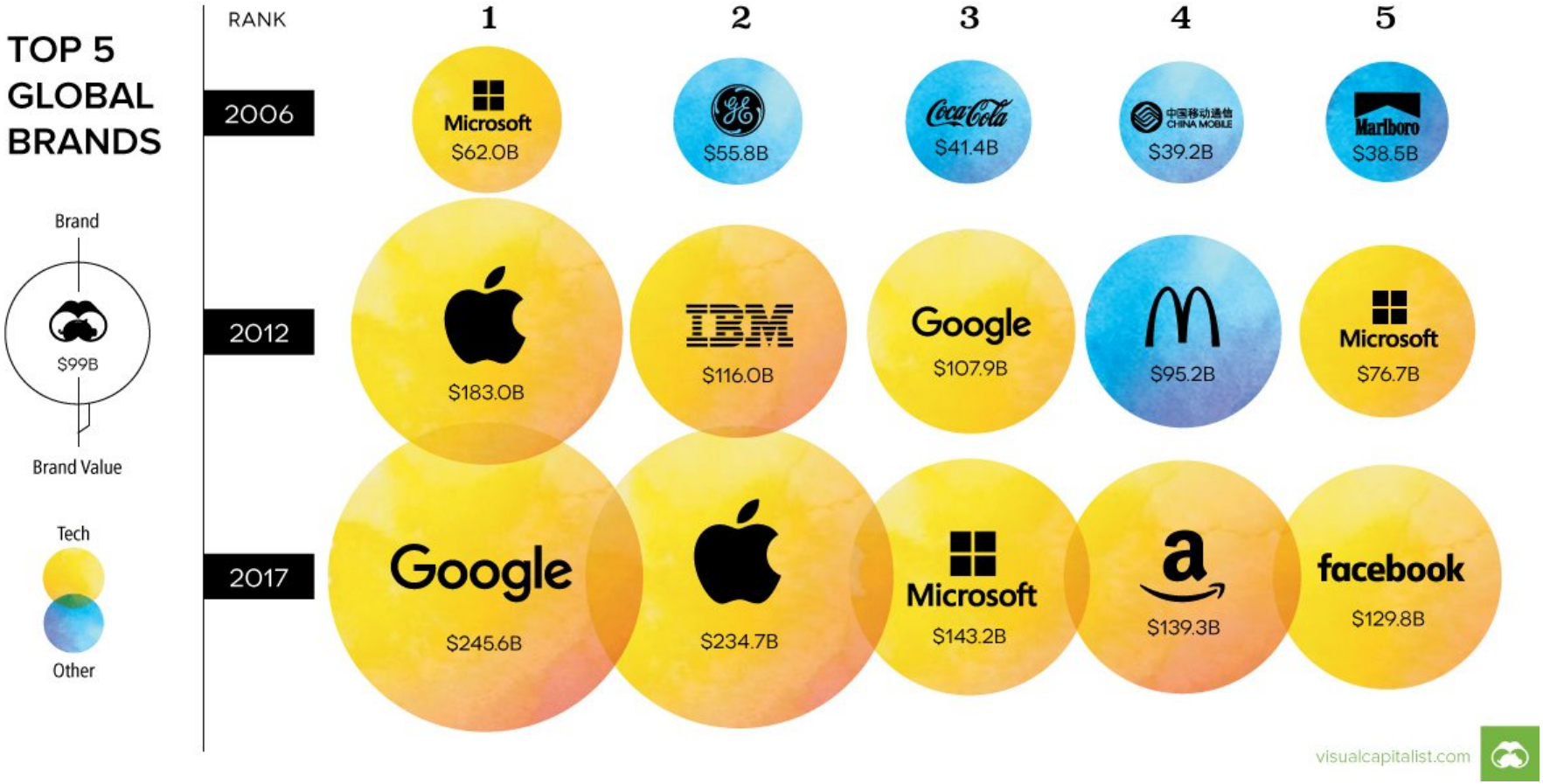
Chapter1: Introduction

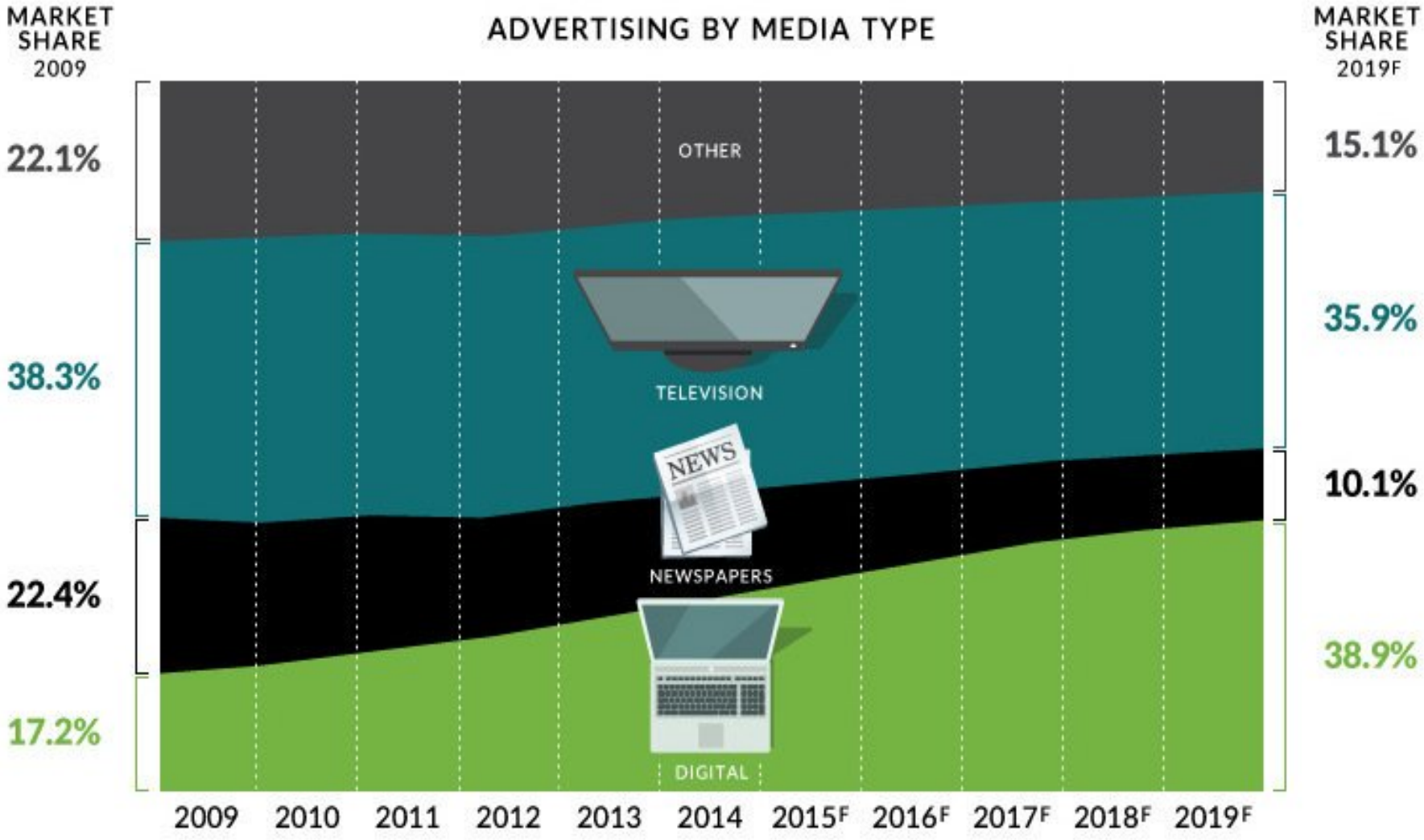
Dr. Ashim Saha

Asst. Prof, CSE Dept.
NIT Agartala


2017 This Is What Happens In An Internet Minute
facebook
900,000 Logins
16 Million Text Messages
YouTube
4.1 Million Videos Viewed
342,000 Apps Downloaded
46,200 Posts Uploaded
Instagram
452,000 Tweets Sent
990,000 Swipes
tinder
156 Million Emails Sent
40,000 Hours Listened
Spotify
50 Voice-First Devices Shipped
amazon echo
120 New Accounts Created
LinkedIn
15,000 GIFs Sent via Messenger
1.8 Million Snaps Created
$751,522 Spent Online
NETFLIX
70,017 Hours Watched
Google
3.5 Million Search Queries
60 SECONDS
Created By:
@LoriLewis
@OfficiallyChadd


TRAVEL & HOSPITALITY
CONNECTING IRL
SOCIAL NETWORKS
BLOG/ MICROBLOG
EXPERT Q&A
COMMENTS
SOCIAL COMMERCE
SOCIAL STREAMS
LOCATION
NICHEWORKING
ENTERPRISE MESSAGING
WIKI
DISCUSSION & FORUMS
BUSINESS NETWORKING
SERVICE NETWORKING
REVIEWS & RATINGS
SOCIAL CURATION
VIDEO
CONTENT NETWORKING
EVENTS
MUSIC
LIVESTREAMING
IMAGERY/ ALBUM
SOCIAL BOOKMARKING
INFLUENCE
QUANTIFIED SELF
MESSAGING
CROWDFUNDING
BRAND
COMMUNITY
PERSONA
SOCIAL CAPITAL
INFLUENCE
POPULARITY
REACH
RELEVANCE
RESONANCE
LISTEN
ENGAGE
LEARN
CO-CREATE
YOU

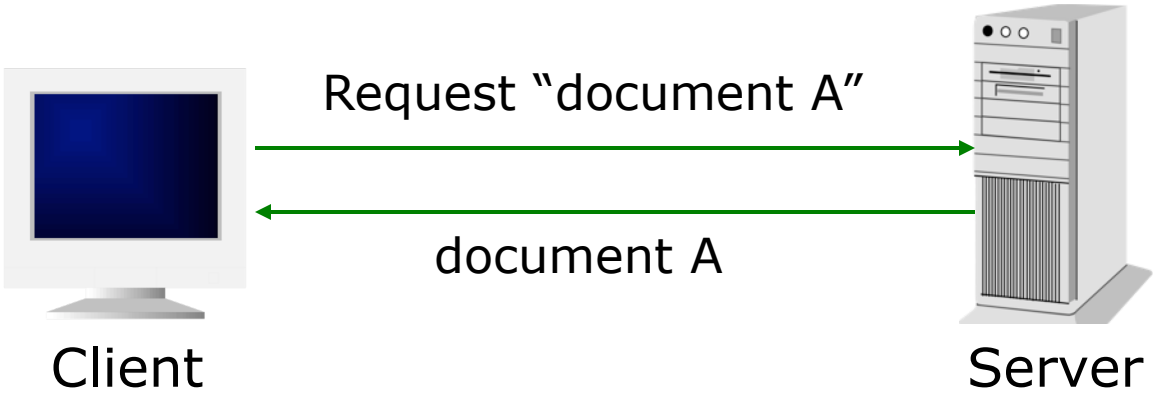

TOP 5 GLOBAL BRANDS
RANK
1
2
3
4
5
2006
Microsoft $62.0B
$55.8B
$41.4B
$39.2B
Marlboro $38.5B
2012
$183.0B
IBM $116.0B
Google $107.9B
$95.2B
Microsoft $76.7B
2017
Google $245.6B
$234.7B
Microsoft $143.2B
$139.3B
facebook $129.8B
Brand
$99B
Brand Value
Tech
Other
visualcapitalist.com


MARKET SHARE 2009
ADVERTISING BY MEDIA TYPE
MARKET SHARE 2019F
22.1%
OTHER
15.1%
38.3%
TELEVISION
35.9%
NEWS
10.1%
NEWSPAPERS
22.4%
38.9%
17.2%
DIGITAL
2009
2010
2011
2012
2013
2014
2015F
2016F
2017F
2018F
2019F


Language Framework
Microsoft .net
php
Java
Product Suits
FusionCharts
Visual Studio
Database
Joomla
SQL Server 2008
ORACLE DATABASE
MySQL

Web Essentials

- **Client**: web browsers, used to surf the Web
- **Server** systems: used to supply information to these browsers
- **Computer Networks**: used to support the browser-server communication



Internet v.s. Web

- **The Internet**: a inter-connected computer networks, linked by wires, cables, wireless connections, etc.
- **Web**: a collection of interconnected documents and other resources.
- The world wide web (**WWW**) is accessible via the Internet, as are many other services including email, file sharing, etc.

How does the Internet Work?

- Through communication protocols
- A **communication protocol** is a specification of how communication between two computers will be carried out.
  - **IP** (Internet Protocol): defines the packets that carry blocks of data from one node to another.
  - **TCP** (Transmission Control Protocol) and **UDP** (User Datagram Protocol): the protocols by which one host sends data to another.
  - Other application protocols: **DNS** (Domain Name Service), **SMTP** (Simple Mail Transmission Protocol), and **FTP** (File Transmission Protocol)

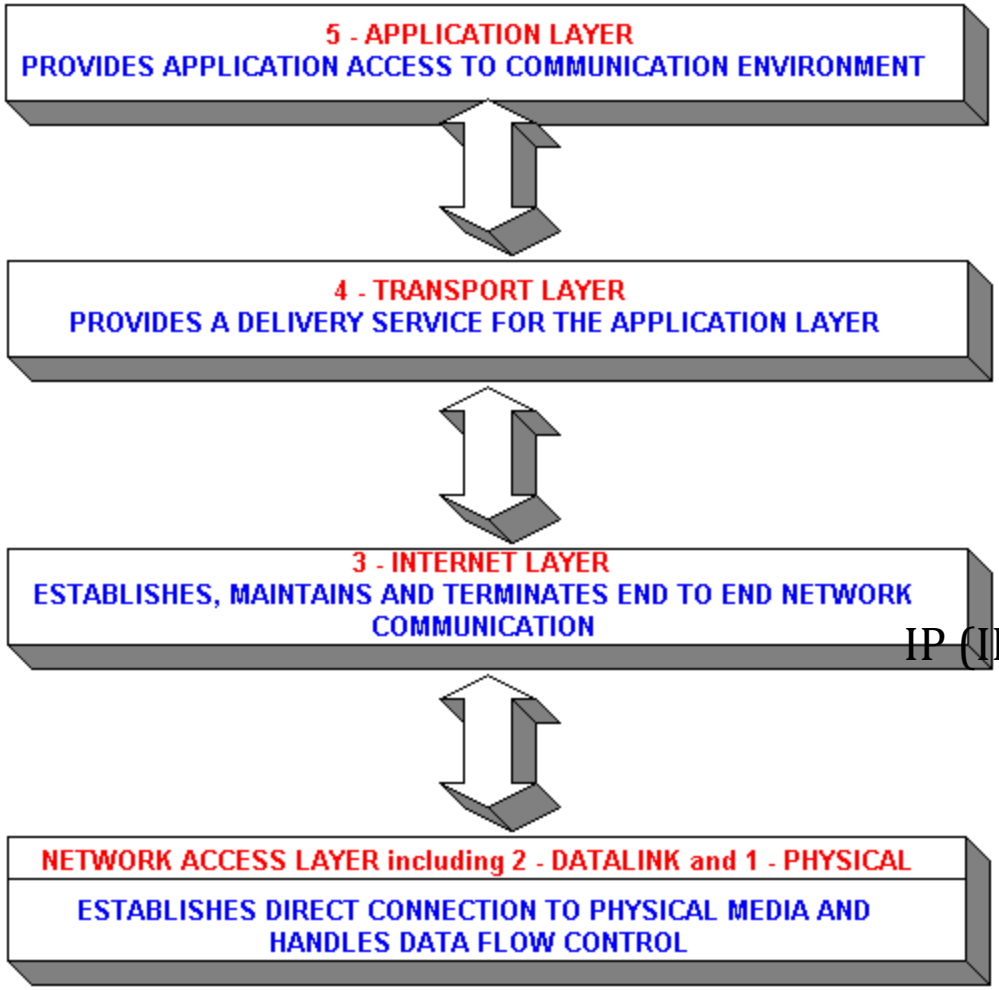
The Internet Protocol (IP)

- A key element of IP is IP address, a 32-bit number.
- The Internet authorities assign ranges of numbers to different organizations.
- IP is responsible for moving packet of data from node to node.
- A packet contains information such as the data to be transferred, the source and destination IP addresses, etc.
- Packets are sent through different local network through gateways.
- A checksum is created to ensure the correctness of the data; corrupted packets are discarded
- IP-based communication is unreliable.

Transmission Control Protocol (TCP)

- TCP is a higher-level protocol that extends IP to provide additional functionality: reliable communication
- TCP adds support to detect errors or lost data and to trigger retransmission until the data is correctly and completely received.
- Connection.
- Acknowledgment

TCP/IP Protocol Suites



HTTP, FTP, Telnet, DNS, SMTP, etc.

TCP, UDP

IP (IPv4, IPv6)

Application Layer Protocols

An application layer protocol defines how an application processes (clients and servers) , running on different end systems, pass messages to each other. In particular, an application layer protocol defines:

- The types of messages, e.g., request messages and response messages
- The syntax of the various message types, i.e., the fields in the message and how the fields are delineated.
- The semantics of the fields, i.e., the meaning of the information that the field is supposed to contain;
- Rules for determining when and how a process sends messages and responds to messages.

Application Type	Application-layer protocol	Transport Protocol
Electronic mail	Send: Simple Mail Transfer Protocol SMTP [RFC 821]	TCP 25
	Receive: Post Office Protocol v3 POP3 [RCF 1939]	TCP 110
Remote terminal access	Telnet [RFC 854]	TCP 23
World Wide Web (WWW)	HyperText Transfer Protocol 1.1 HTTP 1.1 [RFC 2068]	TCP 80
File Transfer	File Transfer Protocol FTP [RFC 959]	TCP 21
	Trivial File Transfer Protocol TFTP [RFC 1350]	UDP 69
Remote file server	NFS [McKusik 1996]	UDP or TCP
Streaming multimedia	Proprietary (e.g., Real Networks)	UDP or TCP
Internet telephony	Proprietary (e.g., Vocaltec)	Usually UDP

SMTP

SMTP (Simple Mail Transfer Protocol):

- One of the most popular network service is electronic mail (e-mail).
- The TCP/IP protocol that supports electronic mail on the Internet is called Simple Mail Transfer Protocol (SMTP).
- SMTP transfers messages from senders' mail servers to the recipients' mail servers using TCP connections.
- Users based on e-mail addresses.
- SMTP provides services for mail exchange between users on the same or different computers.

Multipurpose Internet Mail Extensions (MIME):

- It is an extension of SMTP that allows the transfer of multimedia messages.
- If binary data is included in a message MIME headers are used to inform the receiving mail agent:
- Content-Transfer-Encoding: Header alerts the receiving user agent that the message body has been ASCII encoded and the type of encoding used.
- Content-Type: Header informs the receiving mail agent about the type of data included in the message.

POP (Post Office Protocol)

- POP is also called as POP3 protocol.
- This is a protocol used by a mail server in conjunction with SMTP to receive and holds mail for hosts.
- POP3 mail server receives e-mails and filters them into the appropriate user folders. When a user connects to the mail server to retrieve his mail, the messages are downloaded from mail server to the user's hard disk.

The Internet Mail Access Protocol (IMAP)

- POP3 is a very good and simple protocol for retrieving messages to your UA. However, its simplicity results in a lack of several desired features.
- For instance, POP3 only works in offline mode, meaning that the messages are downloaded to the UA and deleted from the server.
- IMAP is a standard email protocol that stores email messages on a mail server, but allows the end user to view and manipulate the messages as though they were stored locally on the end user's computing device(s).

POP3 Versus IMAP4

The advantages of POP3 are

- It is very simple.
- It is widely supported.

IMAP4 has several distinct advantages:

- Stronger authentication
- Support for multiple mailboxes
- Greater support for online, offline, or disconnected modes of operation

POP3 Versus IMAP4

- IMAP4's online mode support allows UAs to download only a subset of the messages from the server, search for and download only messages matching a certain criteria, and so on.
- IMAP4 also allows a user or UA to move messages between server folders and delete certain messages.
- IMAP4 is much better suited for the mobile user who needs to work at several different computers, or the user who needs to access and maintain several different mailboxes.

The World Wide Web (WWW)

- **WWW** is a system of interlinked, hypertext documents that runs over the Internet.
- Two types of software:
  - **Client**: a system that wishes to access the information provided by servers must run client software (e.g., web browser)
  - **Server**: an internet-connected computer that wishes to provide information to others must run server software (e.g. XAMPP, Apache, IIS etc.).
  - Client and server applications communicate over the Internet by following a protocol built on top of TCP/IP – Hyper Text Transport Protocol (**HTTP**)

Basics of the WWW

- **Hypertext**: a format of information which allows one to move from one part of a document to another or from one document to another through **hyperlinks.**
- **Uniform Resource Locator** (**URL**): unique identifiers used to locate a particular resource on the network.
- **Markup language**: defines the structure and content of hypertext documents.

Web Client: Browser

- Makes HTTP requests on behalf of the user
  - Reformat the URL entered as a valid HTTP request
  - Use DNS to convert server's host name to appropriate IP address
  - Establish a TCP connection using the IP address
  - Send HTTP request over the connection and wait for server's response
  - Display the document contained in the response
    - If the document is not a plain-text document but instead is written in HTML, this involves rendering the document (positioning text, graphics, creating table borders, using appropriate fonts, etc.)

Web Servers

- Main functionalities:
  - Server waits for connect requests
  - When a connection request is received, the server creates a new process to handle this connection
  - The new process establishes the TCP connection and waits for HTTP requests
  - The new process invokes software that maps the requested URL to a resource on the server
  - If the resource is a file, creates an HTTP response that contains the file in the body of the response message
  - If the resource is a program, runs the program, and returns the output

Static Web: HTML/XHTML, CSS

- **HTML** stands for **H**yper **T**ext **M**arkup **L**anguage
  - It is a text file containing small markup tags that tell the Web browser how to display the page

- **XHTML** stands for e**X**tensible **H**yper**T**ext **M**arkup **L**anguage
  - It is identical to HTML 4.01
  - It is a stricter and cleaner version of HTML

- **CSS** stands for **C**ascading **S**tyle **S**heets
  - It defines how to display HTML elements

Why Programmability?

- What's the drawback to simple document model?
  - Static
  - Assume that documents are created before they are requested

- What are examples of information that might be part of web documents that may not be known before they are requested?

CGI

- Common Gateway Interface:
  - CGI provides a way by which a web server can obtain data from (or send data to) database, and other programs, and present that data to viewers via the web.
  - A CGI program can be written in any programming language.

Server-Side Programmability

- The requests cause the response to be generated

- Server scripting:
  - **CGI/Perl**: Common Gate Way Interface (*.pl, *.cgi)
  - **PHP**: Open source, strong database support (*.php)
  - **ASP**: Microsoft product, uses .Net framework (*.asp)
  - **Java** via JavaServer Pages (*.jsp)

Client-Side Programmability

- Scripting language: a lightweight programming language

- Browser scripting:
  - Designed to add interactivity to HTML pages
  - Usually embedded into HTML pages
  - Browsers must support the used scripting language.
  - Browsers are disabled for scripting to prevent the risk of misuse.

Client-Side Programmability

- **JavaScript :**
  - Put dynamic text into an HTML page
  - React to events
  - A JavaScript consists of lines of executable computer code
  - JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
  - Read and write HTML elements
  - Validate data before it is submitted to a server
  - Create cookies

Client-Side Programmability

- **Are Java and JavaScript the Same?**
  - NO!
  - Java and JavaScript are two completely different languages in both concept and design!
  - Java (developed by Sun Microsystems) is a powerful and much more complex programming language - in the same category as C and C++.

Client-Side Programmability

- **JavaScript Example?**

```
<html>
<body>
  <script type="text/javascript">
    document.write("Hello World!")
  </script>
</body>
</html>
```

Client-Side Programmability

- **Where to put a JavaScript?**
  - Head section
  - Body section
  - External scripts

Client-Side Programmability

- **JavaScript on head section?**

```
<html>
<head>
  <script type="text/javascript">
    function message()
      {
         alert("This alert box was called with the onload event")
      }
  </script>
</head>

<body onload="message()">
</body>
</html>
```

Client-Side Programmability

- **JavaScript on body section?**

```
<html>
<head>
</head>
<body>
  <script type="text/javascript">
        document.write("Hello World!")
   </script>
</body>
```

Client-Side Programmability

- **JavaScript from External sources?**

```
<html>
<head>
  <script src="xxx.js">
  </script>
</head>
<body>
</body>
</html>
```

Client-Side Programmability

- **JavaScript:**

```
<html>
<body>
  <script type="text/javascript">
    var d = new Date()
    var time = d.getHours()

     if (time < 12)
       {
         document.write("<b>Good morning</b>")
       }
      else
       {
         document.write("<b>Good day</b>")
       }
  </script>
```

Client-Side Programmability

- **VBScript?**
  - VBScript is a Microsoft scripting language.
  - A scripting language is a lightweight programming language.
  - VBScript is a light version of Microsoft's programming language Visual Basic.

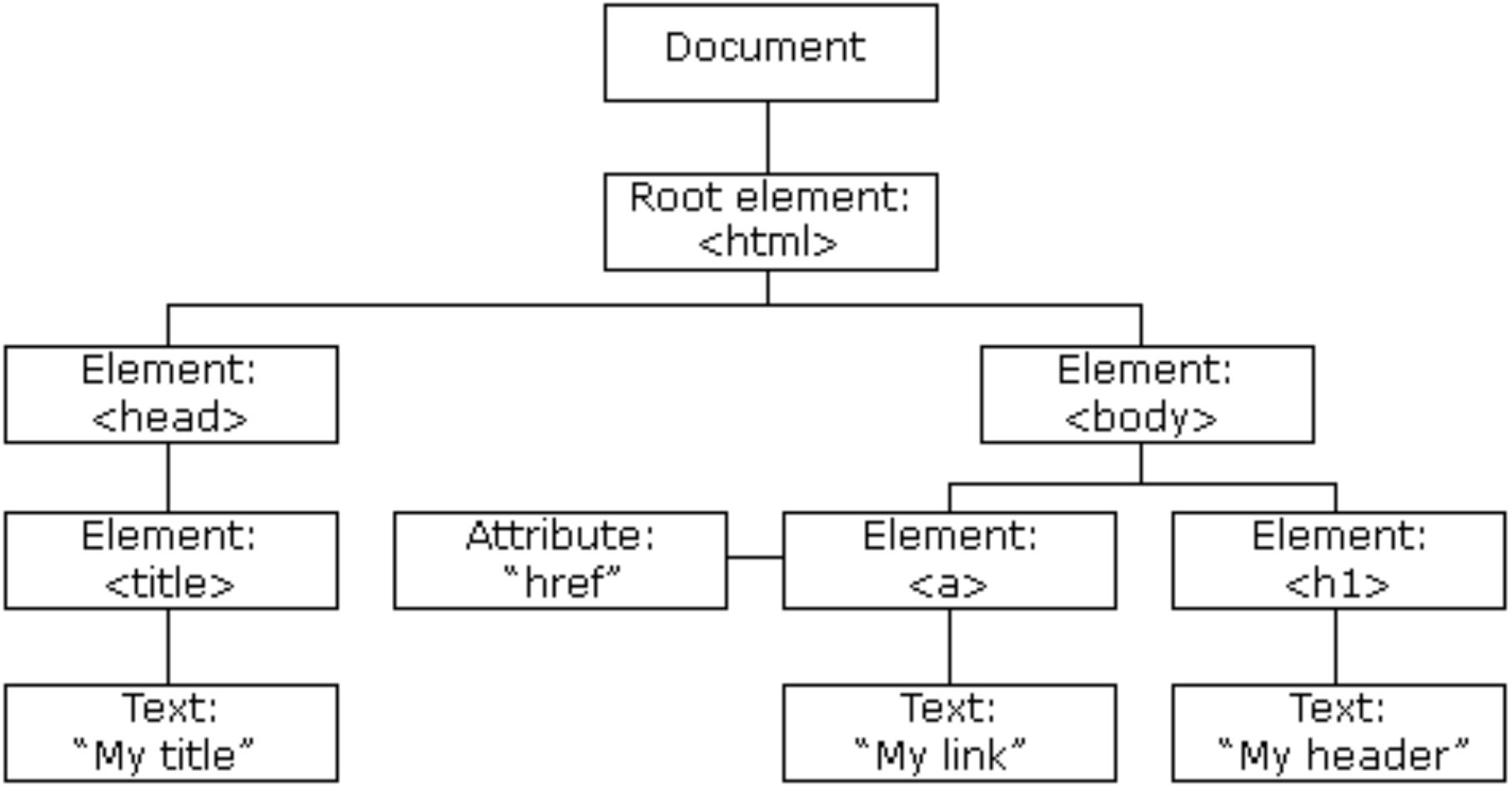
DOM

- **DOM?**
  - The HTML Document Object Model (HTML DOM) defines a standard way for accessing and manipulating HTML documents.
  - The DOM presents an HTML document as a tree-structure (a node tree), with elements, attributes, and text.
  - With JavaScript you can restructure an entire HTML document. You can add, remove, change, or reorder items on a page.

DOM

- **DOM?**
  - This access, along with methods and properties to add, move, change, or remove HTML elements, is given through the Document Object Model (DOM).
  - All browsers have implemented this recommendation, and therefore, incompatibility problems in the DOM have almost disappeared.
  - The DOM can be used by JavaScript to read and change HTML, XHTML, and XML documents.

Document Tree



Session

- HTTP is a stateless protocol. All requests and responses are independent.
- But sometimes you need to keep track of client's activity across multiple requests.
- For eg. When a User logs into your website, not matter on which web page he visits after logging in, his credentials will be with the server, until he logs out. So this is managed by creating a session.

Session Management

- Session Management is a mechanism used by the Web container to store session information for a particular user.
- There are four different techniques used by Servlet application for session management. They are as follows:
  - Cookies
  - Hidden form field
  - URL Rewriting
  - HttpSession

Session Management (Cookie)

- Cookies are small pieces of information that are sent in response from the web server to the client. Cookies are the simplest technique used for storing client state.
- Cookies are stored on client's computer. They have a lifespan and are destroyed by the client browser at the end of that lifespan.

Session Management (Cookie)

- **Creating a new Cookie**

```
Cookie ck = new Cookie("username",name);
```

creating a new cookie object

- **Setting up lifespan for a cookie**

```
ck.setMaxAge(30*60);
```

setting maximum age of cookie

- **Sending the cookie to the client**

```
response.addCookie(ck);
```

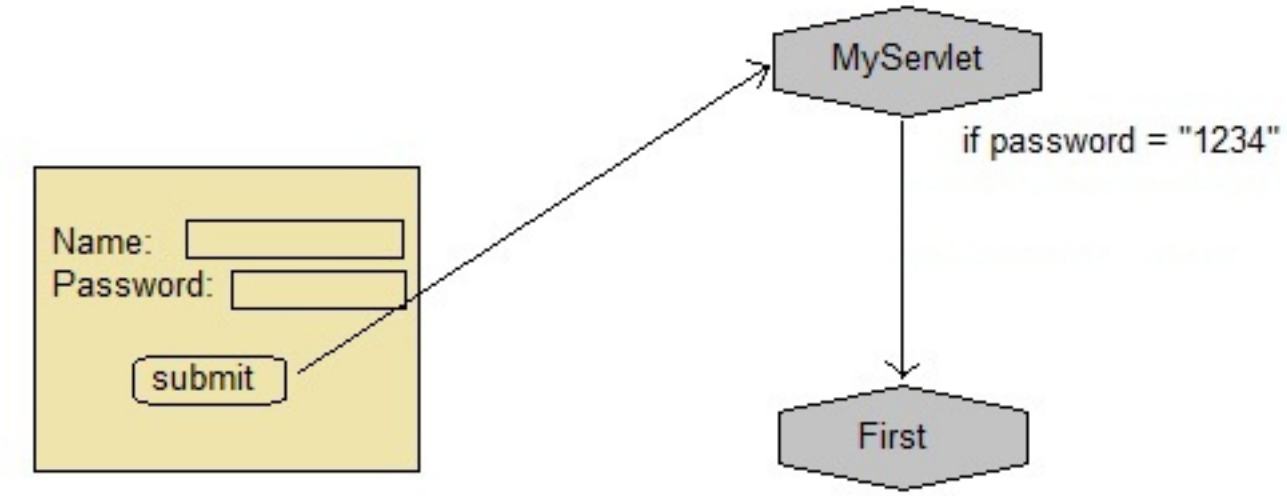
adding cookie to **response** object

- **Getting cookies from client request**

```
Cookie[] cks = request.getCookies();
```

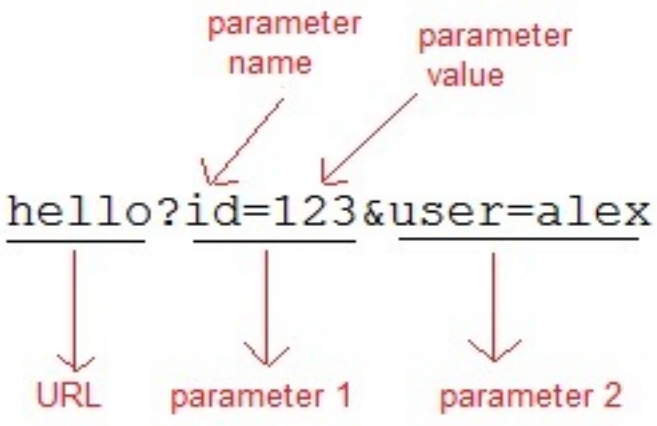
getting the cookie for **request** object

Session Management (Cookie)



Session Management (URL rewriting)

- If the client has disabled cookies in the browser then session management using cookie wont work. In that case URL Rewriting can be used as a backup. URL rewriting will always work.
- In URL rewriting, a token(parameter) is added at the end of the URL. The token consist of name/value pair separated by an equal(=) sign.



Session Management (Using Hidden Form Field)

- Hidden form field can also be used to store session information for a particular client. In case of hidden form field a hidden field is used to store client state. In this case user information is stored in hidden field value and retrieved from another servlet.

- **Advantages** :Does not have to depend on browser whether the cookie is disabled or not.
- Inserting a simple HTML Input field of type hidden is required. Hence, its easier to implement.

Session Management (HttpSession)

- **HttpSession** object is used to store entire session with a specific client.
- We can store, retrieve and remove attribute from HttpSession object. Any servlet can have access to HttpSession object throughout the getSession() method of the HttpServletRequest object.

What's Ahead?

- HTML, XHTML
- CSS
- Simple client-side interactivity (JavaScript)
- Simple server-side interactivity (CGI/Perl)
- Overview of the basics, and learn how to use the web resources to help build a web page

XML

XML is not…

- **A replacement for HTML**
  (but HTML can be generated from XML)
- **A presentation format**
  (but XML can be converted into one)
- **A programming language**
  (but it can be used with almost any language)
- **A network transfer protocol**
  (but XML may be transferred over a network)
- **A database**
  (but XML may be stored into a database)

But then – what is it?

XML is a meta markup language for text documents / textual data

XML allows to define languages („applications“) to represent text documents / textual data

XML by Example

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in 10 Years</title>
</article>
```

- Easy to understand for human users
- Very expressive (semantics along with the data)
- Well structured, easy to read and write from programs

This looks nice, but…

XML by Example

… this is XML, too:

```
<t108>
  <x87>Gerhard Weikum</x87>
  <g10>The Web in 10 Years</g10>
</t108>
```

- Hard to understand for human users
- Not expressive (no semantics along with the data)
- Well structured, easy to read and write from programs

XML by Example

… and what about this XML document:

```
<data>
  ch37fhgks73j5mv9d63h5mgfkds8d984lgnsmcns983
</data>
```

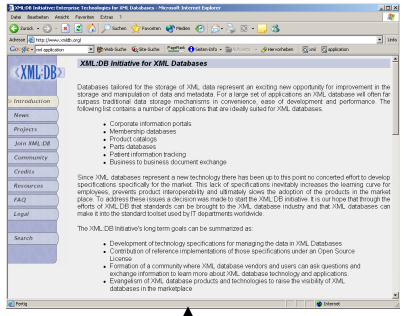
- Impossible to understand for human users
- Not expressive (no semantics along with the data)
- Unstructured, read and write only with special programs

The actual benefit of using XML highly depends on the design of the application.

Possible Advantages of Using XML

- Truly Portable Data
- Easily readable by human users
- Very expressive (semantics near data)
- Very flexible and customizable (no finite tag set)
- Easy to use from programs (libs available)
- Easy to convert into other representations (XML transformation languages)
- Many additional standards and tools
- Widely used and supported

App. Scenario 1: Content Mgt.



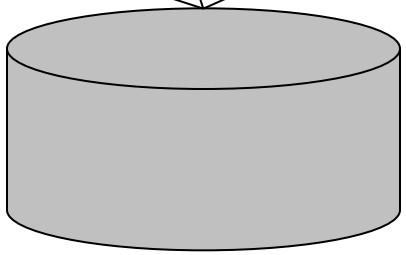
Clients

XML2HTML

XML2WML

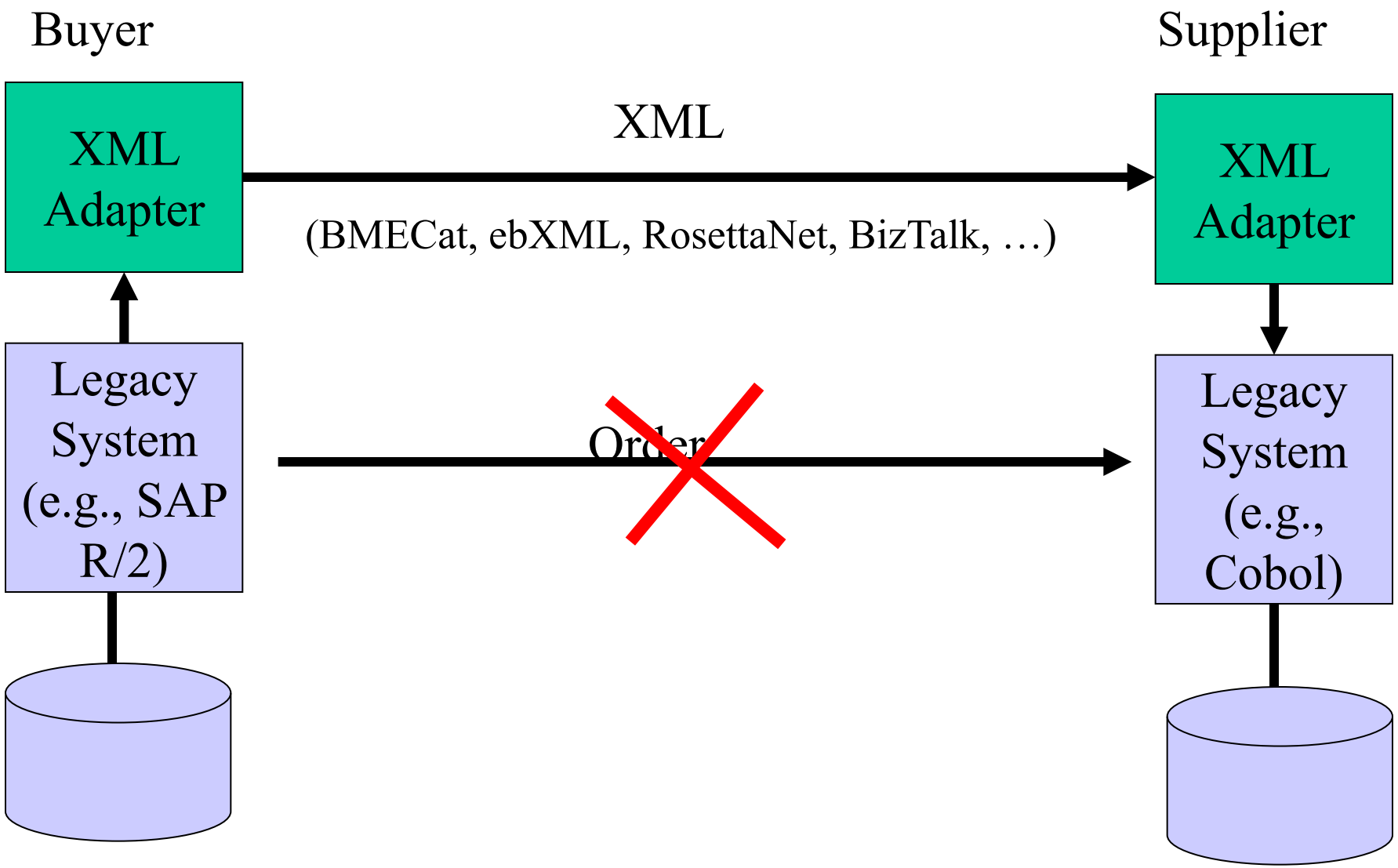
XML2PDF

Converters

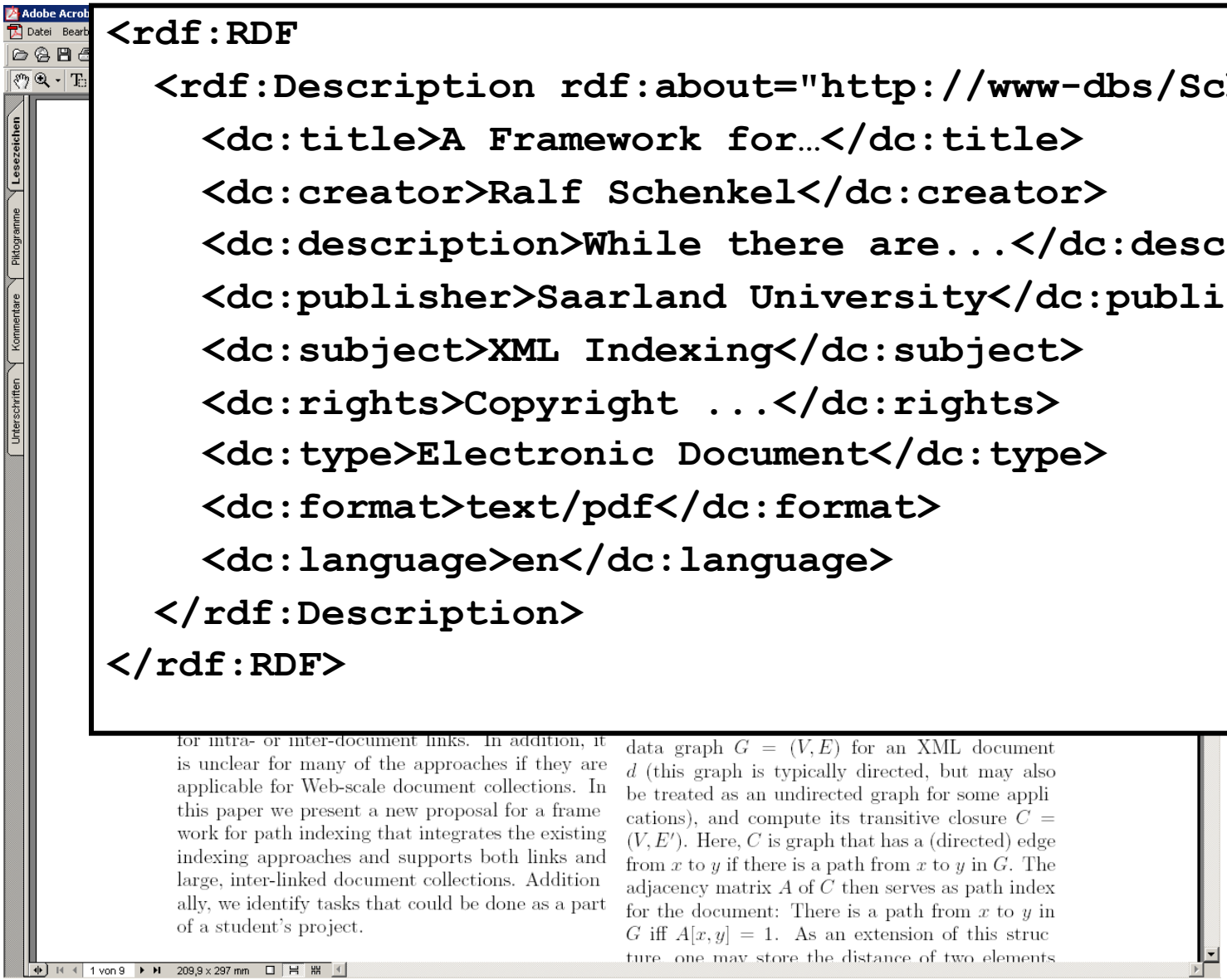


Database with
XML documents

App. Scenario 2: Data Exchange



App. Scenario 3: XML for Metadata



```
<rdf:RDF
  <rdf:Description rdf:about="http://www-dbs/Sch03.pdf">
    <dc:title>A Framework for…</dc:title>
    <dc:creator>Ralf Schenkel</dc:creator>
    <dc:description>While there are...</dc:description>
    <dc:publisher>Saarland University</dc:publisher>
    <dc:subject>XML Indexing</dc:subject>
    <dc:rights>Copyright ...</dc:rights>
    <dc:type>Electronic Document</dc:type>
    <dc:format>text/pdf</dc:format>
    <dc:language>en</dc:language>
  </rdf:Description>
</rdf:RDF>
```

App. Scenario 4: Document Markup

```
<article>
  <section id=„1“ title=„Intro“>
    This article is about <index>XML</index>.
  </section>
  <section id=„2“ title=„Main Results“>
    <name>Weikum</name> <cite idref=„Weik01“/> shows
   the following theorem (see Section <ref idref=„1“/>)
    <theorem id=„theo:1“ source=„Weik01“>
      For any XML document x, ...
    </theorem>
  </section>
  <literature>
    <cite id=„Weik01“><author>Weikum</author></cite>
  </literature>
</article>
```

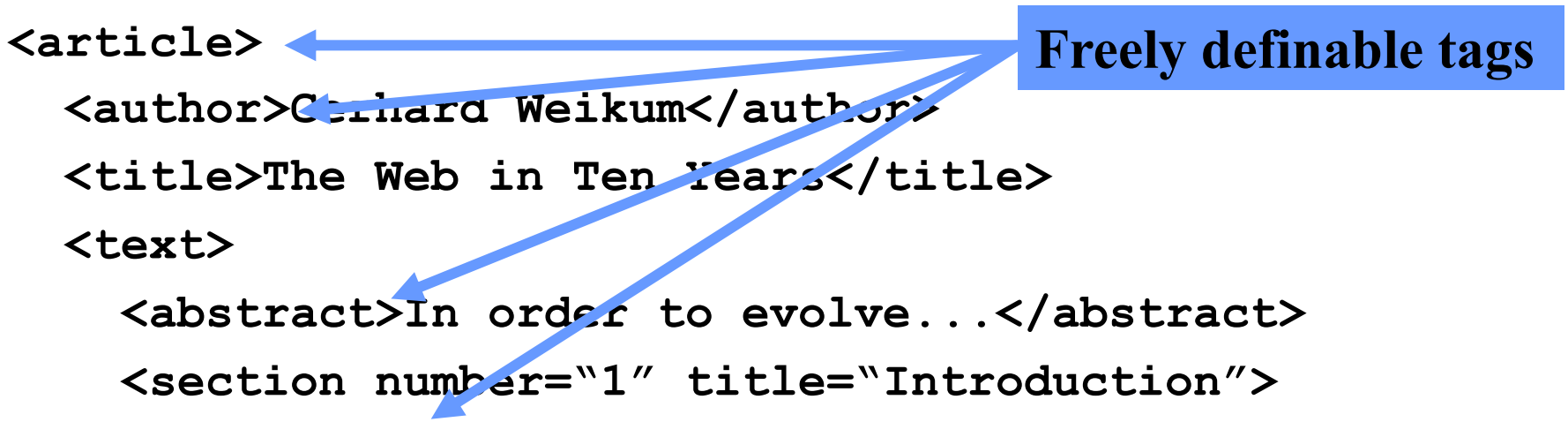
App. Scenario 4: Document Markup

- Document Markup adds structural and semantic information to documents, e.g.
  - Sections, Subsections, Theorems, …
  - Cross References
  - Literature Citations
  - Index Entries
  - Named Entities

A Simple XML Document

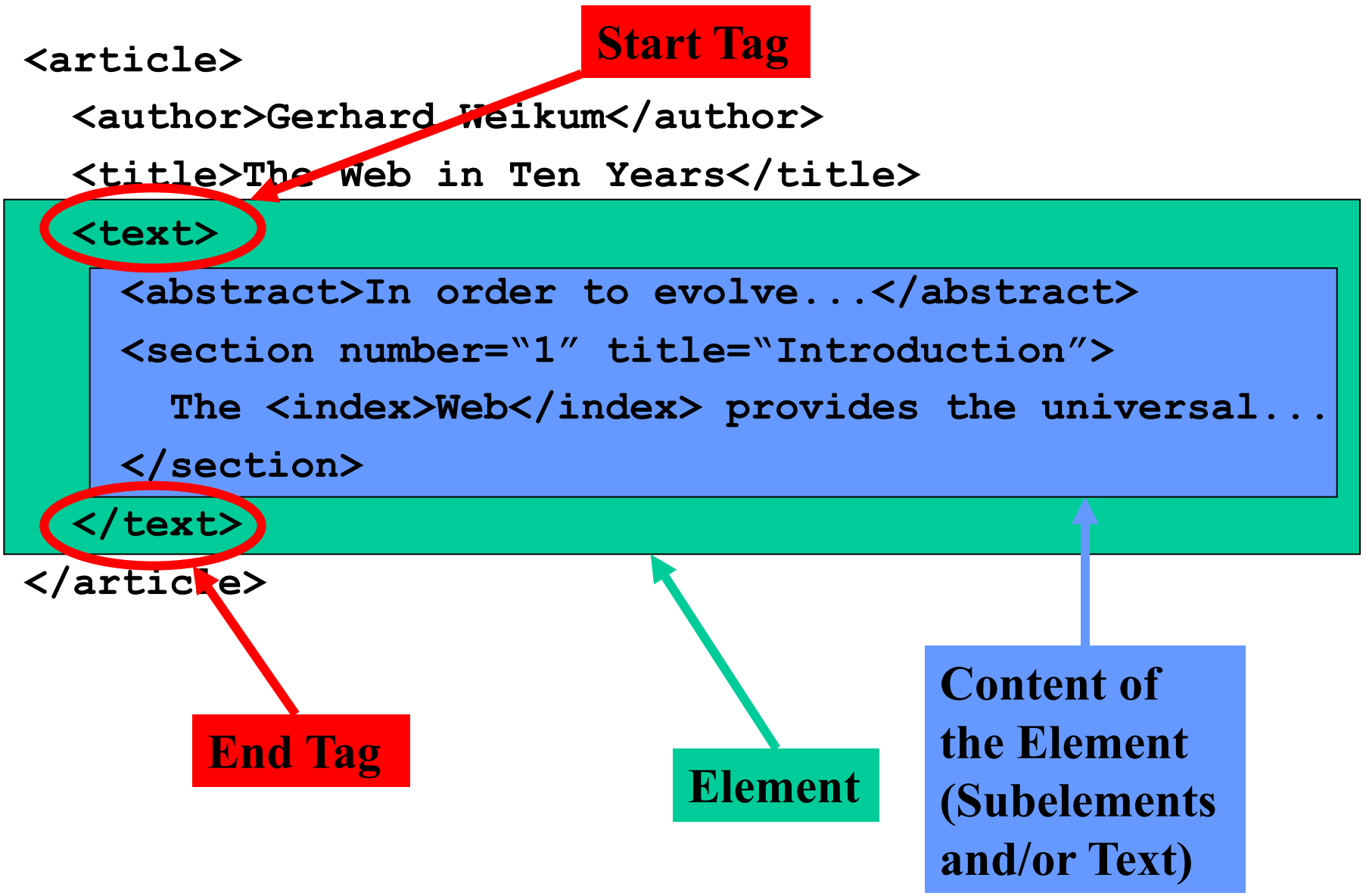
```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

A Simple XML Document



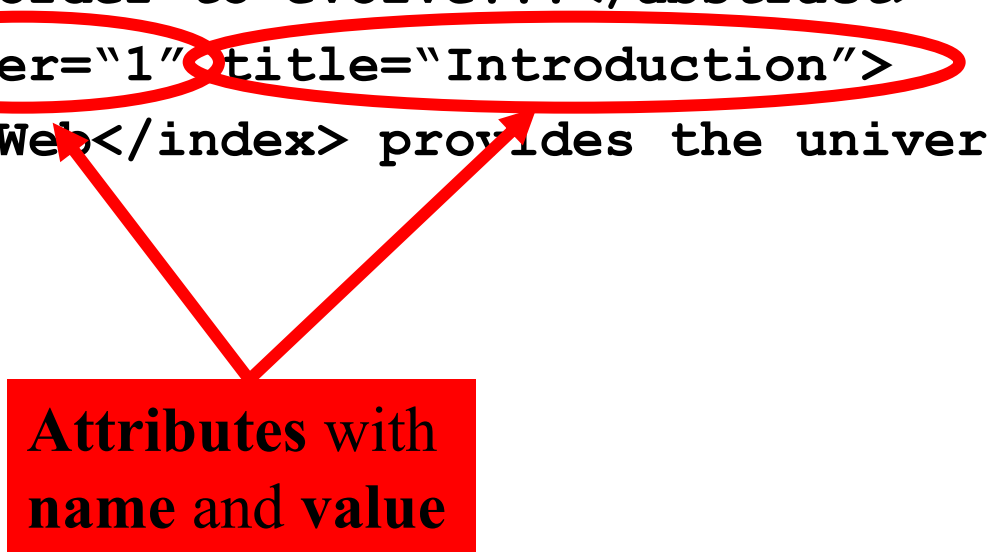

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

A Simple XML Document



A Simple XML Document

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```



Elements in XML Documents

- (Freely definable) **tags**: **`article`**, **`title`**, **`author`**
  - with start tag: **`<article>`** etc.
  - and end tag: **`</article>`** etc.
- **Elements**: **`<article> ... </article>`**
- Elements have a **name** (**`article`**) and a **content** (. . .)
- Elements may be nested.
- Elements may be empty: **`<this_is_empty/>`**
- Element content is typically parsed character data (PCDATA), i.e., strings with special characters, and/or nested elements (*mixed content* if both).
- Each XML document has exactly one root element and forms a tree.
- Elements with a common parent are ordered.

Elements vs. Attributes

Elements may have **attributes** (in the start tag) that have a **name** and a **value**, e.g. **`<section number="1">`**.

What is the difference between elements and attributes?

- Only one attribute with a given name per element (but an arbitrary number of subelements)
- Attributes have no structure, simply strings (while elements can have subelements)

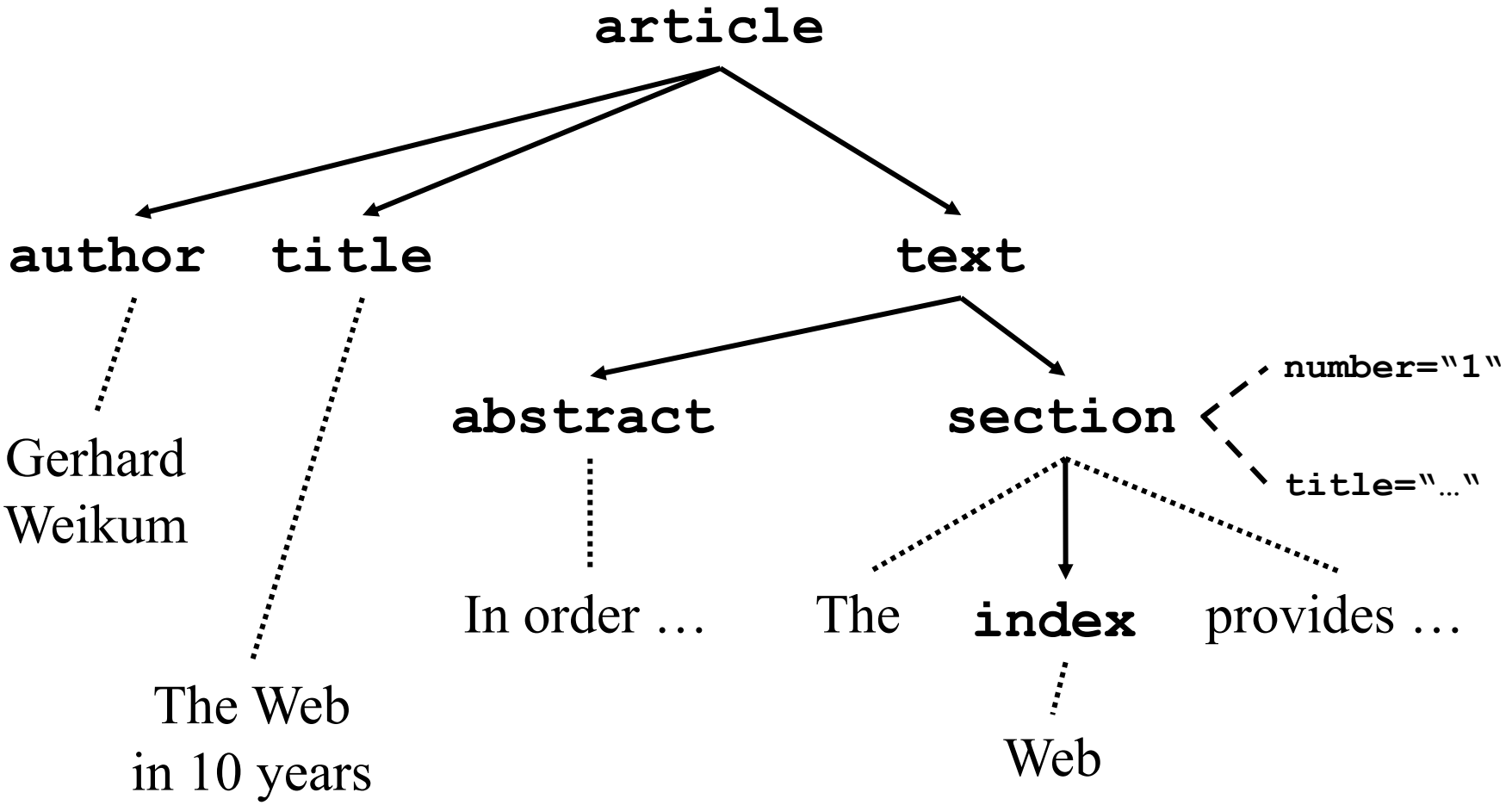
As a *rule of thumb*:

- Content into elements
- Metadata into attributes

Example:

```
<person born="1912-06-23" died="1954-06-07">
Alan Turing</person> proved that…
```

XML Documents as Ordered Trees



More on XML Syntax

- Some special characters must be escaped using **entities:**

  `<` → `<`

  `&` → `&`

  (will be converted back when reading the XML doc)

- Some other characters may be escaped, too:

  `>` → `>`

  `"` → `"`

  `'` → `'`

Well-Formed XML Documents

A **well-formed** document must adher to, among others, the following rules:

- Every start tag has a matching end tag.
- Elements may nest, but must not overlap.
- There must be exactly one root element.
- Attribute values must be quoted.
- An element may not have two attributes with the same name.
- Comments and processing instructions may not appear inside tags.
- No unescaped `<` or `&` signs may occur inside character data.

Well-Formed XML Documents

A **well-formed** document must adher to, among others, the following rules:

- Every start tag has a matching end tag.
- Elements may nest, but must not overlap.
- The
- Att
- An me nam
- Comments and processing instructions may not appear inside tags.
- No unescaped < or & signs may occur inside character data.

Only well-formed documents can be processed by XML parsers.

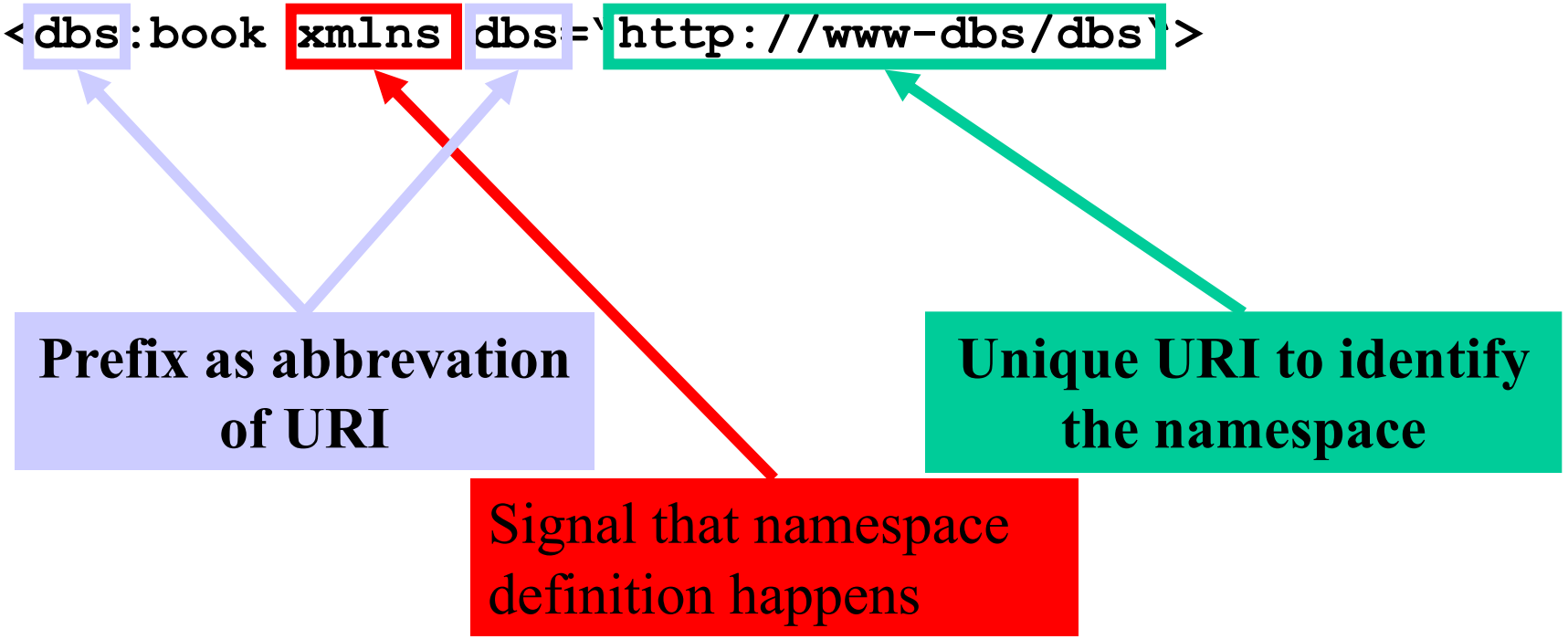
2.3 Namespaces

```
<library>
  <description>Library of the CS Department</description>
  <book bid="HandMS2000">
    <title>Principles of Data Mining</title>
    <description>
      Short introduction to <em>data mining</em>, useful
      for the IRDM course
    </description>
  </book>
</library>
```

Semantics of the `description` element is ambigous
Content may be defined differently
Renaming may be impossible (standards!)

$\Rightarrow$ Disambiguation of separate XML applications using unique prefixes

Namespace Syntax



Namespace Example

```
<dbs:book xmlns:dbs="http://www-dbs/dbs">
  <dbs:description> ... </dbs:description>
  <dbs:text>
    <dbs:formula>
      <mathml:math
   xmlns:mathml="http://www.w3.org/1998/Math/MathML">
        ...
      </mathml:math>
    </dbs:formula>
  </dbs:text>
</dbs:book>
```

Default Namespace

- Default namespace may be set for an element and its content (but *not* its attributes):

```
<book xmlns="http://www-dbs/dbs">
  <description>...</description>
<book>
```

- Can be overridden in the elements by specifying the namespace there (using prefix or default namespace)

3.1 Document Type Definitions

Sometimes XML is *too* flexible:

- Most Programs can only process a subset of all possible XML applications
- For exchanging data, the format (i.e., elements, attributes and their semantics) must be fixed

$\Rightarrow$**Document Type Definitions** (**DTD**) for establishing the vocabulary for one XML application (in some sense comparable to *schemas* in databases)

A document is **valid with respect to a DTD** if it conforms to the rules specified in that DTD.

Most XML parsers can be configured to validate.

DTD Example: Elements

```
<!ELEMENT article     (title,author+,text)>
<!ELEMENT title       (#PCDATA)>
<!ELEMENT author      (#PCDATA)>
<!ELEMENT text        (abstract,section*,literature?)>
<!ELEMENT abstract    (#PCDATA)>
<!ELEMENT section     (#PCDATA|index)+>
<!ELEMENT literature  (#PCDATA)>
<!ELEMENT index       (#PCDATA)>
```

Content of the `title` element is parsed character data

Content of the `text` element may contain zero or more `section` elements in this position

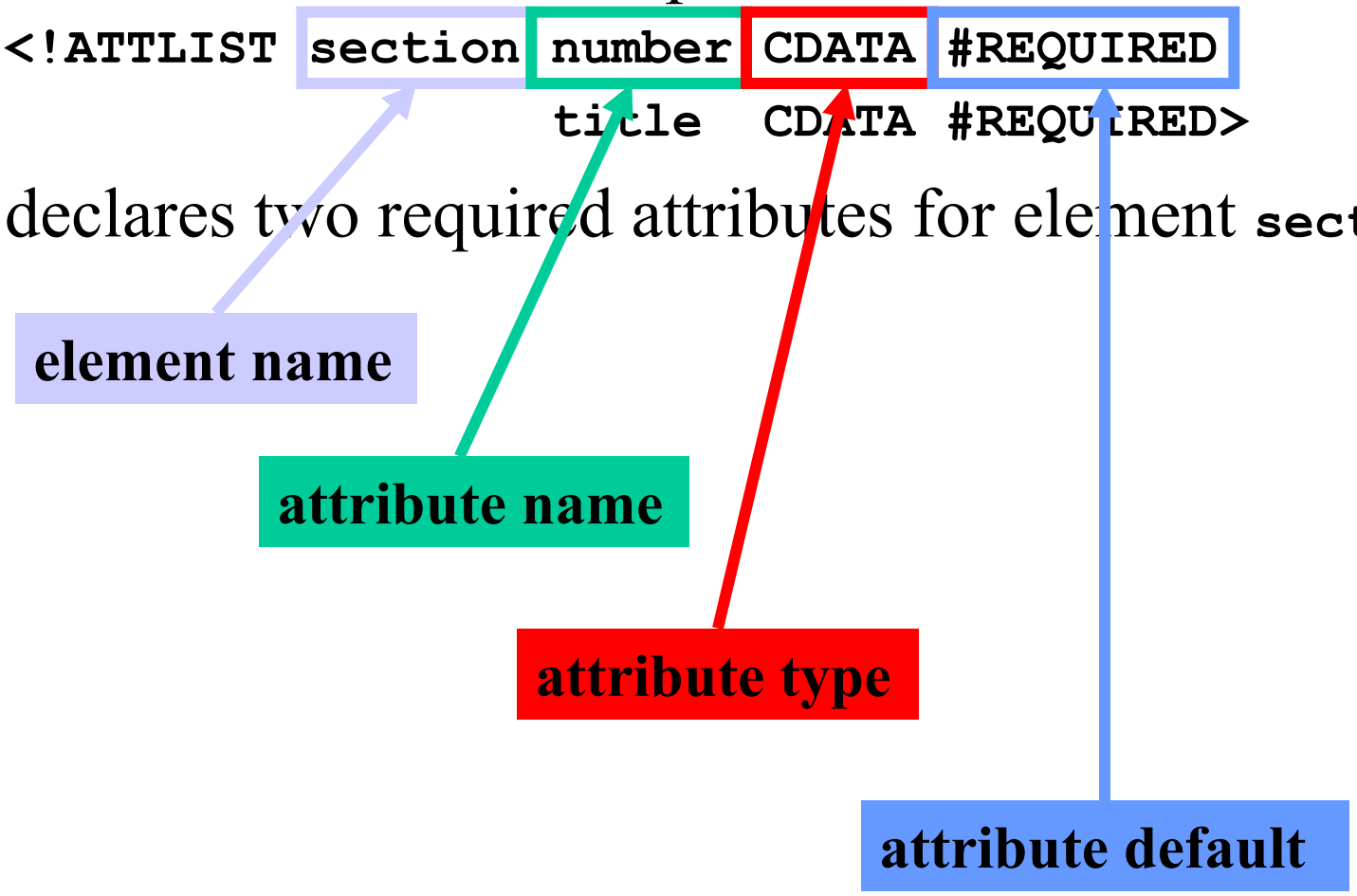
Content of the `article` element is a `title` element, followed by one or more `author` elements, followed by a `text` element

Attribute Declarations in DTDs

Attributes are declared per element:

```
<!ATTLIST section number CDATA #REQUIRED
                title  CDATA #REQUIRED>
```

declares two required attributes for element `section`.



Attribute Declarations in DTDs

Attributes are declared per element:

```
<!ATTLIST section number CDATA #REQUIRED
                 title  CDATA #REQUIRED>
```

declares two required attributes for element `section`.

Possible attribute defaults:

- `#REQUIRED` is required in each element instance
- `#IMPLIED` is optional
- `#FIXED default` always has this default value
- `default` has this default value if the attribute is omitted from the element instance

Attribute Types in DTDs

- **`CDATA`** string data
- **`(A1|…|An)`** enumeration of all possible values of the attribute (each is XML name)
- **`ID`** unique XML name to identify the element
- **`IDREF`** refers to **`ID`** attribute of some other element („intra-document link“)
- **`IDREFS`** list of **`IDREF`**, separated by white space
- plus some more

Attribute Examples

```
<ATTLIST publication type  (journal|inproceedings) #REQUIRED
                    pubid ID #REQUIRED>
<ATTLIST cite       cid   IDREF #REQUIRED>
<ATTLIST citation   ref   IDREF #IMPLIED
                    cid   ID #REQUIRED>

<publications>
  <publication type="journal" pubid="Weikum01">
    <author>Gerhard Weikum</author>
    <text>In the Web of 2010, XML <cite cid=„12"/>...</text>
    <citation cid=„12" ref=„XML98"/>
    <citation cid=„15">...</citation>
  </publication>
  <publication type="inproceedings" pubid="XML98">
    <text>XML, the extended Markup Language, ...</text>
  </publication>
</publications>
```

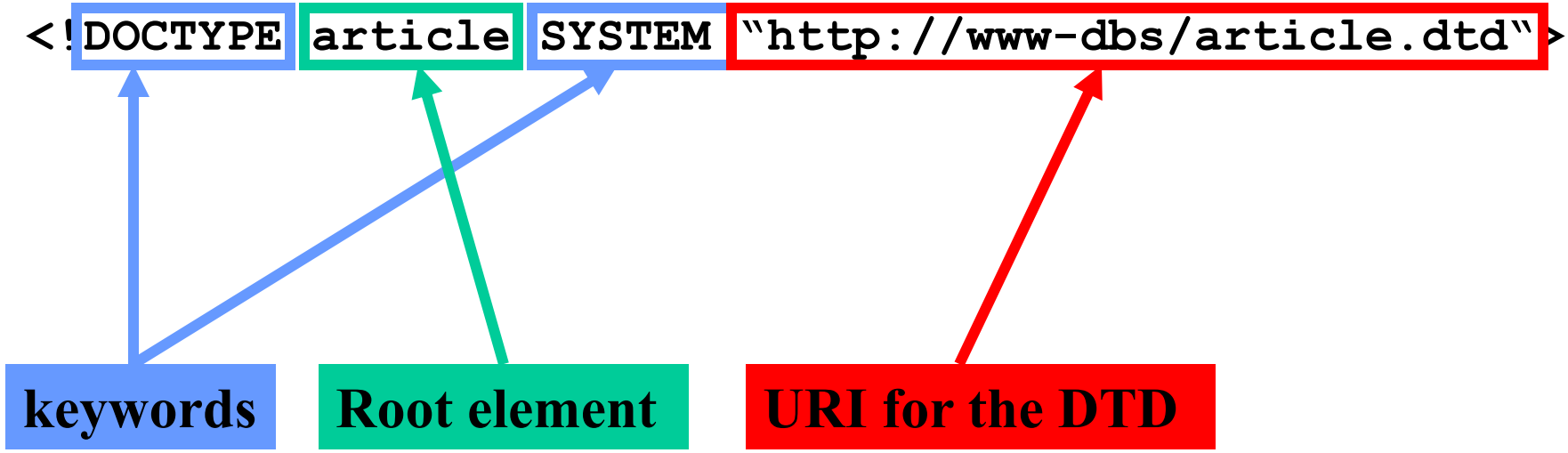
Attribute Examples

```
<ATTLIST publication type   (journal|inproceedings) #REQUIRED
                    pubid ID #REQUIRED>
<ATTLIST cite       cid   IDREF #REQUIRED>
<ATTLIST citation   ref   IDREF #IMPLIED
                    cid   ID #REQUIRED>

<publications>
  <publication type="journal" pubid="Weikum01">
    <author>Gerhard Weikum</author>
    <text>In the Web of 2010, XML <cite cid=„12"/>...</text>
    <citation cid=„12" ref=„XML98"/>
    <citation cid=„15">...</citation>
  </publication>
  <publication type="inproceedings" pubid="XML98">
    <text>XML, the extended Markup Language, ...</text>
  </publication>
</publications>
```

Linking DTD and XML Docs

- Document Type Declaration in the XML document:



Flaws of DTDs

- No support for basic data types like integers, doubles, dates, times, …
- No structured, self-definable data types
- No type derivation
- id/idref links are quite loose (target is not specified)

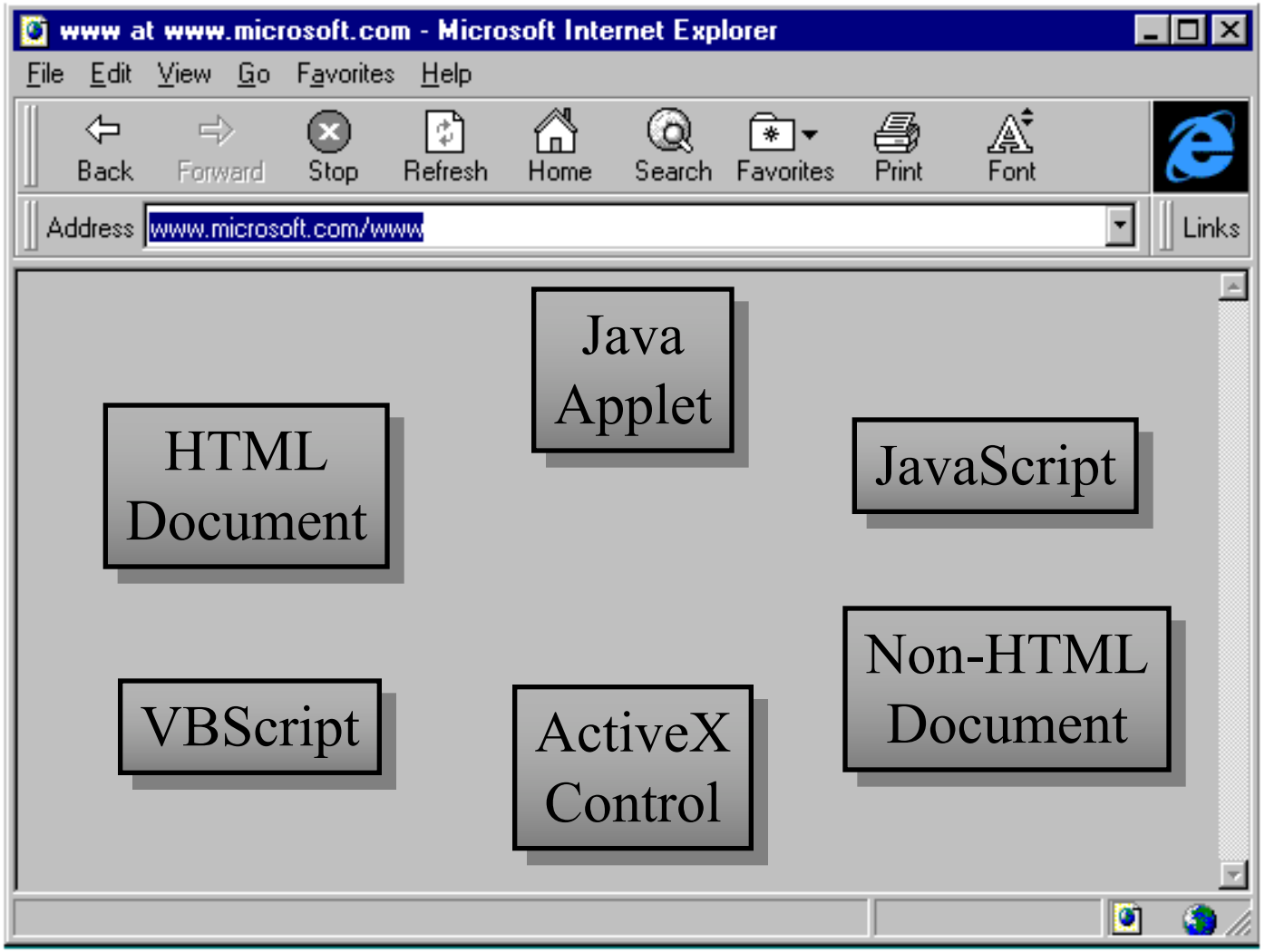
4.1 XPath

- XPath is a simple language to identify parts of the XML document (for further processing)
- XPath operates on the tree representation of the document
- Result of an XPath expression is a set of elements or attributes
- Discuss abbreviated version of XPath

What is ActiveX?

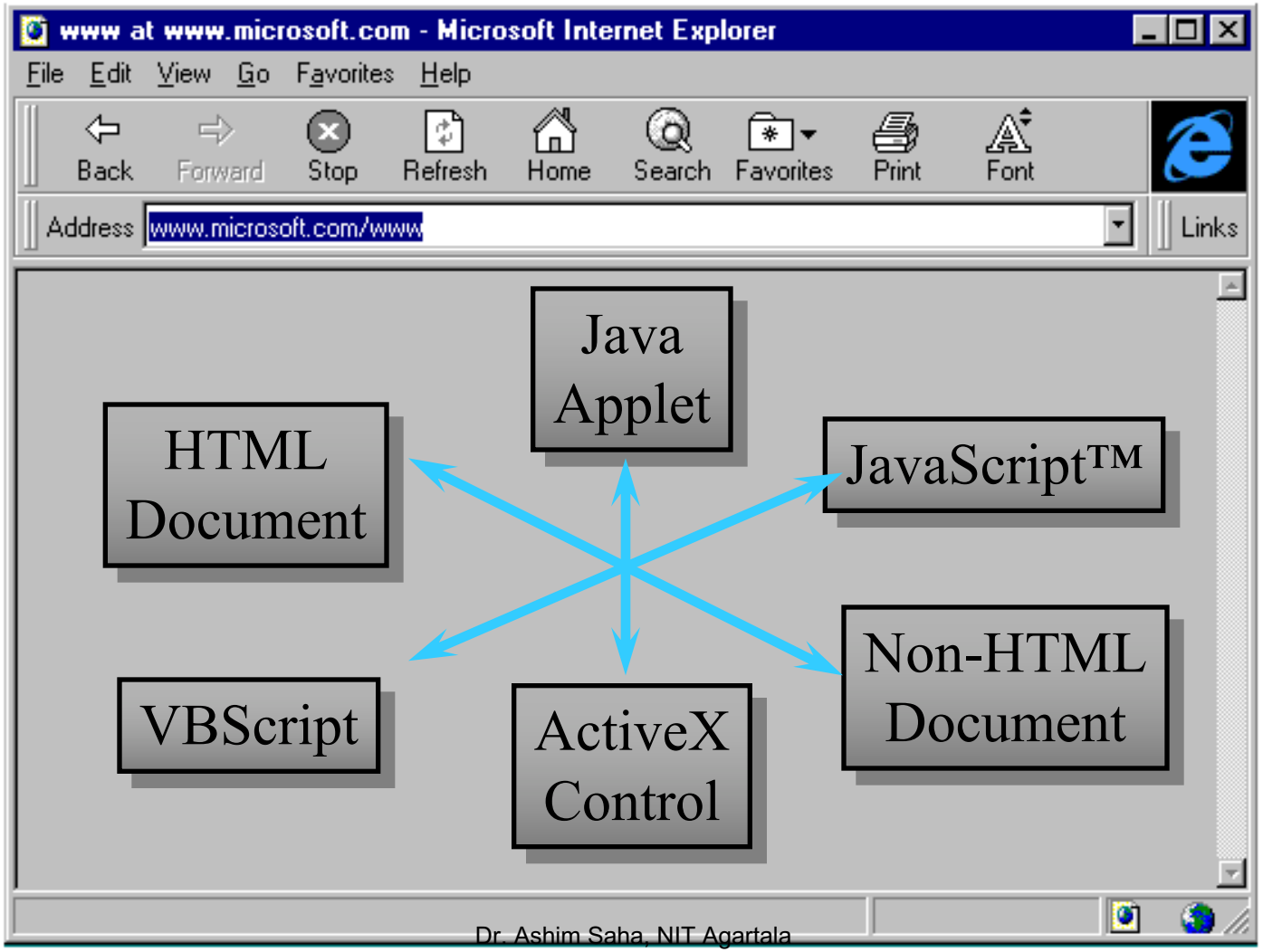
- A marketing name for a set of technologies and services, all based on the Component Object Model (COM)

The Web Today: “Islands” of Code



ActiveX Bridges the “Islands”

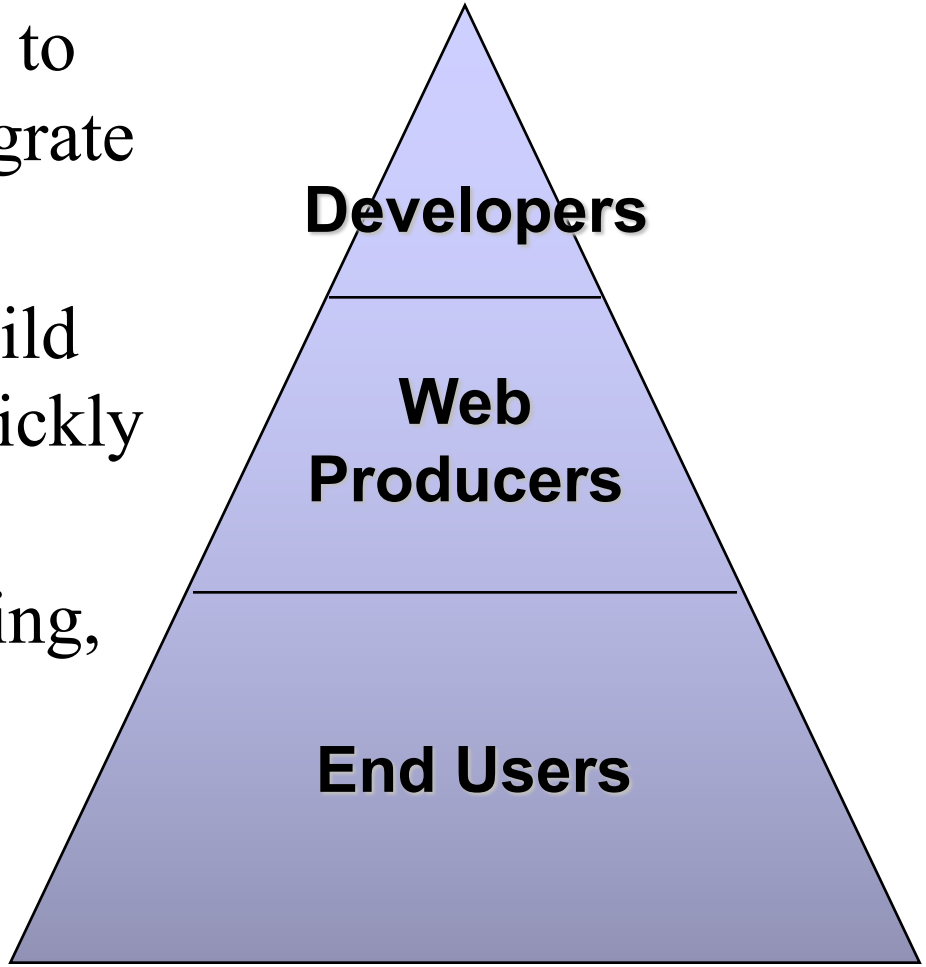
Makes it easy to integrate and reuse any component



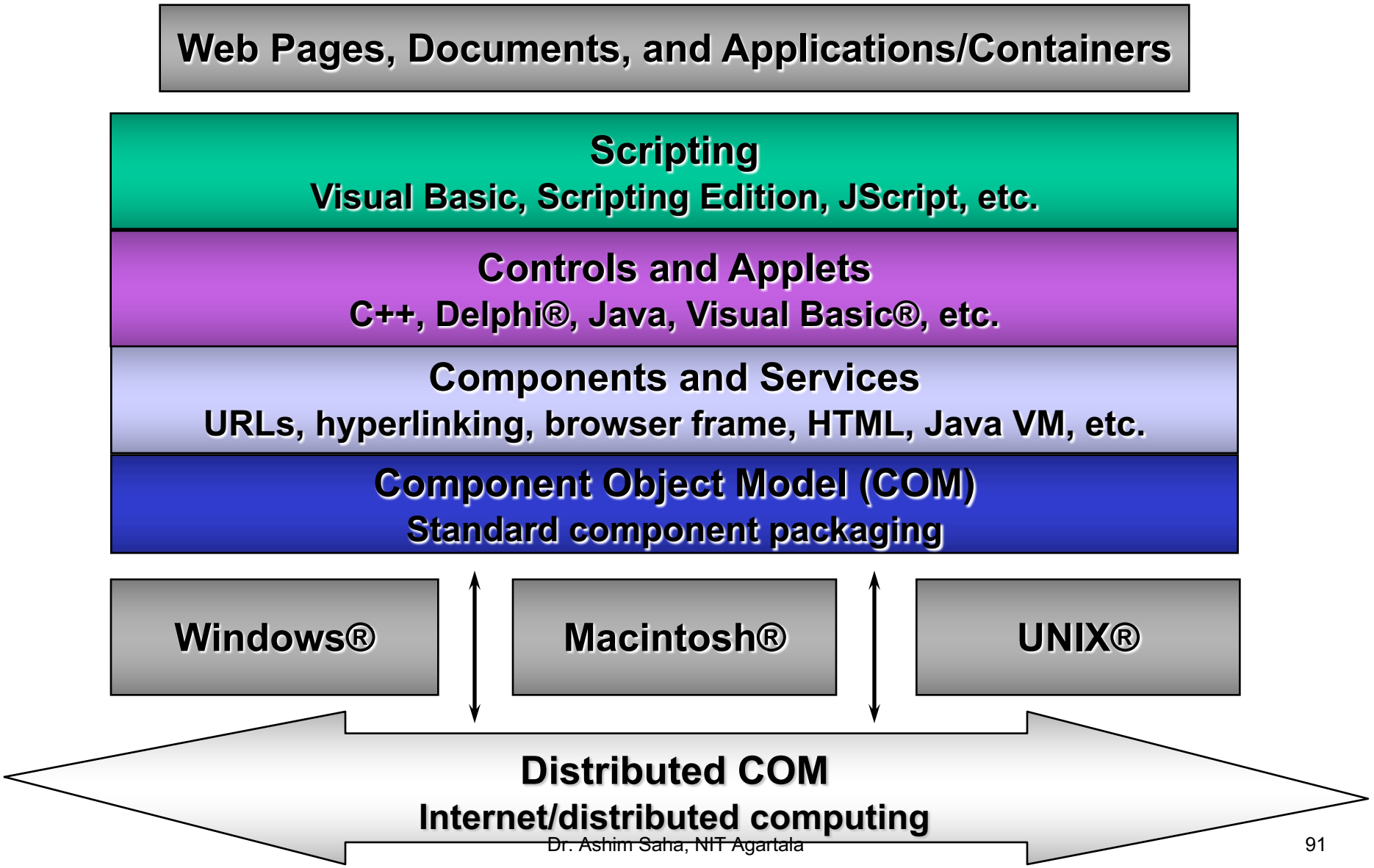
Motivation For ActiveX

Bring Component-based Development to the Internet

- Software Developers - want to easily create, reuse and integrate software components
- Web Producers - want to build rich, engaging Web sites quickly and easily
- End Users - want an intriguing, exciting experience



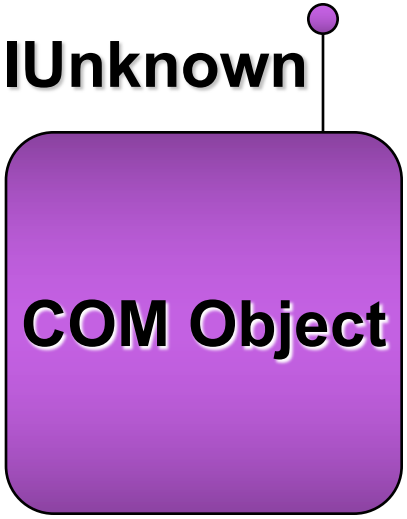
Elements Of ActiveX



Component Object Model (COM)

Foundation for ActiveX components

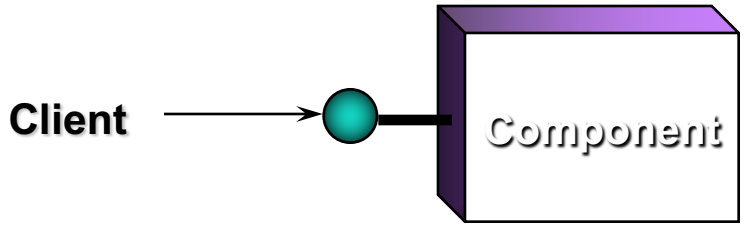
- Versionable, programmable
- Lightweight, fast
- Open standard
  - Language and tool-neutral
  - Works with today's applications
  - Cross-platform
- ActiveX components are COM objects
- Distributed COM extends interactions between components across networks



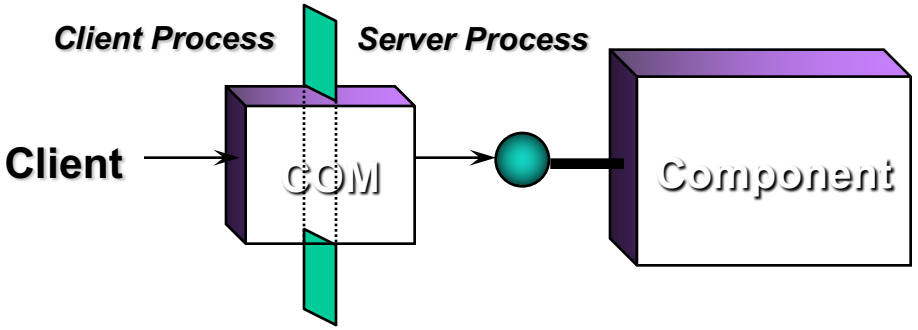
The COM Architecture

A scalable programming model

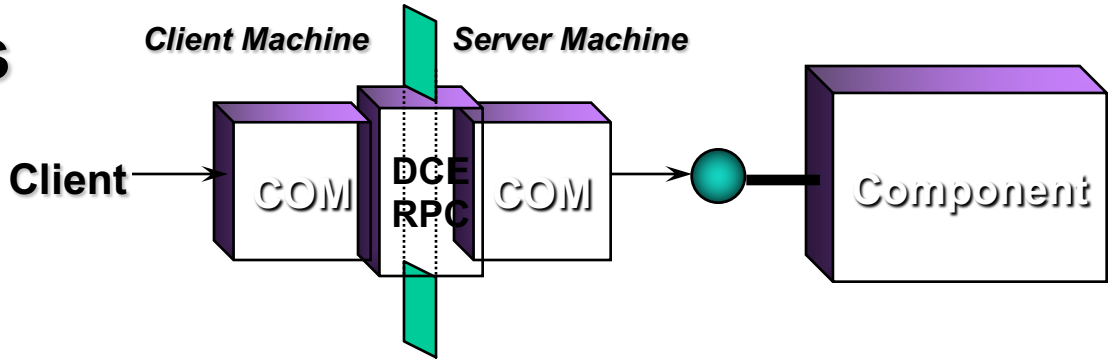
- **In the same process**
  - Fast, direct function calls



- **On the same machine**
  - Fast, secure IPC



- **Across machines**
  - Secure, reliable and flexible DCE-RPC based *DCOM* protocol



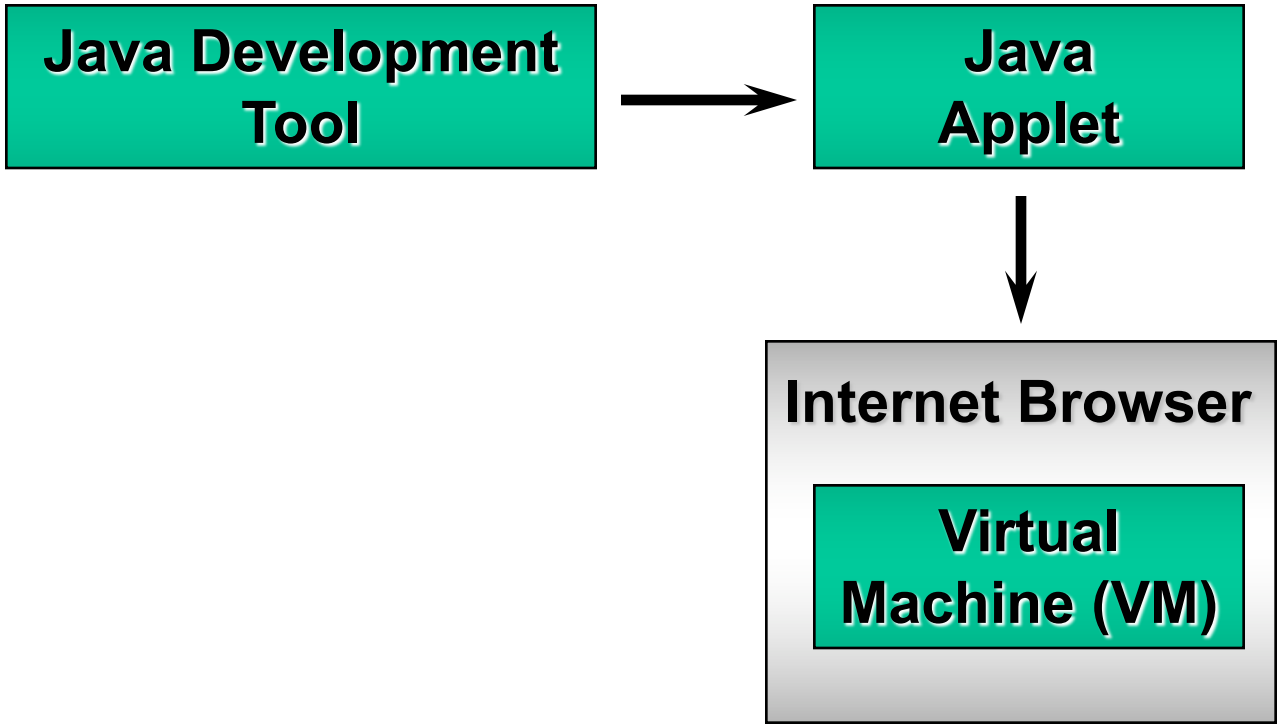
What Are ActiveX Controls?

- Controls are components that can be manipulated visually by GUI development tools
  - Distinction between design-time and run-time user-interface
- Code that makes the Web alive
- Programmable by other components and scripts

ActiveX and Java

What is Java?

- Programming language
- Virtual machine/Byte codes



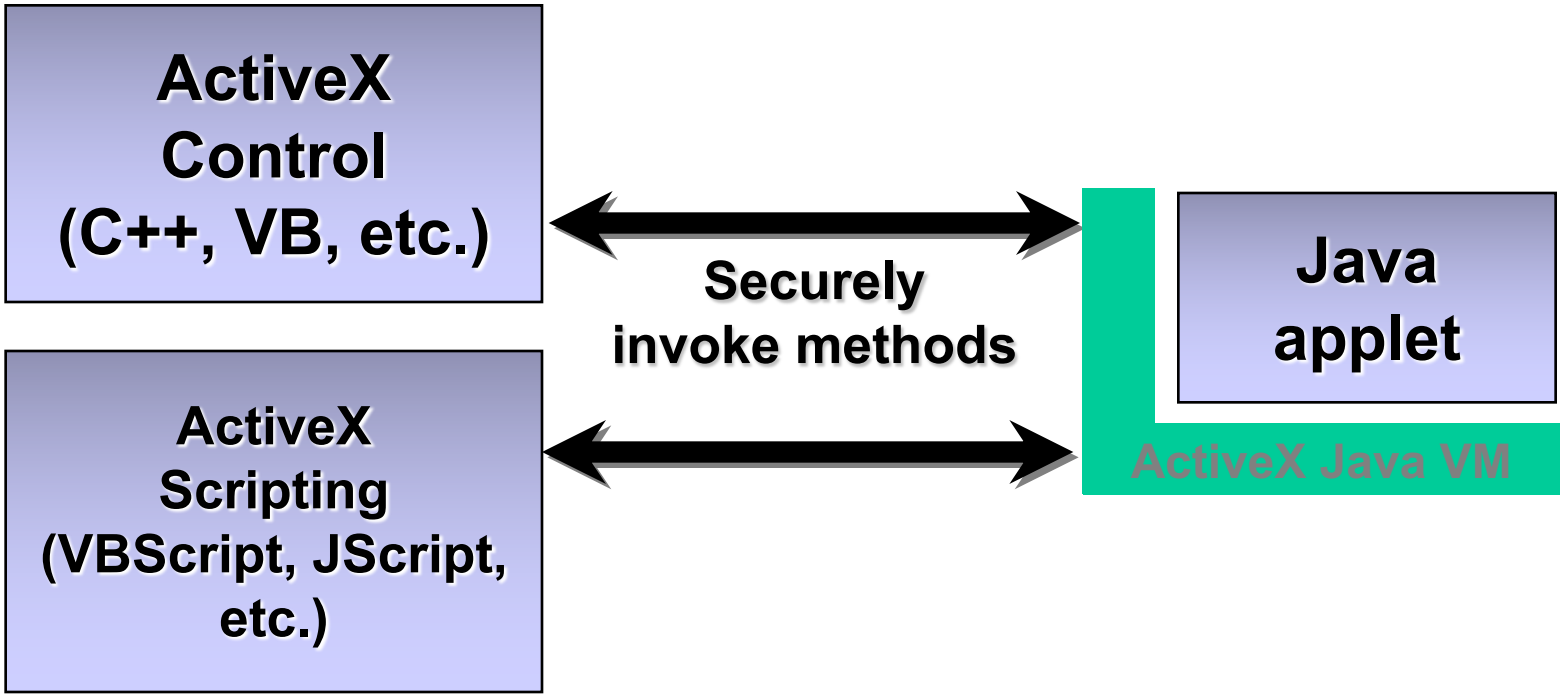
ActiveX Complements Java

- ActiveX and Java DO NOT COMPETE!
- ActiveX is a set of integration technologies
  - Every Java object *is* an ActiveX object!
- ActiveX is *NOT*
  - A language
  - A platform
  - An operating system

How Does Java Fit With ActiveX?

- Java VM is an ActiveX component
  - Run Java applets in any application, not just browser
- Java component is an ActiveX component
- Automatic integration of Java applets with other languages and scripts
  - Java applets talk to other Java applets
  - Java applets talk to ActiveX components
  - Gives Java applets access to all PC functionality

How Does Java Fit With ActiveX?



Microsoft's Java Commitment

- Microsoft is hard core about Java the language
  - Reference Java VM for Windows
  - Fastest Just-in-Time Java Compiler
  - Visual J++ development tool
  - Java support on the server
- Windows is the best execution environment for Java applets
  - Fastest safest, most functional
- Microsoft will compete with Java the operating system

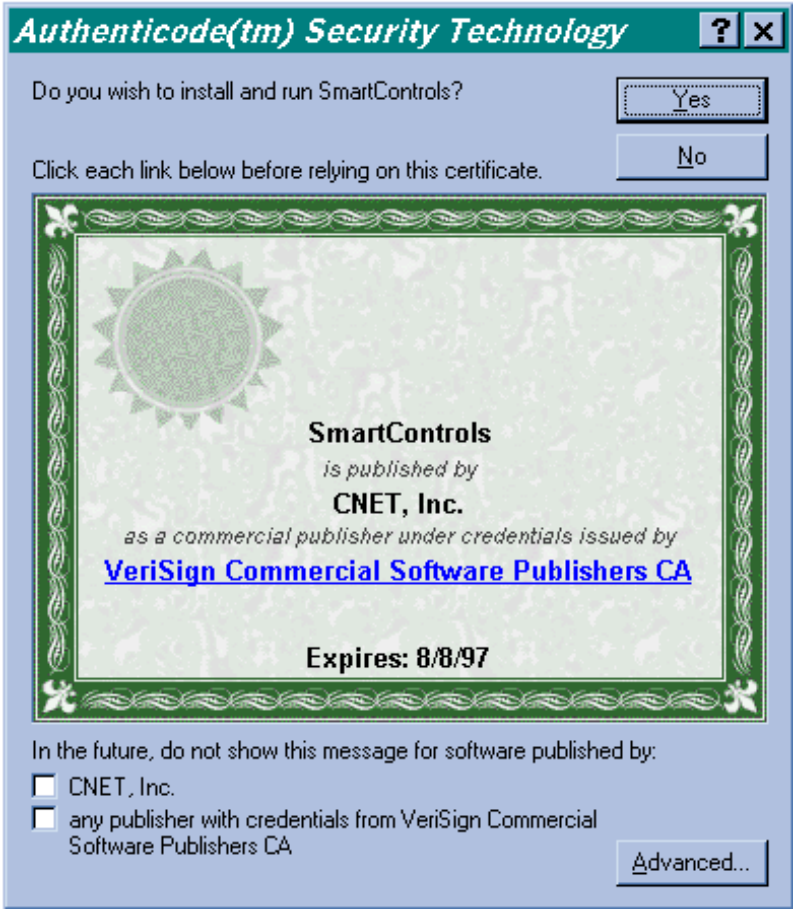
Code-signing and Sandboxing

- Code-signing - “shrink wraps” code
  - Identifies publisher; validates integrity of code
  - Doesn’t prevent bugs or malicious operation
  - Gives accountability for market and legal recourse

- Sandbox - code isolated inside virtual machine
  - Doesn’t prevent bugs; limits malicious operation by limiting functionality
  - Least common denominator capabilities
  - Relies on airtight implementation, proven difficult in practice

Authenticode™

- Code-signing feature in Internet Explorer from v3.0
- Uses industry standard certificates (X.509)
- Contractual relationship with multiple certificate authorities
- Allows end users or administrators to set “trust policies”



What Are ActiveX Documents?

- Lets users view and edit non-HTML documents through the browser
- ActiveX Documents
  - Use the entire client area
  - Provides printing support
  - Menu merging
  - Toolbars
  - Integrates existing documents into browser or any other application

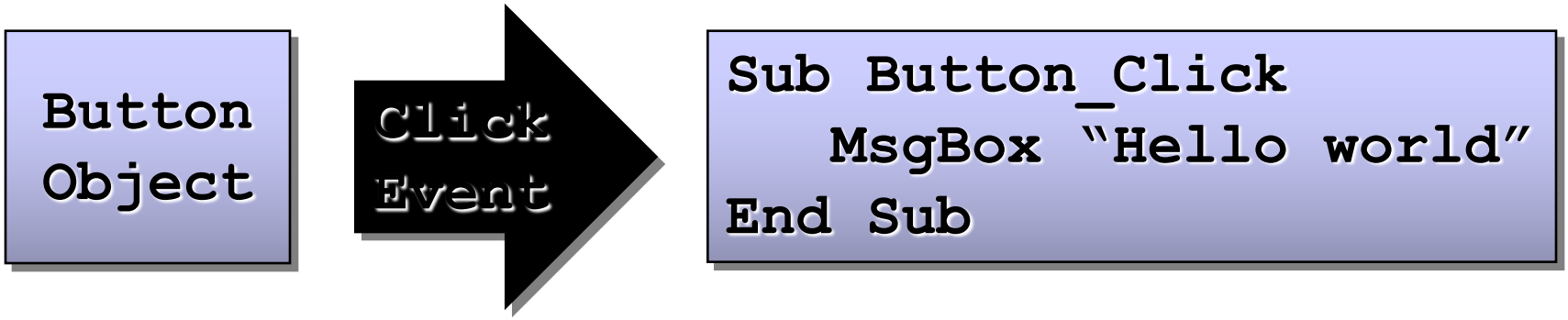
What Is ActiveX Scripting?

- Mechanism for rapid “writing” together of diverse set of components
- Supports any scripting language - VBScript, JScript, Perl, PowerScript etc.
- Scripting languages are the most broadly accessible development tools
- Scripting plus HTML is the fastest and easiest cross platform solution
- Internet Explorer 3.0 ships with VBScript, JScript runtimes

Scripting: A Simple Solution

Pioneered by Visual Basic

- Objects expose events for the purpose of coordinating their activity with the rest of the world
- Script code can be attached to an object's events to customize behavior - this is "scripting"



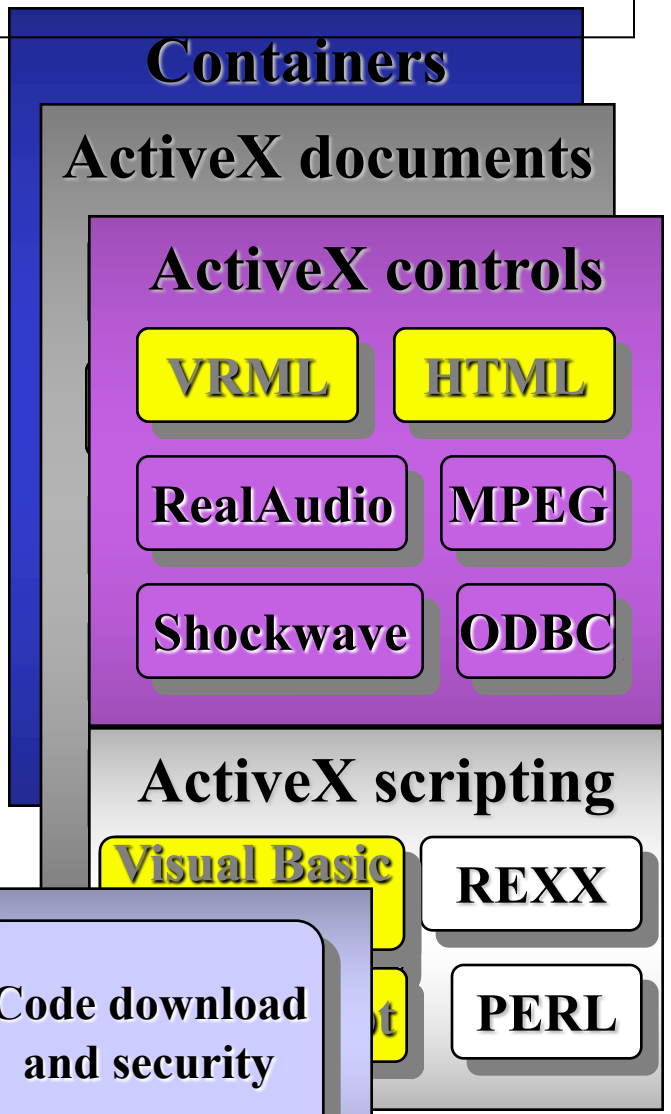
VBScript

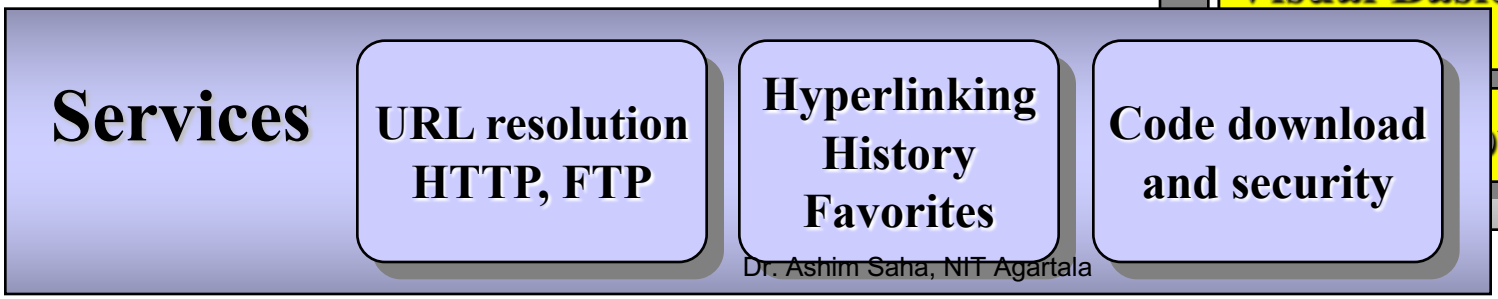
- Small, fast, and safe
- Scripts both ActiveX Controls and Java applets
- Proper subset of Visual Basic for Applications
  - 100% compatible with Visual Basic
  - Built into Internet Explorer 3.0 and Windows NT™
  - Over 3 million users familiar with Visual Basic
- Cross-platform for easy mobility
  - All clients, servers
  - Licensable, not just for the Internet
- Free distribution

JScript

- High performance JavaScript-compatible scripting language
- More secure than JavaScript
- JavaScript related to Java only by name
- JavaScript Netscape proprietary despite public commitments
- JScript and VBScript source licensable (free!)
- Ships with Internet Explorer 3.0

Architecture Overview





ActiveX Benefits: Power

- Develop highest performance, “no compromises” applications
- Provides richest and most robust array of programming interfaces
- Full and open support for industry standards: HTTP, HTML, TCP/IP, Java, etc.
- Authenticated, accountable environment
- Built on solid, proven foundation of COM

ActiveX Benefits: Choice

- Mix and match from huge industry arsenal of tools, languages, code and expertise
- Deliver applications for Internet and/or Intranet
- Run components in the browser or other applications (any container)
- Program with any tool or language
- Supports Macintosh®, UNIX® and Windows®
- Integrate components using any scripting language

ActiveX Benefits

Broadest industry support

- Market proven, compatible, third generation technology
- Choose today from over 1,000 controls from hundreds of vendors
  - Giga Information Group: $240 million industry in 1996, growing to over $2 billion by 2000
- Run components in the most popular apps
  - Microsoft Office, Internet Explorer, Lotus® Notes, Lotus SmartSuite®, Sybase® PowerBuilder
- Program with familiar tools
  - Microsoft Visual C++, Microsoft Visual Basic, Borland C++, Borland Delphi, Java and others

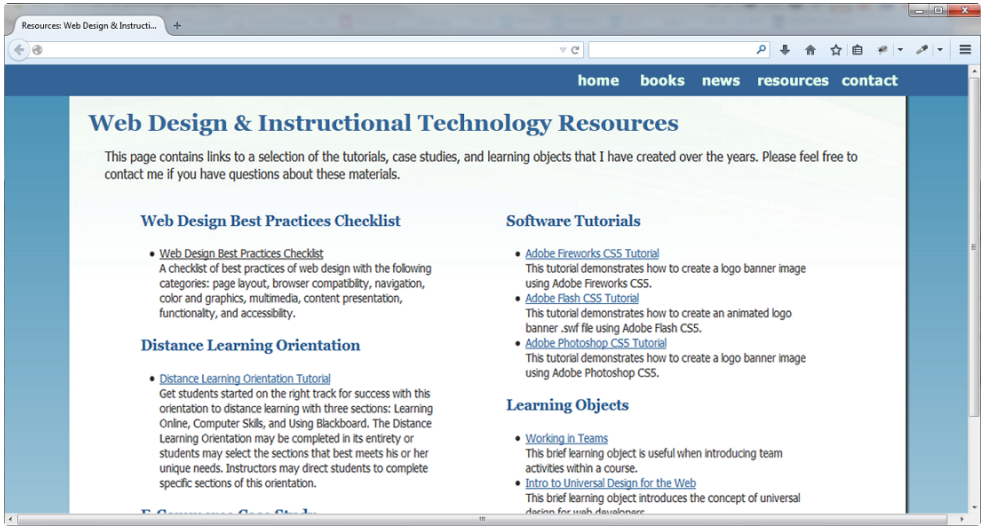
Summary

- ActiveX is the first and only component technology to achieve commercial success
- Today's thriving ActiveX software component industry makes it quick and easy to build great web sites
- Componentized architecture is not just the future; it is here today with ActiveX

Multimedia and Web Application

Design for Your Target Audience

Consider the target audience of these sites.

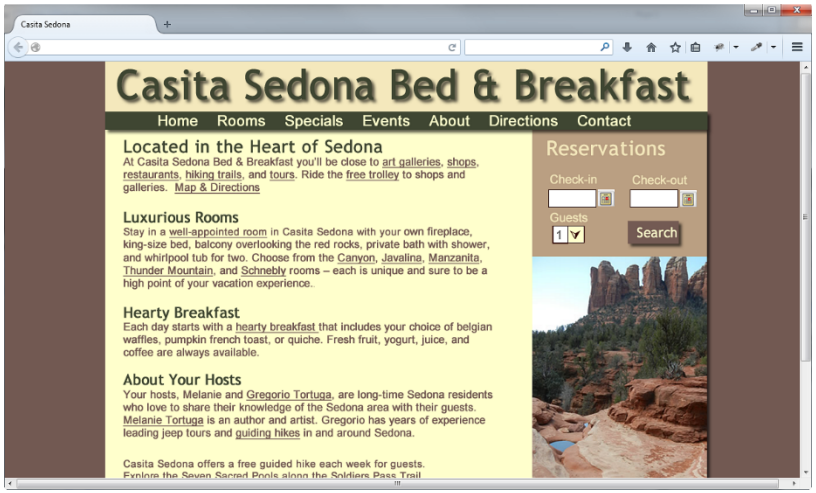


Browser Compatibility

- Web pages do NOT look the same in all the major browsers
- Test with current and recent versions of:
  - Internet Explorer
  - Firefox, Mozilla
  - Opera
  - Safari
- Progressive Enhancement:
  - Website functions well in browsers commonly used by your target audience
  - Add enhancements with CSS3 and/or HTML5 for display in modern browsers

Screen Resolution

- Test at various screen resolutions
  - Most widely used: 1024x768, 1366x768, and 1280x800

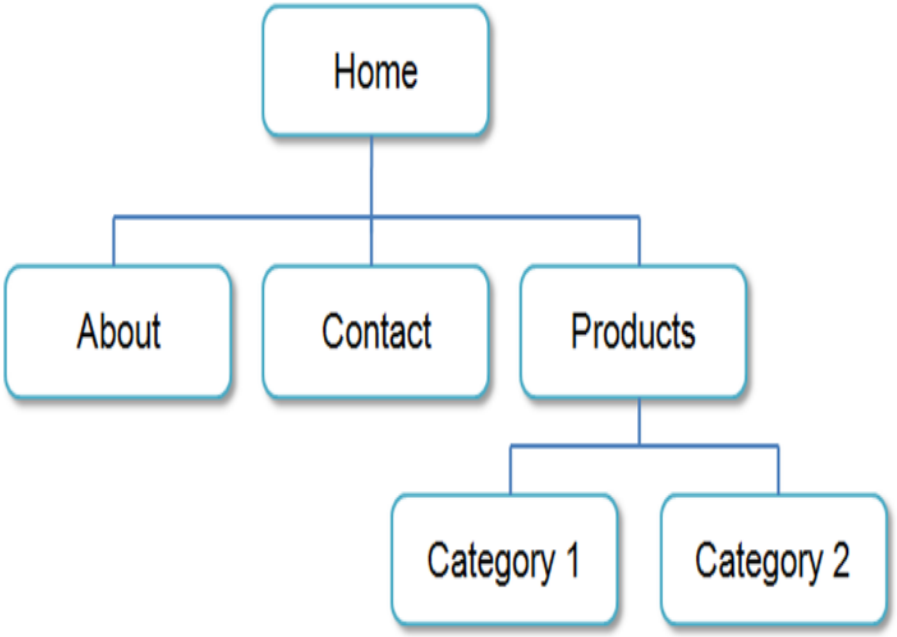


- Design to look good at various screen resolutions
  - Centered page content
  - Set to either a fixed or percentage width

Website Organization

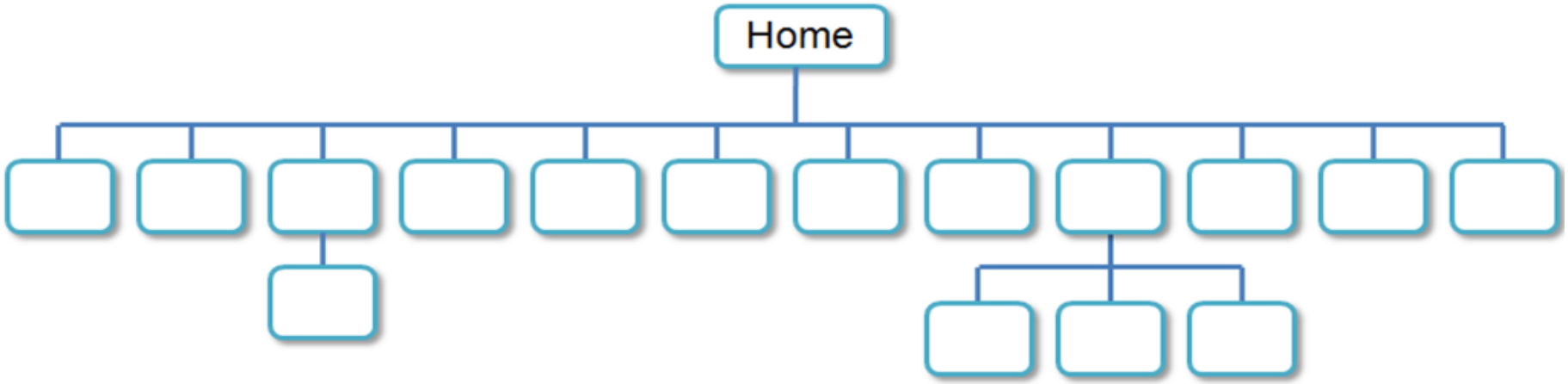
- Hierarchical
- Linear
- Random
  (sometimes called Web Organization)

Hierarchical Organization

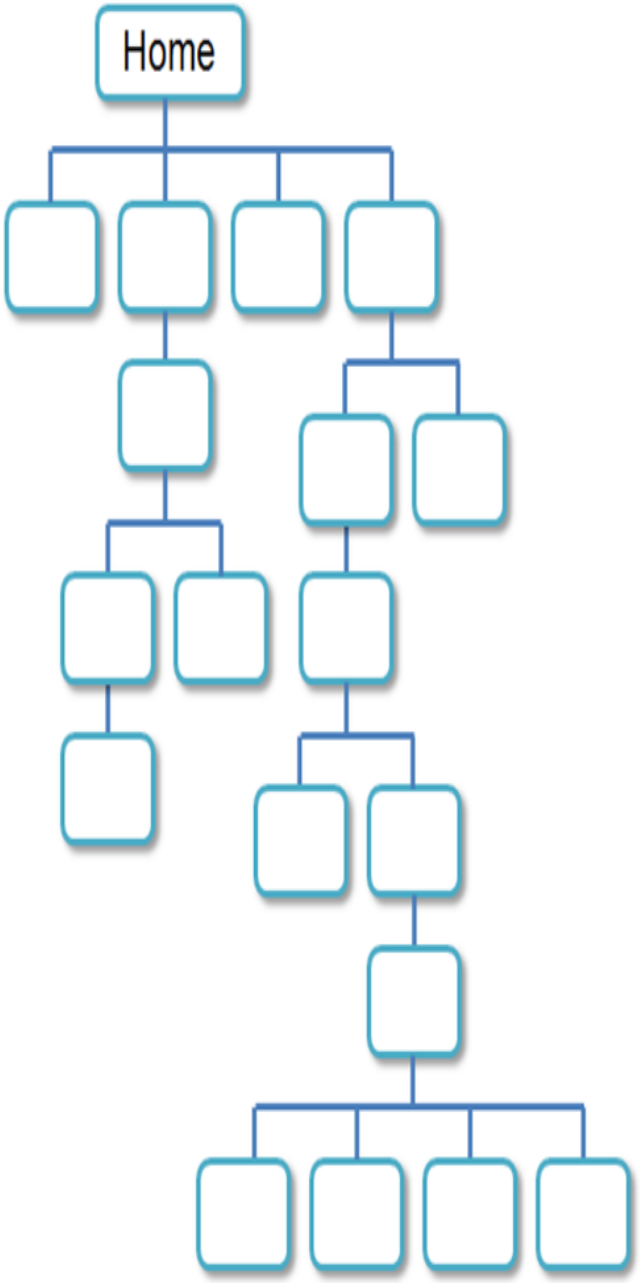


- A clearly defined home page
- Navigation links to major site sections
- Often used for commercial and corporate websites

Hierarchical: Too Shallow



- Be careful that the organization is not too shallow.
- Too many immediate choices → a confusing and less usable website.
- Group, or “chunk”, related areas



Hierarchical: Too Deep

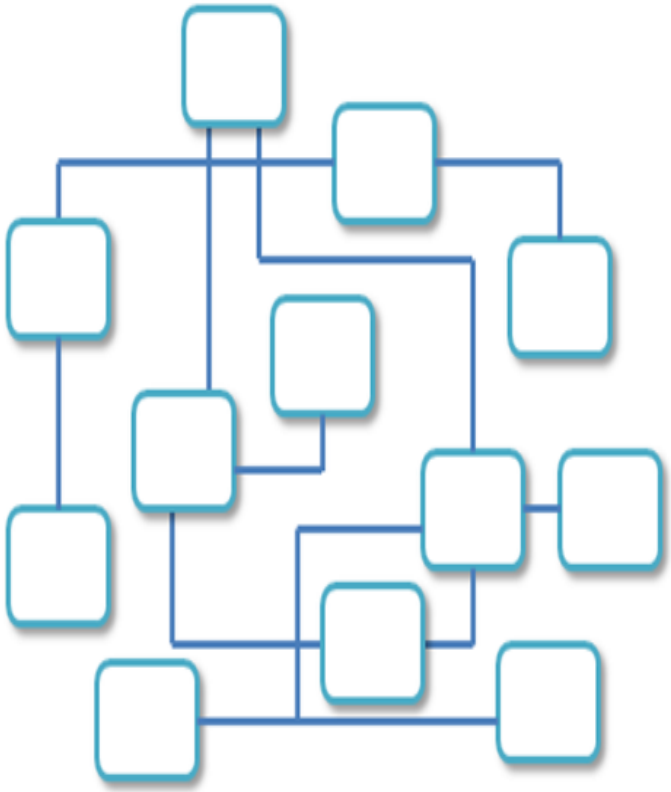
- Be careful that the organization is not too deep.
  - This results in many “clicks” needed to drill down to the needed page.
  - User Interface “Three Click Rule”
    - A web page visitor should be able to get from any page on your site to any other page on your site with a maximum of three hyperlinks.

Linear Organization

- A series of pages that provide a tutorial, tour, or presentation.
- Sequential viewing

Random Organization

- Sometimes called “Web” Organization
- Usually there is no clear path through the site
- May be used with artistic or concept sites
- Not typically used for commercial sites



Visual Design Principles

- Repetition
  - Repeat visual elements throughout design
- Contrast
  - Add visual excitement and draw attention
- Proximity
  - Group related items
- Alignment
  - Align elements to create visual unity

Design to Provide for Accessibility

"The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect." – Tim Berners-Lee

- Who benefits from increased accessibility?
  - A person with a physical disability
  - A person using a slow Internet connection
  - A person using an old, out-dated computer
  - A person using a mobile phone
- Legal Requirement: Section 508
- Standards: WCAG 2.0

Design for Accessibility

- Web Content Accessibility Guidelines 2.0
  WCAG 2.0
  - http://www.w3.org/TR/WCAG20/Overview
  - http://www.w3.org/WAI/WCAG20/quickref

- Based on Four Principles (**POUR**)
  1. **P**erceivable
     Content must be **easy to see or hear**
  2. **O**perable
     Interface components in the content must be **operable by both mouse and keyboard**
  3. **U**nderstandable
     Content and controls must be **easy to read and well-organized**
  4. **R**obust.
     Content use correct syntax and function on popular operating systems, browsers, and assistive technologies.

Writing for the Web

- Avoid long blocks of text
- Use bullet points
- Use headings and subheadings
- Use short paragraphs

Design “Easy to Read” Text

- Use common fonts:
  - Arial, Helvetica, Verdana, Times New Roman
- Use appropriate text size:
  - medium, 1em, 100%
- Use appropriate line length
  - Between 50-75 characters is recommended
- Use strong contrast between text & background
- Use columns instead of wide areas of horizontal text

More Text Design Considerations

- Carefully choose text in hyperlinks
  - Avoid “click here”
  - Hyperlink key words or phrases
  - Do not hyperlink **not entire sentences**
- Chek yur spellin (Check your spelling)

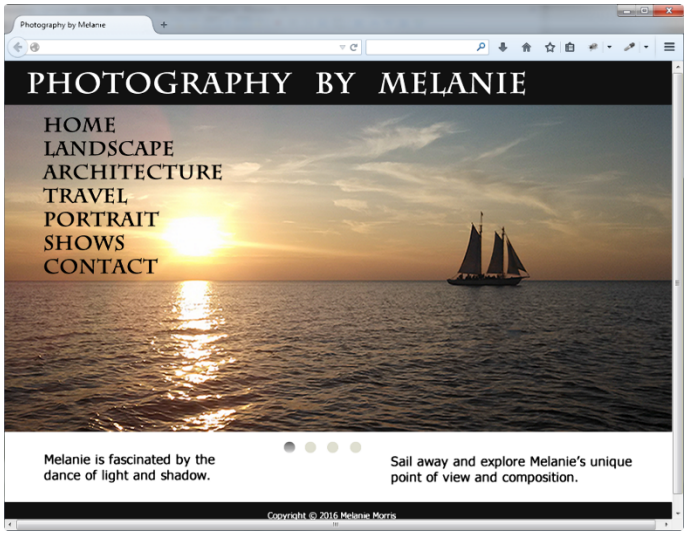
Using Color on Web Pages

- Computer monitors display color as intensities of red, green, and blue light
- RGB Color
- The values of red, green, and blue vary from 0 to 255.
- Hexadecimal numbers (base 16) represent these color values.

Web Color Palette

- A collection of 216 colors
- Display the most similar on the Mac and PC platforms
- Hex values:
  00, 33, 66, 99, CC, FF
- Color Chart : http://webdevbasics.net/color

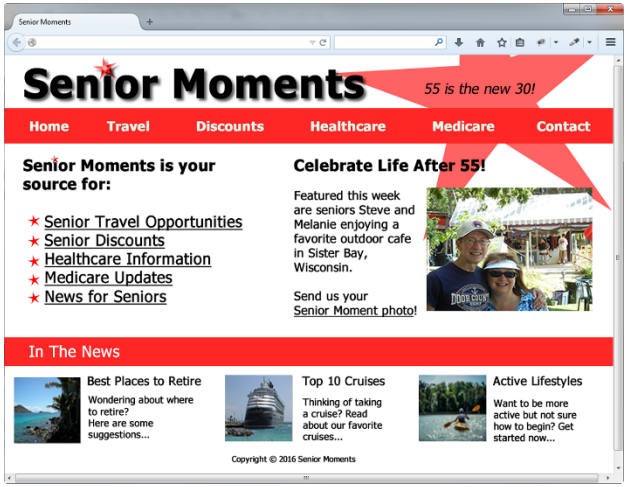
Appealing to Kids & Preteens



Appealing to Everyone

Use of Color

Appealing to Young Adults

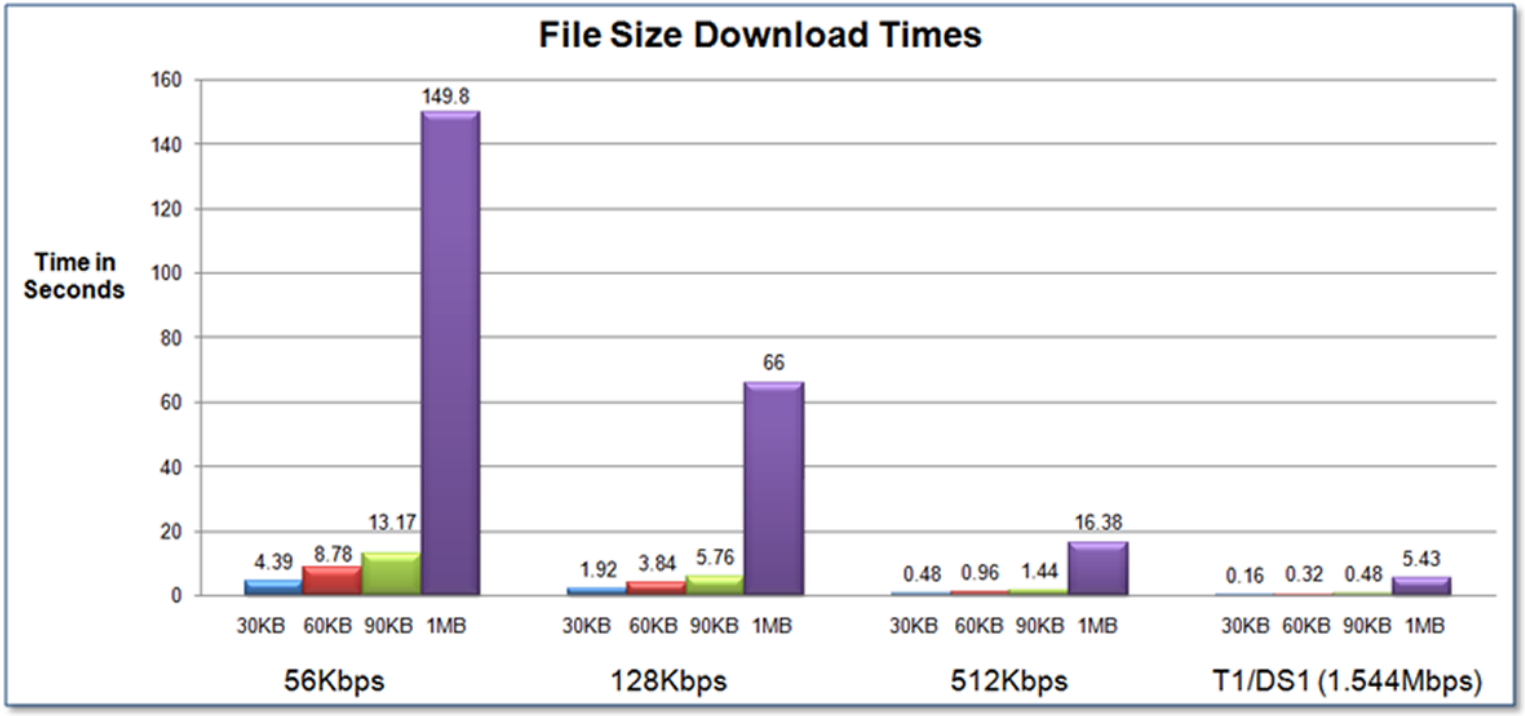


Appealing to Older Adults

Use of Graphics & Multimedia

- File size and dimension matter
- Provide for robust navigation
- Antialiased/aliased text considerations
- Provide alternate text
- Use only necessary multimedia

Web Page Design Load Time



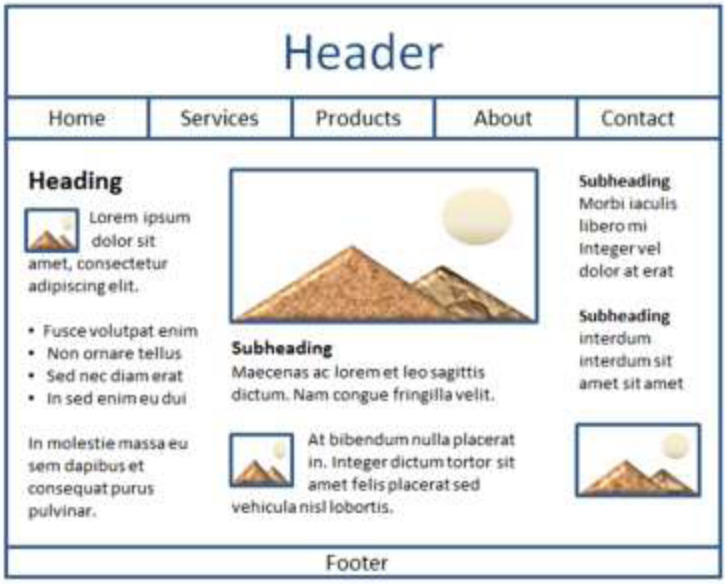
- Watch the load time of your pages
- Try to limit web page document and associated media to under 60K on the home page

Navigation Design

- Make your site easy to navigate
  - Provide clearly labeled navigation in the same location on each page
  - Most common – across top or down left side
- Consider:
  - Navigation Bars
  - Breadcrumb Navigation
  - Using Graphics for Navigation
  - Dynamic Navigation
  - Site Map
  - Site Search Feature

Frame

- A sketch of blueprint of a web page
- Shows the structure of the basic page elements, including:
  - Logo
  - Navigation
  - Content
  - Footer



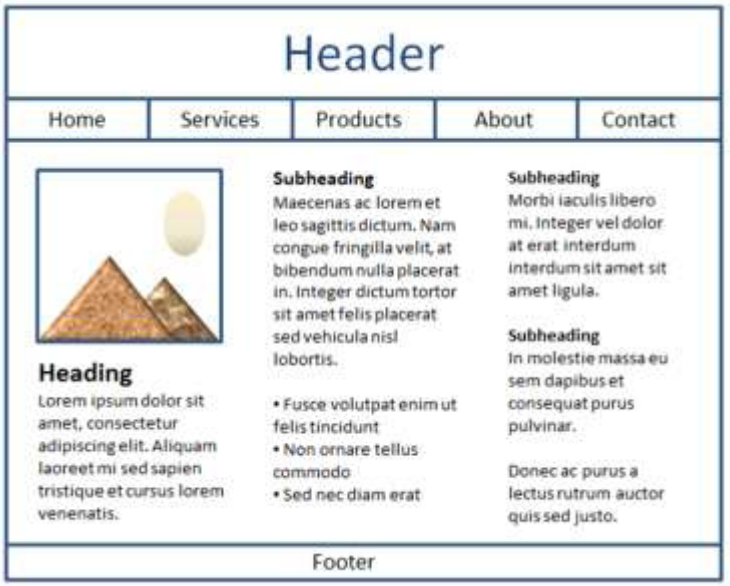
Page Layout (1)

- Place the most important information "above the fold"
- Use adequate "white" or blank space
- Use an interesting page layout

This is usable, but a little boring. See the next slide for improvements in page layout.

Page Layout (2)

Better

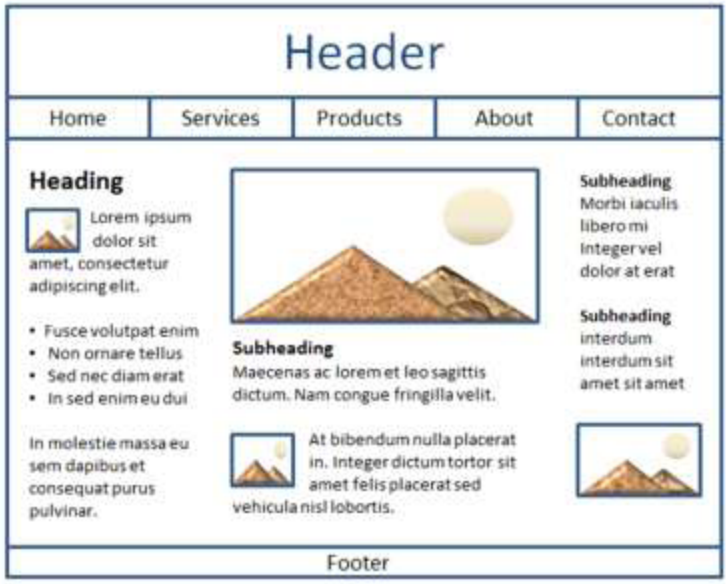


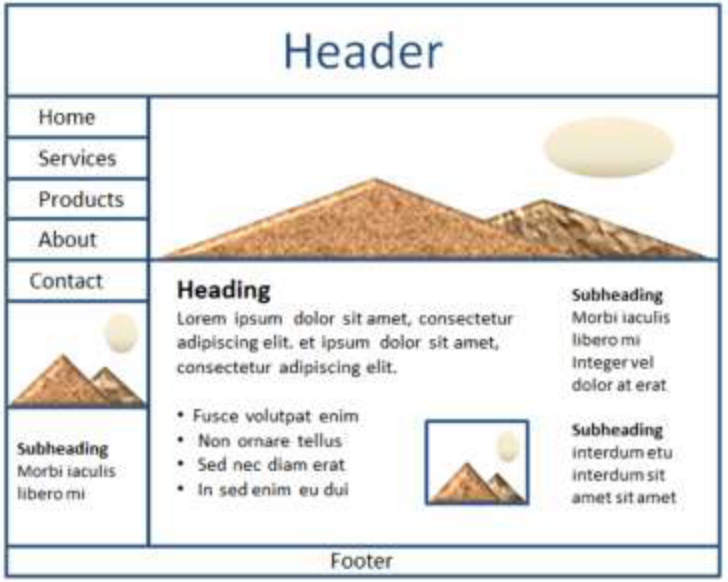
Columns make the page more interesting and it's easier to read this way.

Page Layout (3)

Best

Columns of different widths interspersed with graphics and headings create the most interesting, easy to read page.

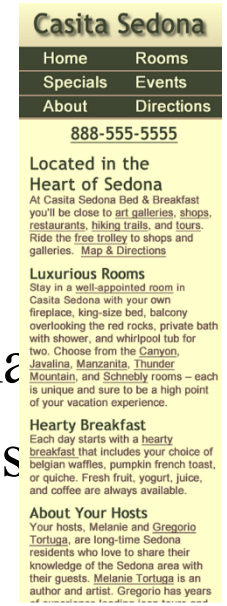




Design for the Mobile Web

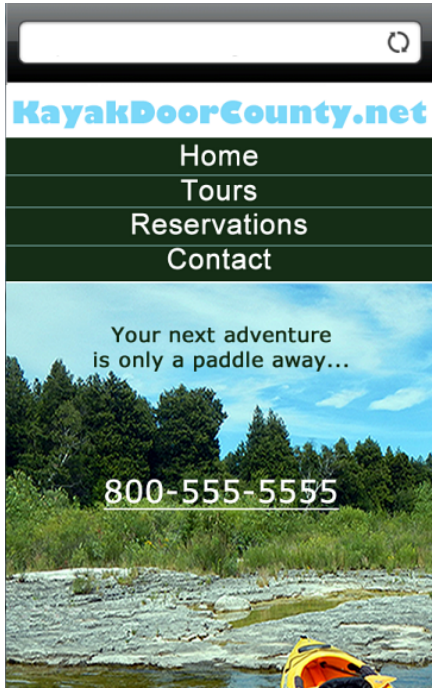
- More users will access websites using mobile devices than with desktop computers

- Three Approaches:
  - Separate .mobi mobile site
  - Host the mobile site within your current doma
  - Configure your current website for mobile dis using responsive web design techniques



Mobile Design Quick Checklist

- Small screen size
- Bandwidth issues
- Single-column layout
- Maximize contrast
- Optimize images for mobile display
- Descriptive alternate text for images
- Avoid display of non-essential content



Responsive Web Design

- Progressively enhancing a web page for different viewing contexts (such as smartphones and tablets) through the use of coding techniques, including flexible layouts and media queries.

Electronics Commerce

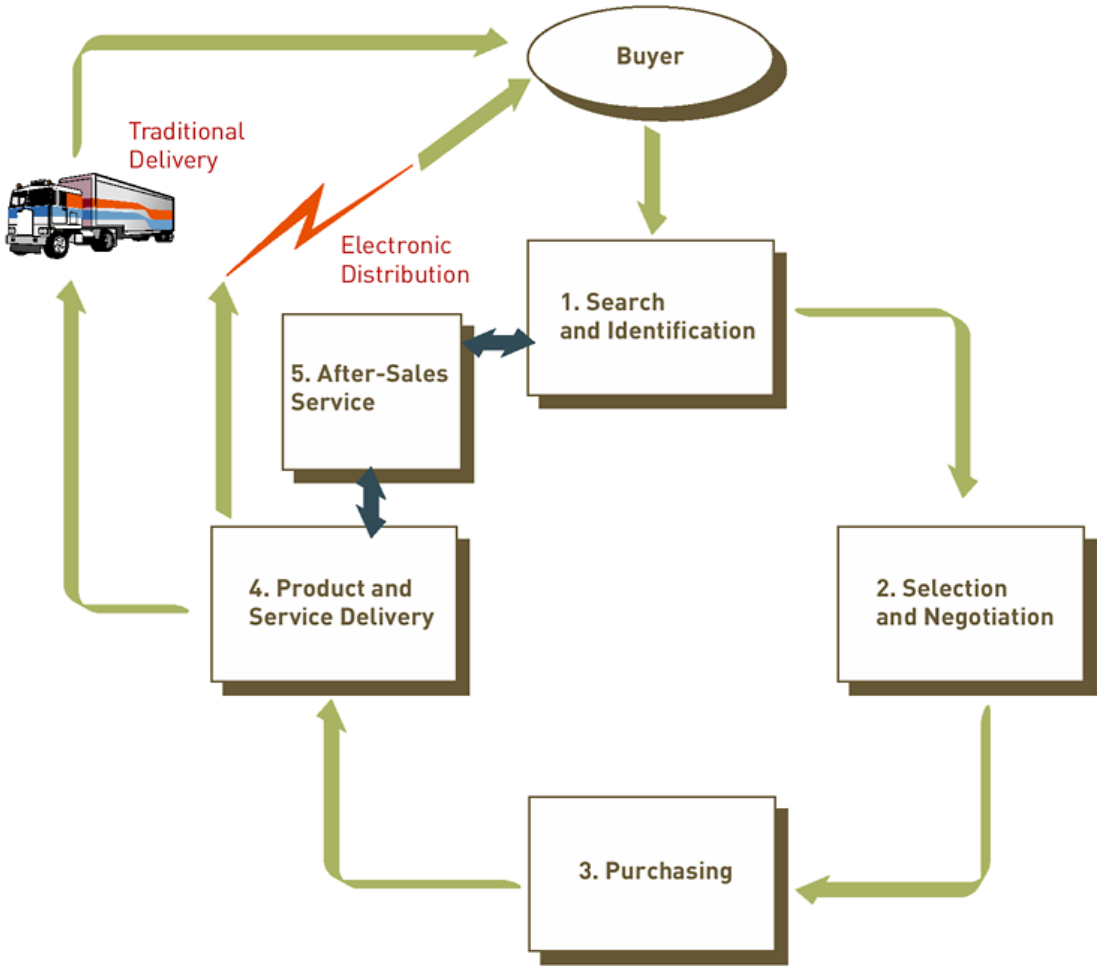
Multistage Model for E-commerce

- Search and identification
- Selection and negotiation
- Purchasing products and services electronically
- Product and service delivery
- After-sales service

An Introduction to Electronic Commerce

- **Business-to-consumer (B2C) e-commerce:** customers deal directly with the organization, avoiding any intermediaries
- **Business-to-business (B2B) e-commerce:** participants are organizations
- **Consumer-to-consumer (C2C) e-commerce:** participants are individuals, with one serving as the buyer and the other as the seller

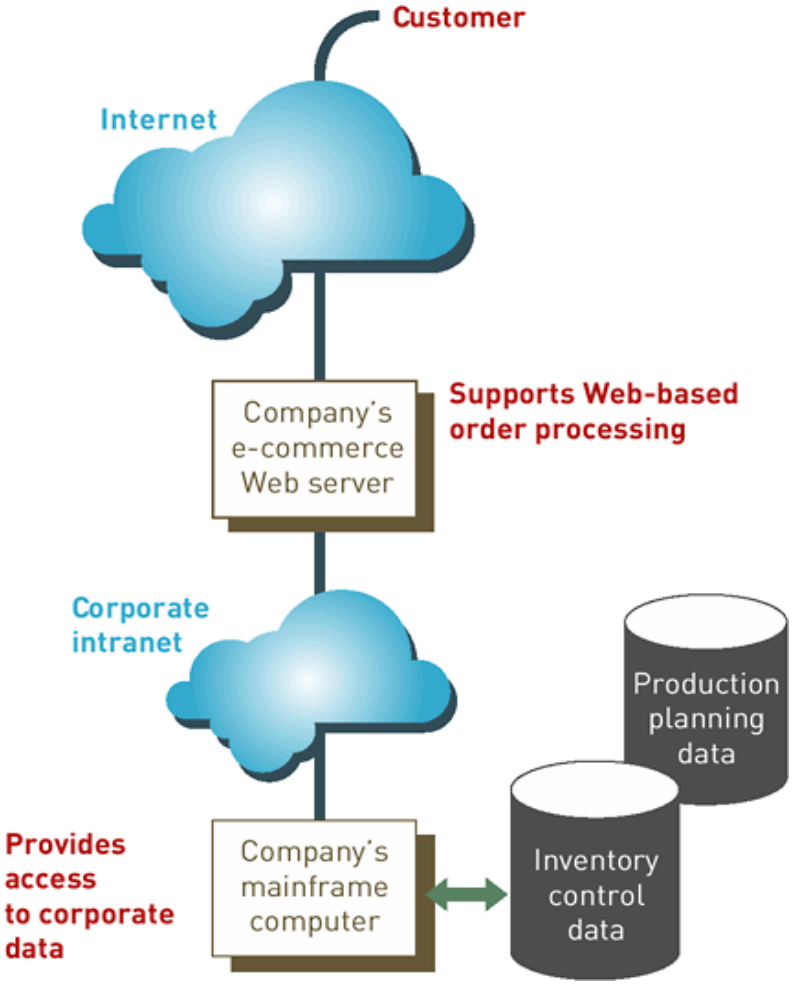
Multistage Model for E-Commerce (B2B and B2C)



E-Commerce Challenges

- Define an effective e-commerce model and strategy
- Need to change distribution systems and work processes
- Integrate Web-based order processing with traditional systems

Web-Based Order Processing Must Be Linked to Traditional Back-End Systems



The E-Commerce Supply Chain

- Supply chain management is a key value chain composed of:
  - Demand planning
  - Supply planning
  - Demand fulfillment

Supply Chain Management

The E-Commerce Supply Chain (continued)

- E-commerce supply chain management allows businesses an opportunity to achieve:
  - Increased revenues and decreased costs
  - Improved customer satisfaction
  - Inventory reduction across the supply chain

Business-to-Business (B2B) E-Commerce

- Allows manufacturers to buy at a low cost worldwide
- Enterprises can sell to a global market
- Offers great promise for developing countries

Business-to-Consumer (B2C) E-Commerce

- Convenience
- Many goods and services are cheaper when purchased via the Web
- Comparison shopping
- **Disintermediation:** elimination of intermediate organizations between the producer and the consumer

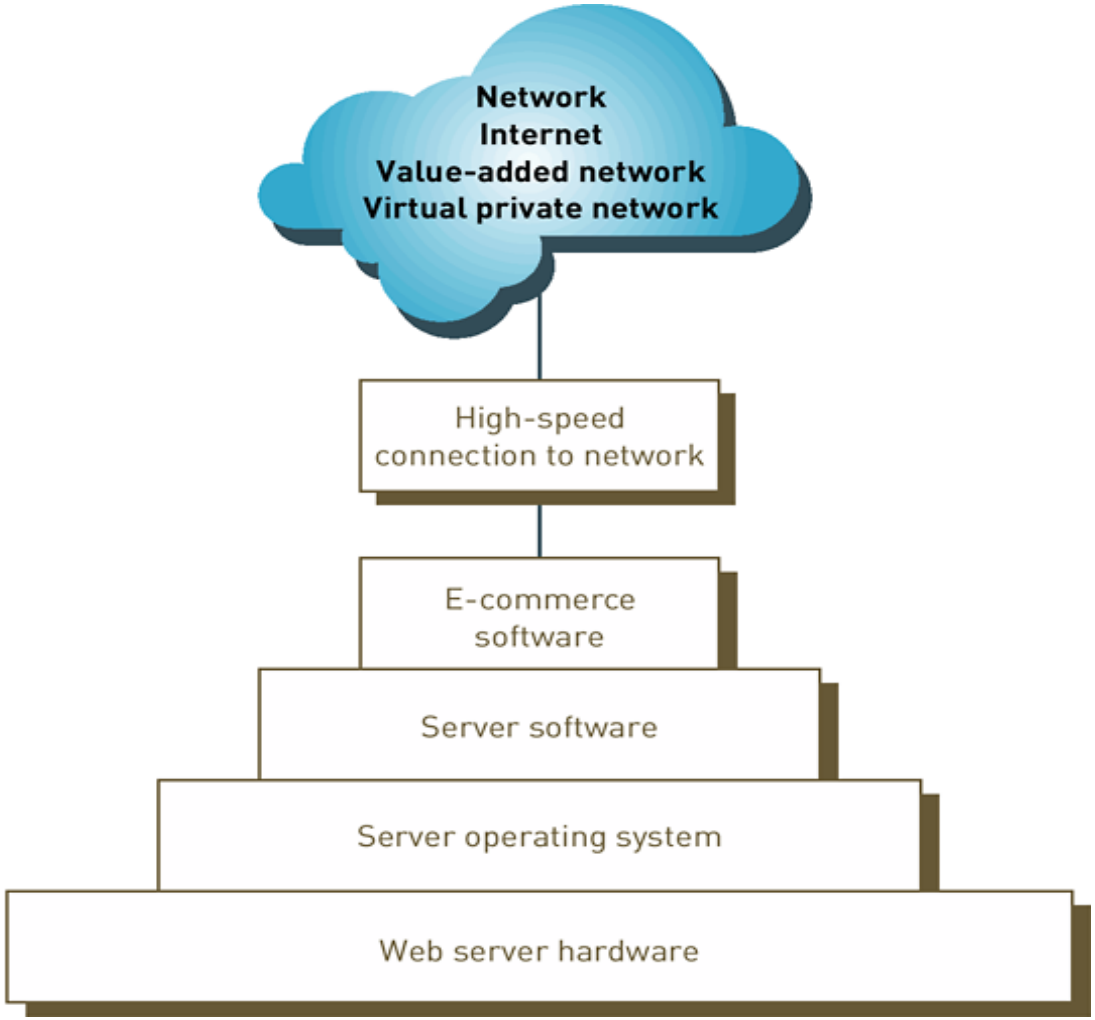
Consumer-to-Consumer (C2C) E-Commerce

- Often done through Web auction sites such as eBay
- Growth of C2C is responsible for reducing the use of the classified pages of newspapers to advertise and sell personal items

Mobile Commerce

- Mobile commerce (m-commerce) relies on the use of wireless devices, such as personal digital assistants, cell phones, and smart phones, to place orders and conduct business
- Issues confronting m-commerce
  - User-friendliness of the wireless device
  - Network speed
  - Security

Key Technology Infrastructure Components



Hardware

- Storage capacity and computing power required of the Web server depends on:
  - Software that will run on the server
  - Volume of e-commerce transactions
- Website hosting

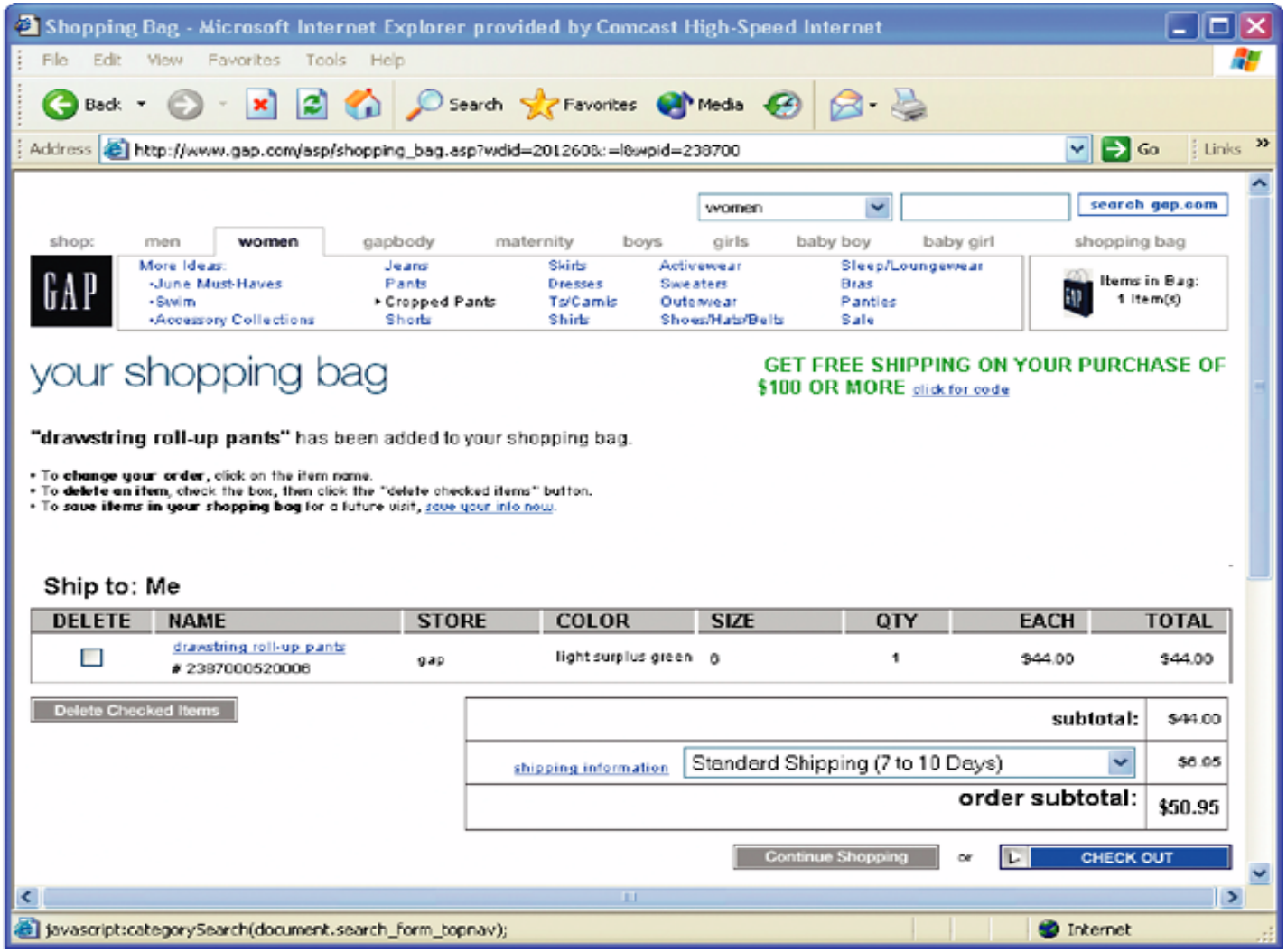
Web Server Software

- Security and identification
  - Web sites must be designed to protect against attacks
  - Denial-of-service (DOS) attack
- Retrieving and sending Web pages
- Web site tracking

E-Commerce Software

- Catalog management
- Product configuration
- Shopping cart facilities
- E-commerce transaction processing
- Web traffic data analysis

Electronic Shopping Cart



E-Commerce Transaction Processing

- **E-commerce transaction processing software:** connects participants in the e-commerce economy and enables communication between trading partners, regardless of their technical infrastructure
- Fully automates transaction processes from order placement to reconciliation
- **Web site traffic data analysis software:** processes and analyzes data from the Web log file to provide useful information to improve Web site performance

Component of Electronic Payment Systems

- **Digital certificate:** an attachment to an e-mail message or data embedded in a Web page that verifies the identity of a sender or a Web site
- **Certificate authority (CA):** a trusted third party that issues digital certificates
- **Secure Sockets Layer (SSL):** a communications protocol used to secure sensitive data
- **Electronic cash:** an amount of money that is computerized, stored, and used as cash for e-commerce transactions

Electronic Payment Systems (continued)

- **Electronic wallet:** a computerized stored value that holds credit card information, electronic cash, owner identification, and address information
- Credit card
- Charge card
- Debit card
- Smart card

Threats to E-Commerce

- Theft of intellectual property
  - **Intellectual property:** music, books, inventions, paintings, and other special items protected by patents, copyrights, or trademarks
  - Patents on business processes

Fraud

- **Phishing:** bogus messages purportedly from a legitimate institution to pry personal information from customers by convincing them to go to a “spoof” Web site
- Online auction fraud
- **Spam:** e-mail sent to a wide range of people and Usenet groups indiscriminately

Search Engine Optimization (SEO)

Definition of Search Engine Optimization

- "Natural," or "organic," search engine optimization (SEO) is designing, writing, and HTML-coding a Web site to maximize the chance its pages will appear at the top of spider-based search engine results for selected keywords and phrases

- Organic Listings**:** Listings that search engines do not sell (unlike paid listings)

Why is it important?

- Search ranking →more site visitors
- Internet users tend not to click through
- Depends on webs role in your economic model

Search Engine Operations

1. Gather Content
   - Crawler or spider moves recursively downloading content
2. Builds sophisticate index
3. Individual web searches run against index
   - Results are retrieved and ordered
     - PageRank & Relevance

SEO Optimization

SEO Optimization Categories

- Keywords
  - Keyword selection and keyword-rich text
- Crawler
  - A crawler-friendly site navigation scheme
- Links
  - Link popularity

Keyword Recommendations

- Page title: visible HTML text ,“Above the fold”
- Page Size: "100 KB" limit is still is still widely accepted. The optimum page size is 500-3000 words (or 2000 to 20,000 characters)
- Be specific
- Example: “Apple iPod” verses “iPod”
  - exact phrase should appear generously throughout your site copy on every page

More Keyword Recommendations

Meta tags: use but don't stuff

- <meta name="description" content="Free Web tutorials on HTML, CSS, XML, and XHTML">

Alt tags: use for graphics

- <IMG src="star.gif" alt="star logo">

Content is king

- Write good content with relevant and important keywords in mind.

Geo Targeting

- Add geocentric terms to target local areas

Domain Names

- Use keywords as part of domain name

Suggestions to be Crawler Friendly

- Traditional <a href> tag
- Keywords in subfolder names
- Minimize quantity of subfolders
- Cross link relevant terms and phrases within the site
- Multiple paths to pages to eliminate orphans

Search Engine tools

- Google
- Yahoo
- MSN Search

INTRODUCTION TO WEB SERVICES

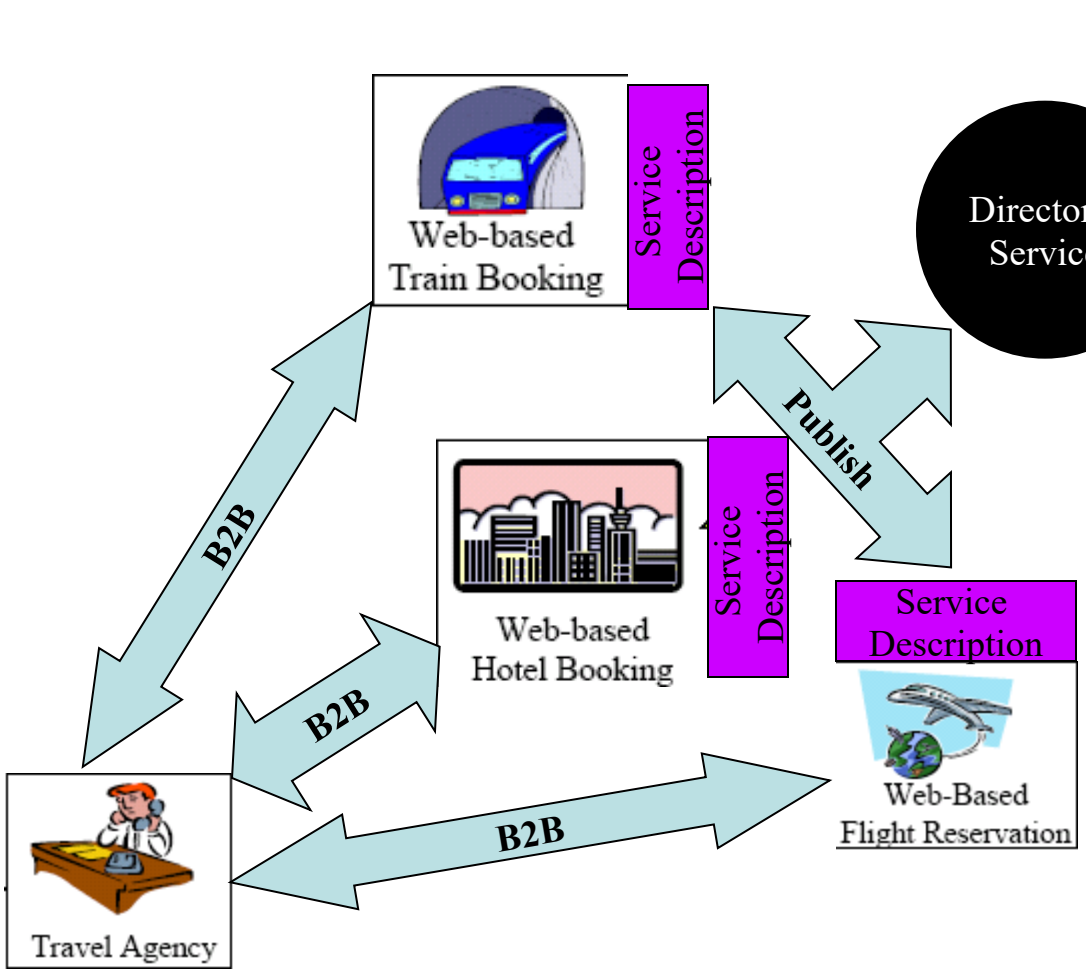
What is a Web Service ?

Web service is a means by which computers talk to each other over the web using HTTP and other universally supported protocols.

A Web service is an application that:

- Runs on a Web server
- Exposes Web methods to interested callers
- Listens for HTTP requests representing commands to invoke Web methods
- Executes Web methods and returns the results

Features Of Web Services



Migration Of Existing web Application

Standards are the Base

Communicate using messages

Robust , Effective, Flexible Security

Services Description

Web Service Discovery

Figure Taken from Intelligent agent course 2003/2004 by Dr. Omar Belakhdar

Web Services Architecture

- Web Services involve three major points
  - Service Provider
  - Service Registry
  - Service Consumer

- Three major operations surround web services
  - Publishing – making a service available
  - Finding – locating web services
  - Binding – using web services

Web Services is based on:

- HTTP (Hypertext Transport Protocol)
- SOAP (Simple Object Access Protocol)
- UDDI (Universal Description, Discovery and Integration)
- WS-POLICY (Web Services Policy) /WSDL

Most Web services expect their Web methods to be invoked using HTTP requests containing SOAP messages. SOAP is an XML-based vocabulary for performing remote procedure calls using HTTP and other protocols.

SOAP

- Actually used to communicate with the Web Service
- Both the request and the response are SOAP messages
- The body of the message (whose grammar is defined by the WSDL) is contained within a SOAP “envelope”
- “Binds” the client to the web service

UDDI

- UDDI is used to register and look up services with a central registry
- Service Providers can publish information about their business and the services that they offer
- Service consumers can look up services that are available by
  - Business
  - Service category
  - Specific service

WSDL

- Describes the Web Service and defines the functions that are exposed in the Web Service
- Defines the XML grammar to be used in the messages
  - Uses the W3C Schema language

What is AJAX?

- AJAX stands for Asynchronous JavaScript and XML.
- AJAX is not a new programming language, but a technique for creating better, faster, and more interactive web applications.
- AJAX is a browser technology independent of web server software.
- AJAX is based on the following web standards:
  - **JavaScript**
  - **XML**
  - **HTML**
  - **CSS**

What is AJAX?

- With AJAX, your JavaScript can communicate directly with the server, using the JavaScript XMLHttpRequest object. With this object, your JavaScript can trade data with a web server, without reloading the page.

- AJAX uses asynchronous data transfer (HTTP requests) between the browser and the web server, allowing web pages to request small bits of information from the server instead of whole pages.

- The AJAX technique makes Internet applications smaller, faster and more user-friendly.

Thank You !

Best Wishes for Examination.