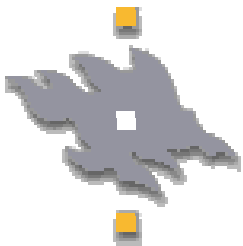
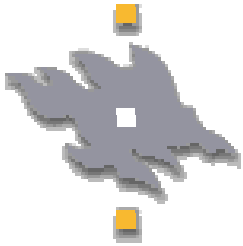
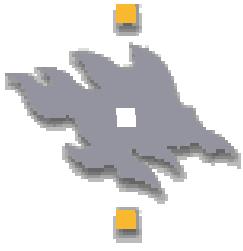


- 
- Web Evolution
 - Emergence of Web Services
 - Brief Overview
 - URI, URN, URL
 - XML Overview
 - XML Namespace



Web Evolution

Suresh Chande



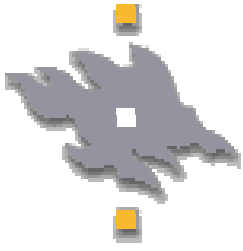
History

- Vannervar wrote an article **in 1945** about a device called Memex (photo-electrical-mechanical device) which could make and follow links between documents
- Doug Engel prototyped an oNLine System(NLS) which enabled hypertext browsing editing, email. Ted Nelson coined the word Hypertext **In 1960's**.
- **1980:** Tim Berner's lee wrote a notebook program "Enquire-Within-Upon-Everything"
- **1990:** World Wide Web was termed for a program which developed a GUI browser for Distributed Hypertext System
- First international WWW conference was held at CERN Geneva **in 1994** and already the thought of Semantics was discussed as the future developments of WWW

Further reading:

History of Internet & WWW: <http://www.netvalley.com/intval1.html>

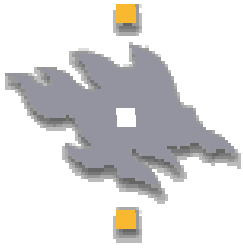
A little History of WWW: <http://www.w3.org/History.html>



Goals of the Web

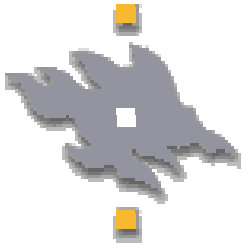
- Web is a universe of network-accessible information available through a network accessible device
- The potential of it being :
 - A means of human-to-human communication,
 - A space in which software agents can, through the access to a vast amount of information and services within information society, science and its problems, become handy tools for us.

[Tim berner's Lee 1998]



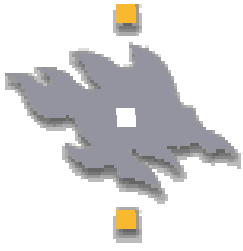
Principles of the Web

- Simplicity
- Modularity
- Tolerance
- Decentralization
- Inter-operable
 - Solution Inter-working
 - Web Language Interworking



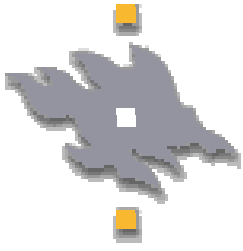
Web Fundamentals

- **URI:** Anything and absolutely anything can be identified by a compact string of characters is the simpler aspect of the web, while being fundamentally the core of the Web.
 - Web is a universal information space in which anything available is addressable.
 - All URI's should be considered as Nouns and not Verbs
 - URI schemes using which resources can be accessed, HTTP, NNTP, SMTP being the schemes



Web Fundamentals [Contd..]

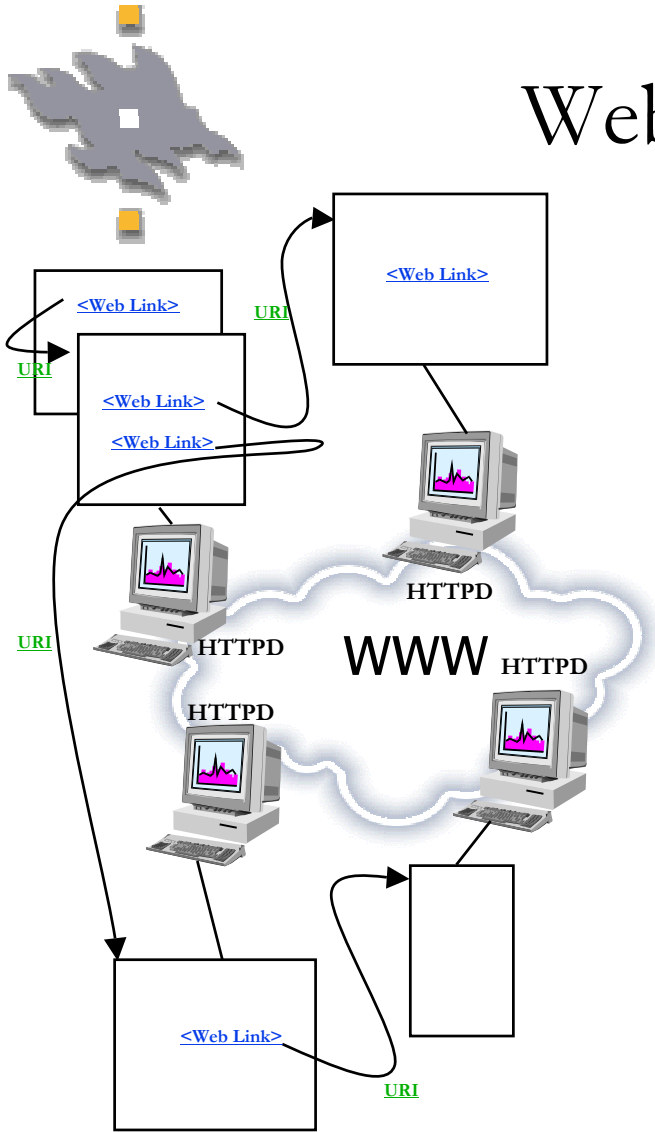
- **HTTP:** Is the URI scheme used over the web to exchange web pages
 - Consists of two parts which are heirarchically delegated
 - One which is for DNS usage
 - The second for the domain name owner



Web Fundamentals [Contd..]

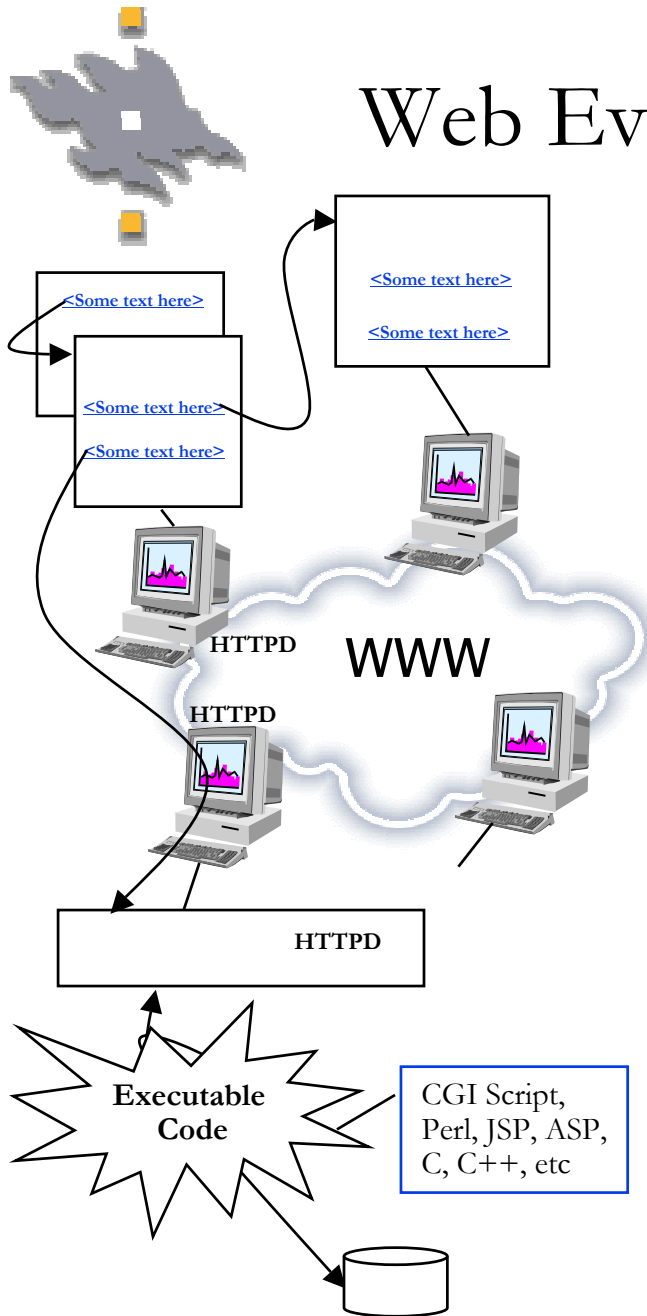
- Remote Operations
 - **Content Level:** HTTP Methods provide side effects over the simple URI's
 - **Objects Level:** Addressing the RPC world which CORBA, DCOM and RMI world require to be satisfied from the Web World.
 - The Web services begun as approaches in this space but move most operational aspects into the message.

Web Evolution – The Static Web



- Direct human consumable information accessible using network enabled User Agents
- Hyperlinked world wide information base
- Static Information Accessable over the web protocols
 - Articles and Information resources
 - Corporate\Portal information distribution media
- Technologies:
 - Web browsers
 - HTML, Extended HTML(Mozilla, Netscape, IE) over HTTP
 - HTTPD Servers
 - HTML Content Editors

Web Evolution – The Semi-Dynamic Web

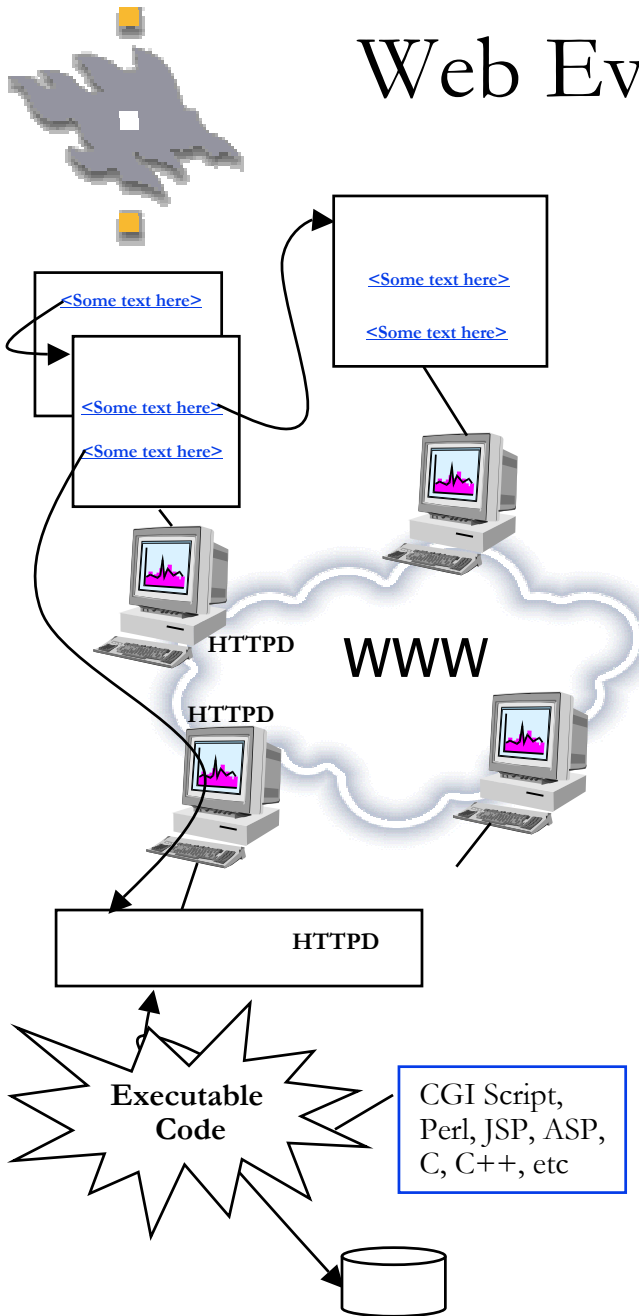


- Rich Media content for human consumption over the web
- Dynamic Information Accessable over web protocols
 - Common Gateway Interfaces
 - Perl, C++, C, Basic, Fortran
 - Web Client/Server processed
 - Applets, JavaScript, ASP, JSP,

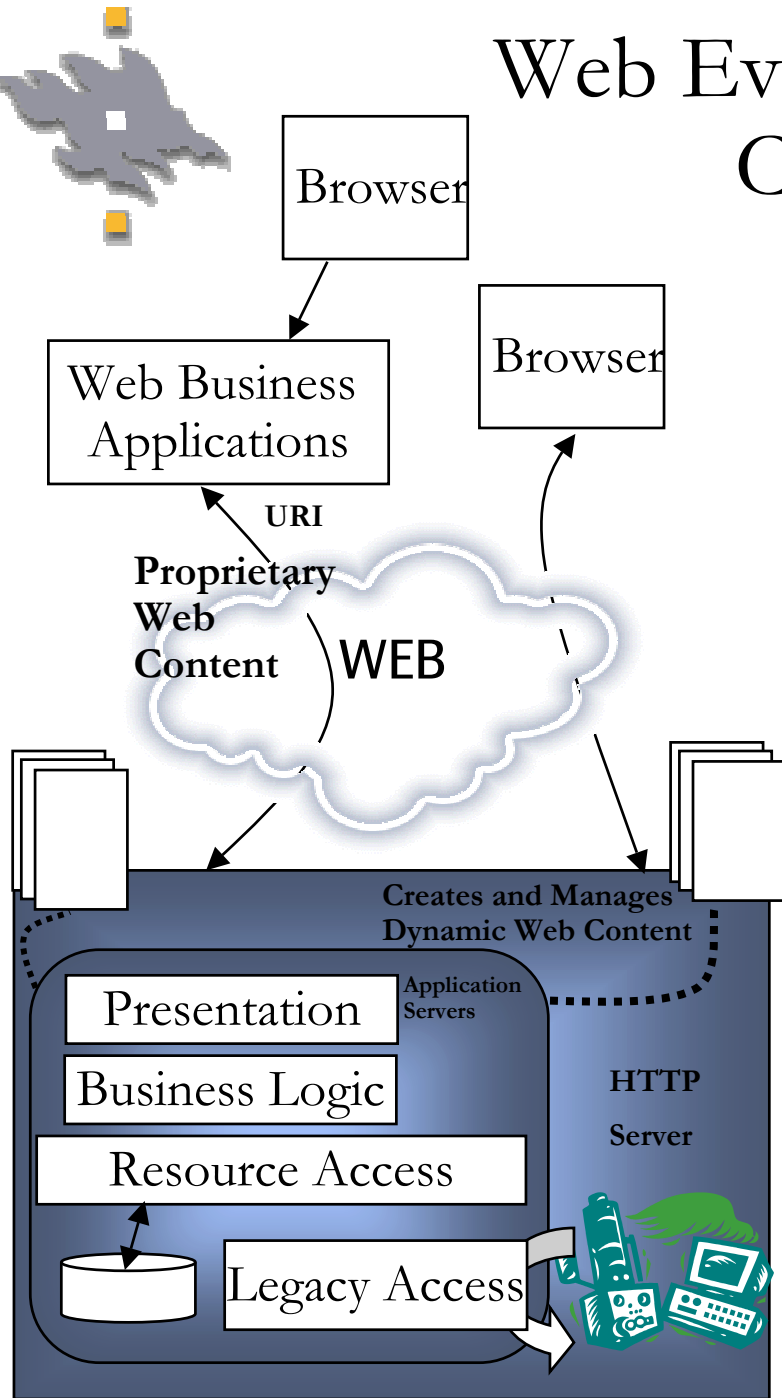
Web Evolution – The Semi-Dynamic Web [Contd..]

Technologies:

- Web Servers with CGI capabilities
 - Hosting and execution of Scripting, interpreted binary code programming Languages
- Web browsers with:
 - Executable Capabilities (Java Applets, etc)
- CGI Application Development tools and Debuggers : IDE



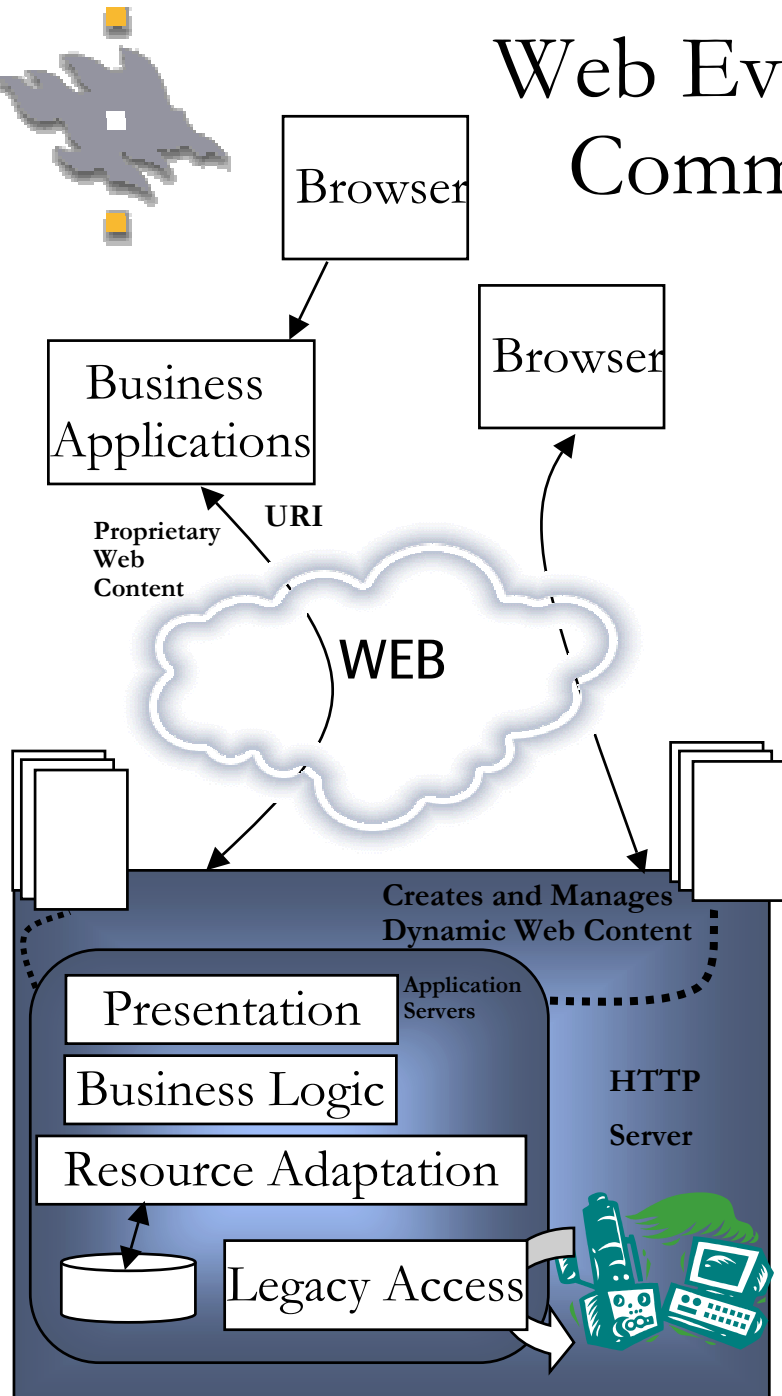
Web Evolution – The Dynamic & Commercialized Web



- The success of Web introduced Web applications and solutions, where web content generated was meant for:

- System to human communications with richer dynamic content
- System to system communications

Web Evolution – The Dynamic & Commercialized Web [Contd..]



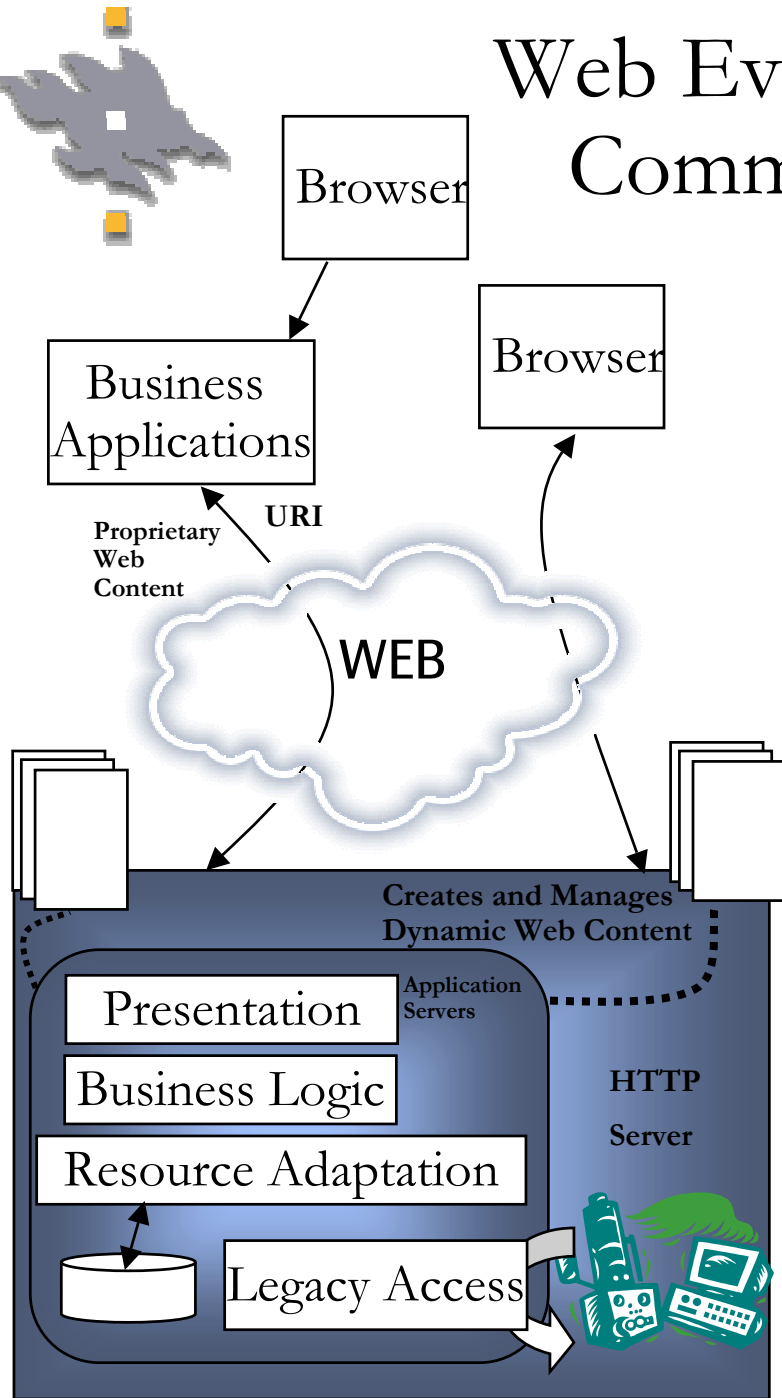
• This led businesses and distributed computing community to embrace the web technologies to solve the various interoperability issues faced by traditional technologies in the internet world:

- CORBA
- DCOM
- RMI

• Approaches of tunneling proprietary protocols over http were attempted:

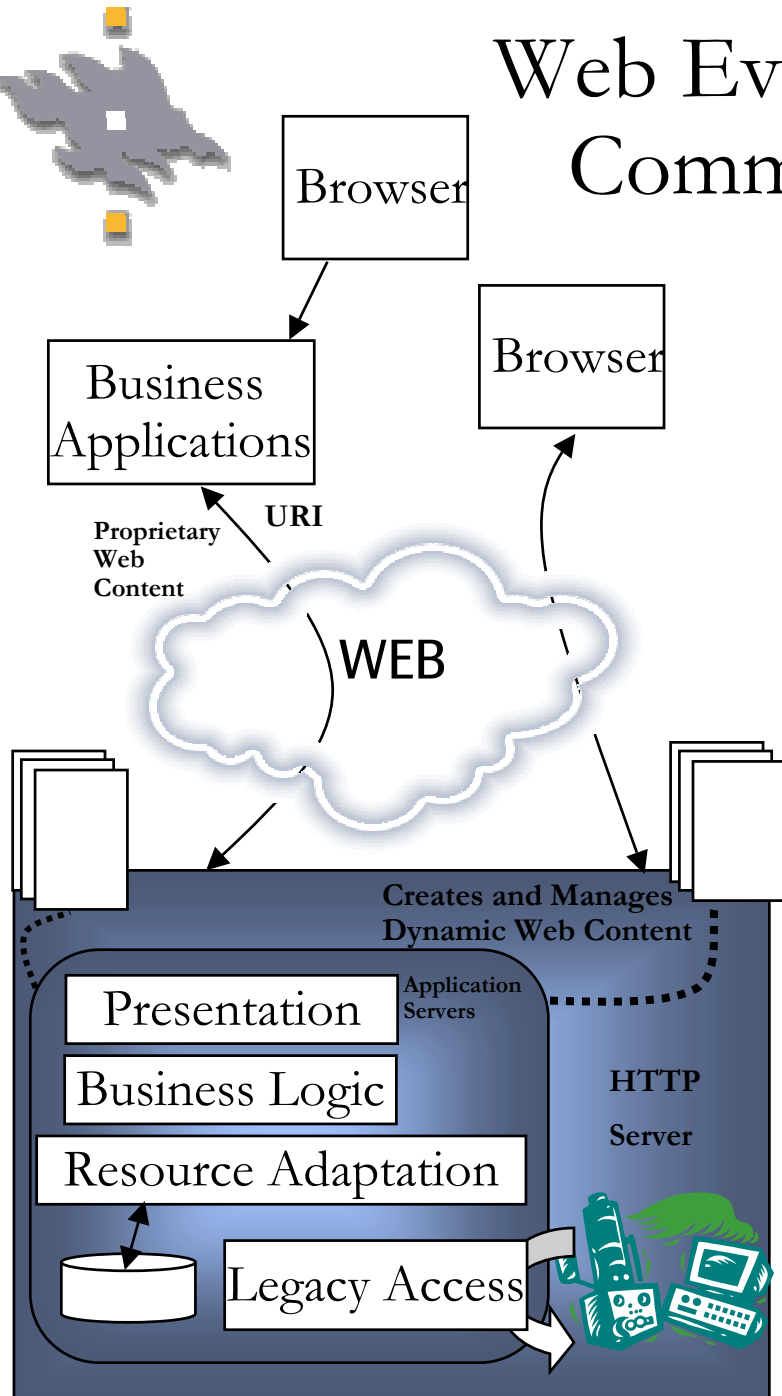
- By pass firewalls
- Access remote objects over the web

Web Evolution – The Dynamic & Commercialized Web [Contd..]



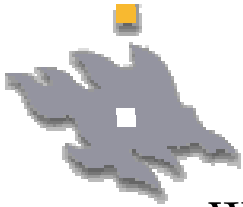
- The ubiquitous access to web content introduced plethora of useragents and ultimately huge explosion web accessible handheld devices.
- This potential outbreak of web clients was supported by deployment of commercial application platforms as the web protocols became more secure and reliable

Web Evolution – The Dynamic & Commercialized Web [Contd..]



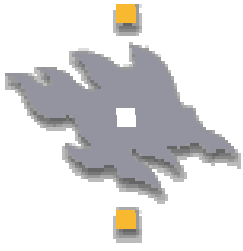
• Commercial applications over the web demanded:

- Security
 - Robust
 - Scalable &
 - Stable systems being Fault tolerant and reliable systems
- These demands led into well deserved and stable platforms (Application Servers)
- J2EE : Servlets, JSPs
 - .NET : ASP, Web Components



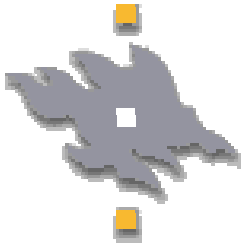
Web Evolution – Information to Data

- Web - A huge source of well presentable information for direct human consumption.
- The content generated is exposed directly to end human users , while
- The input generated by the human user was available in a format suitable to be processed by web systems.
- This was fine until the end user of the web of information/services was required to be a non human which was required to interpret and invoke this information.
- The information available in web required to address two aspects for a non human end user:
 - A well interpretable manner of expressing information (a human language sort of features)
 - An well agreed and simple approach of processing such information (distributing computing paradigms)



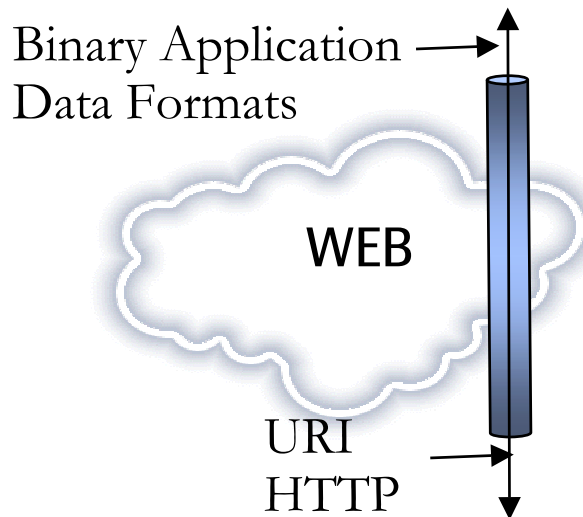
Web Evolution – Information to Data

- These needs were handled by considerably noticeable and inter-related efforts in the web community:
- **Semantic Web:**
 - The idea of having data on the web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications[W3C]
- **Web Services:**
 - An approach to access services using XML based protocols over the Web to find required services, interpret the available service and to invoke such services, able to do all this without a requirement for platform dependencies.



Emergence of Web Services

Web Application Platforms



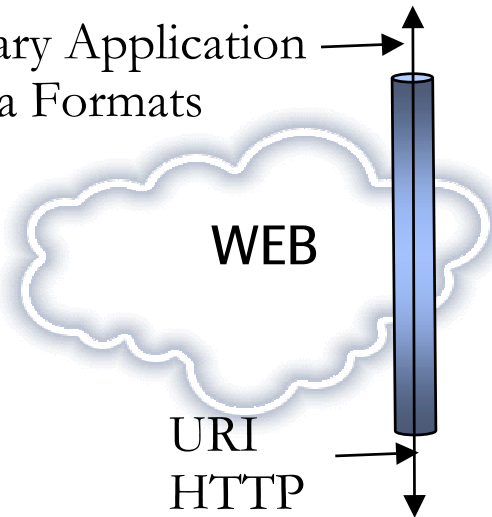
Web Application Platforms

- Movement and evolution of the web from a *human centric web of information* to a *service oriented web system* was meant to serve both humans and non humans. This evolution continued to use of the basic properties of the Web:
 - Ubiquitous access
 - Modular and extensible
 - Flexible and scalable
 - Uniform Addressing mechanism
- Initially network based applications provided access to their functionality over the web by tunnelling platform specific communication protocols(in RPC) in native binary formats over standard web protocols.

Emergence of Web Services [Contd..]

Web Application Platforms

Binary Application
Data Formats



Web Application Platforms

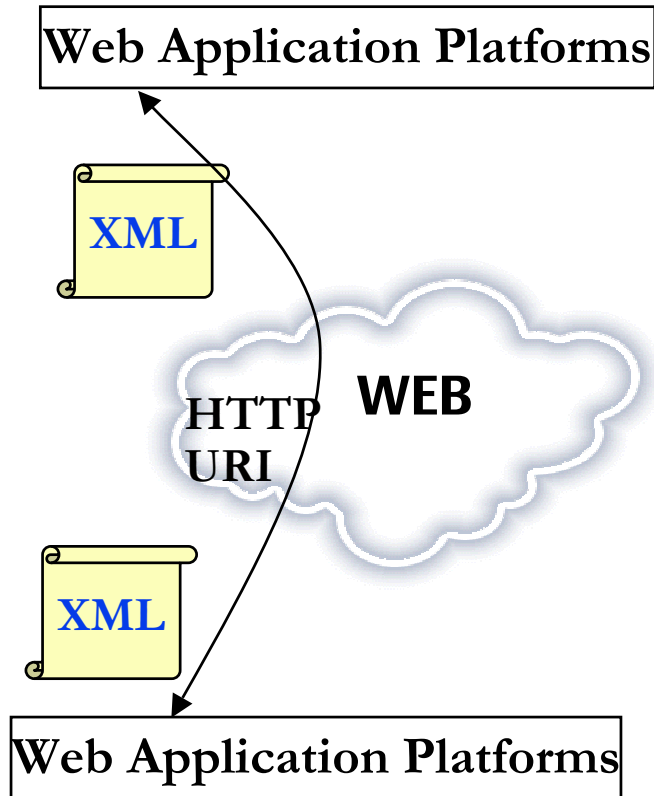
- **Problem:**

- This led to inefficiency and latency as the web was not meant for such applications
- The solution was doomed to fail and not to be taken into wide usage as it leads to huge interoperability problems.

- **Solution:**

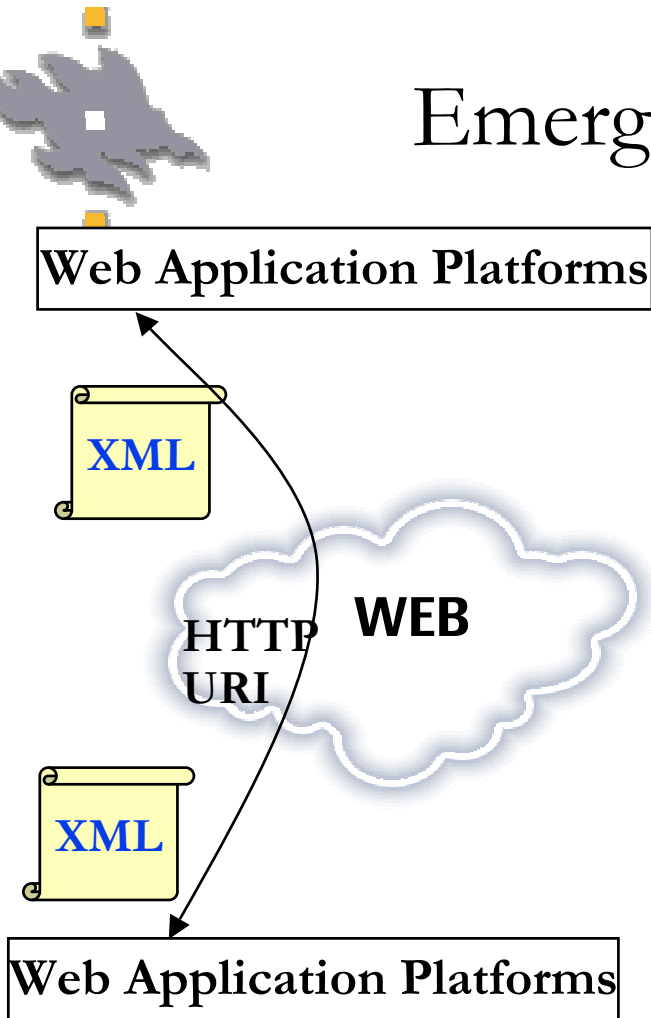
- There was a shift to embrace more web friendly application data format XML, which were both human readable, as well as parsable and interpretable in a platform independent fashion
- XML over the web was applied to bridge the gap between heterogenous application deployment environments.
- The traditional Distributed computing systems were still based on RPC mode of communications. The initial XML based RPC oriented communications were hence introduced (XML-RPC). Department of Computer Science

Emergence of Web Services [Contd..]



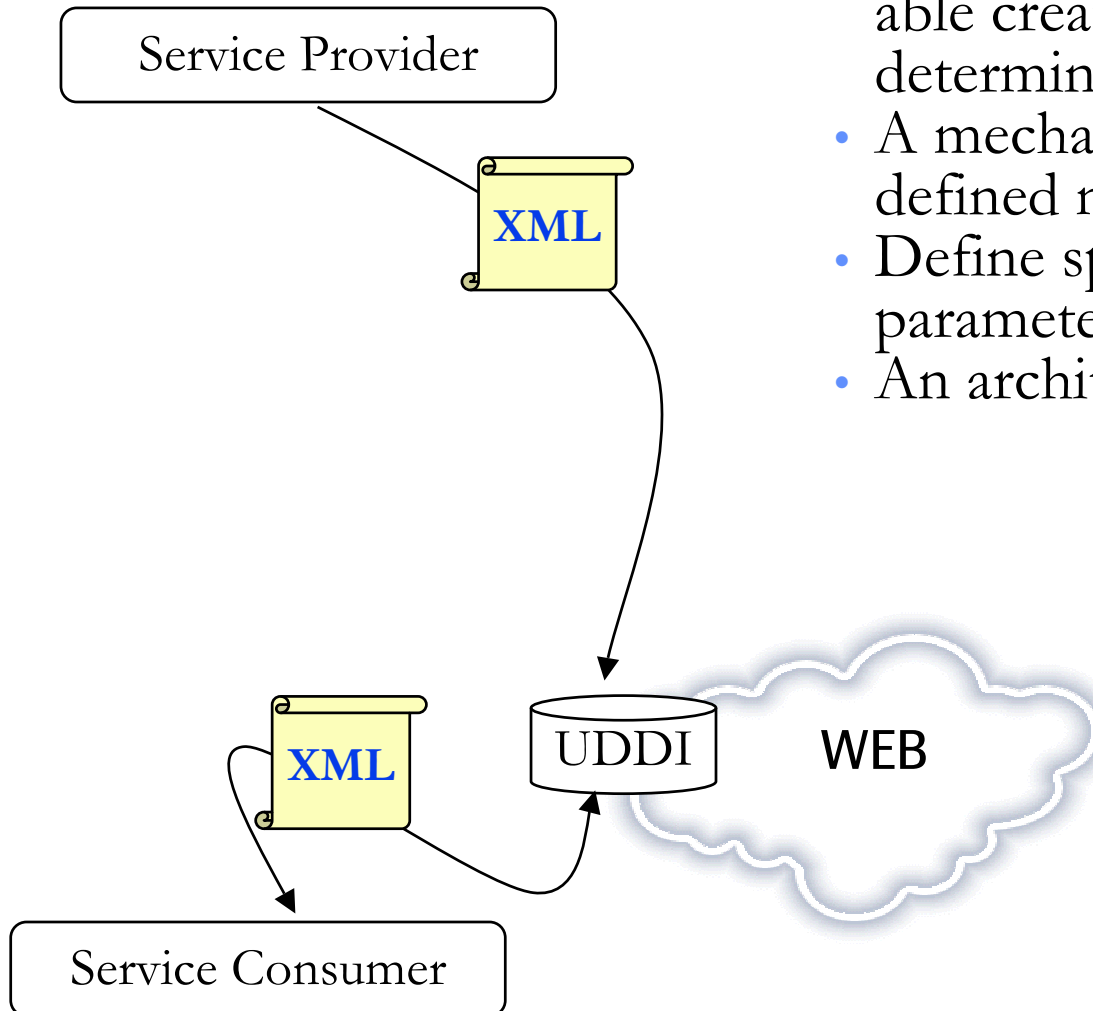
- Inter-operability between heterogeneous deployment platforms lead to seek solutions in a platform neutral manner by using XML as the core technology.
- The non-standard deployment of platform neutral technology solved the inter-operability across heterogeneous platforms
- This still lacked the elegance and conventions to solve the application communication inter-operability issues in a standardized manner

Emergence of Web Services [Contd..]

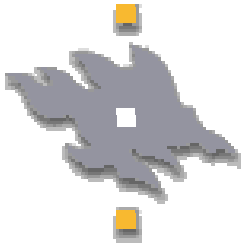


- Few Industry partners and players together in a combined efforts started to sketch a basic service oriented architecture taking XML at the centre of the architecture solution and started defining XML based protocols.
- This later fueled several industry players and standards to agree on a basic framework and a set of protocols to produce inter-operable solutions

Emergence of Web Services [Contd..]

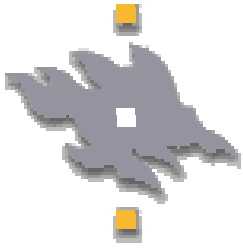


- Framework was developed to :
 - Define message containments in-order to be able create and process messages in a determined manner
 - A mechanism to exchange messages in a defined manner
 - Define specific services, the set of message parameters it can exchange.
 - An architecture which contains
 - » Web service providers being able to define and publish their services
 - » Web services consumer who can find and consume a required service
 - » A Web Services registry which can locate and discover services for the Consumers.



Some Interesting References

- Web design Issues: <http://www.w3.org/DesignIssues/Overview.html>
- A Little History of the World Wide Web : <http://www.w3.org/History.html>
- History of Internet & WWW: <http://www.netvalley.com/intval1.html>



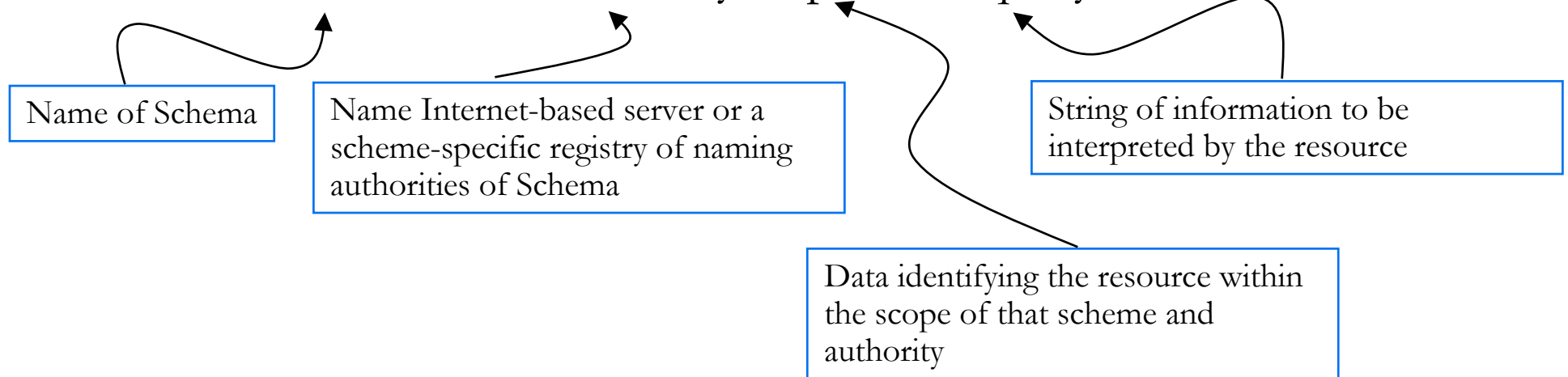
URI, URN's, URL's

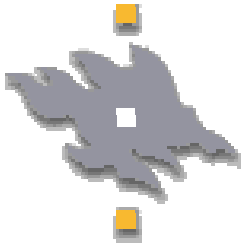
A brief recap of our understanding

URI – A Generic resource identity

- A Uniform Resource Identifier (URI) is a compact string of characters for identifying an abstract or physical resource. It provides a simple and extensible means for identifying a resource.
 - Allows different types of resource identifiers to be used in the same context
 - Uniform means of addressing different resources though they use different access mechanisms
 - Allows introduction of new identifiers and access mechanisms

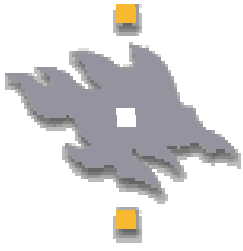
- The generic URI syntax consists of the sequence of four main components:
 - `<scheme>://<authority><path>?<query>`





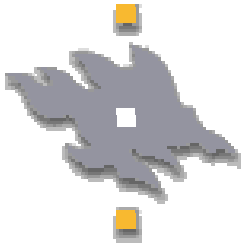
URI, URL, URN- The relation between them

- A URI Schema defines the namespaces of an URI, thus restricting the syntax and semantics of the identifiers using such a namespace.
- Standards, reference documents and even experts use URI/URLs in an interchangeable manner assuming that the community understands the difference. This led to a bit of confusion among IT professionals
- URI's provide a mechanism to define different schemes, For example:
 - Uniform Resource Locator(URL): Is a URI scheme which provides means of locating and accessing a specific resource over the web, for.g: using the “http://”, “gopher://”, “new://”, etc .. scheme



URI, URL, URN- The relation between them

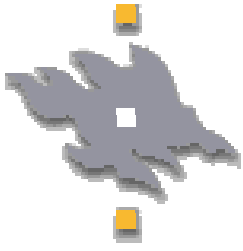
- Uniform Resource Name(URN): Is a subset of URI which provides the unique name globally across the network and persistent even if the resource ceases to exist or becomes unavailable over the network: For e.g: **“urn:isbn:n-nn-nnnnnn-n”**, few other examples : “ietf”, “oasis”, “pin”, “newml”, “xmlorg”
- Uniform Resource Characteristics/Citation(URC): Was proposed to be another subclass of URI which points to metadata about a resource rather than the resource itself, but this was not accepted
- **Further Reading:** About this issue, methods of registration, specifications of new URL/URN Schemas:
 - Web Addressing : <http://www.w3.org/Addressing>
 - URI : <http://www.ietf.org/rfc/rfc2396.txt>
 - URI, URL, URN Clarifications: <http://www.w3.org/TR/uri-clarification/>
 - Guidelines for defining new URL Schemas: <http://www.ietf.org/rfc/rfc2718.txt>
 - Definition of URN namespaces: <http://www.ietf.org/rfc/rfc2611.txt>



XML overview

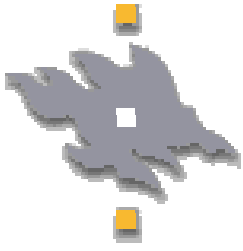
Limitations DTD based XML structuring

Suresh Chande



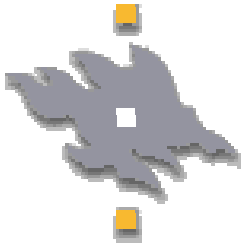
XML – Overview

- XML- > e**X**tensible **M**arkup **L**anguage
- XML is a restricted form of SGML(Standard Generalized Markup Language ISO 8879, 1986). It is a simplified version of SGML developed to better suite light weight architectures such as the Web. It has been derived from SGML.
- XML was originally created in 1996, XML has been W3C's recommendation since 10th February 1998.
- The Whole purpose of XML has been to:
 - To represent documents, information and data in rich structured manner
 - Being able to parse, interpret and manipulate information in a generic manner
 - Human readable and easy to create



XML – Overview

- Exchange of data over internet\web in a markup language that can be parsed and assimilated in a light weight manner and using non proprietary format.
- Being able to rigidly specify the structure of the XML Document(DTD) and being able to validate if a document is compliant with the specified structure(DTD), at the same time not mandating if needed.
- An XML document can be interpreted by a SGML systems as it complies to a subset of the SGML specifications and hence does not break the SGML parsers, applications if required to utilise the XML

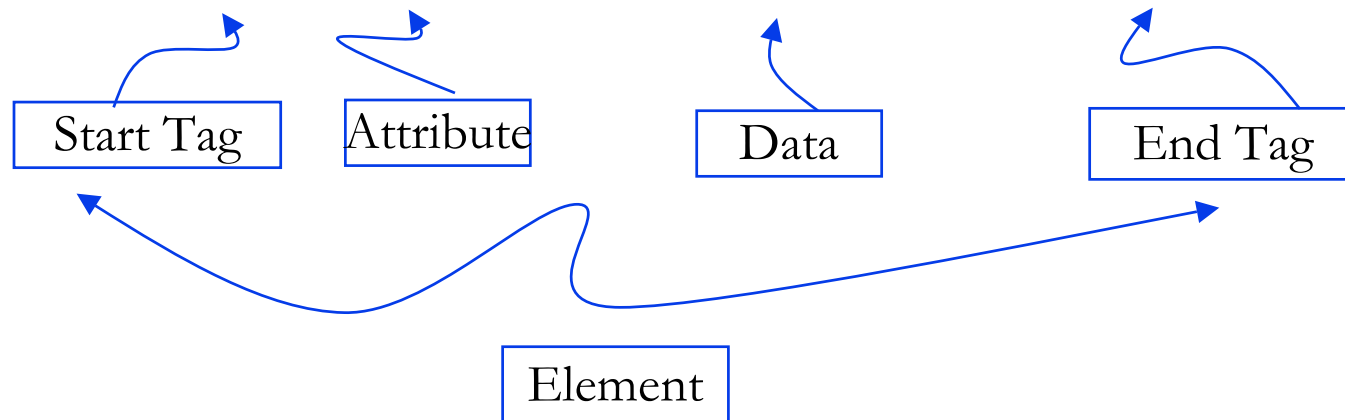


XML – Quick Summary

- XML Documents is made up of 6 different type of markup's that can occur in an XML document,
- **Namely:** Element, Entity Reference, Comments, Processing Instructions, Marked Sections and Document Type declarations

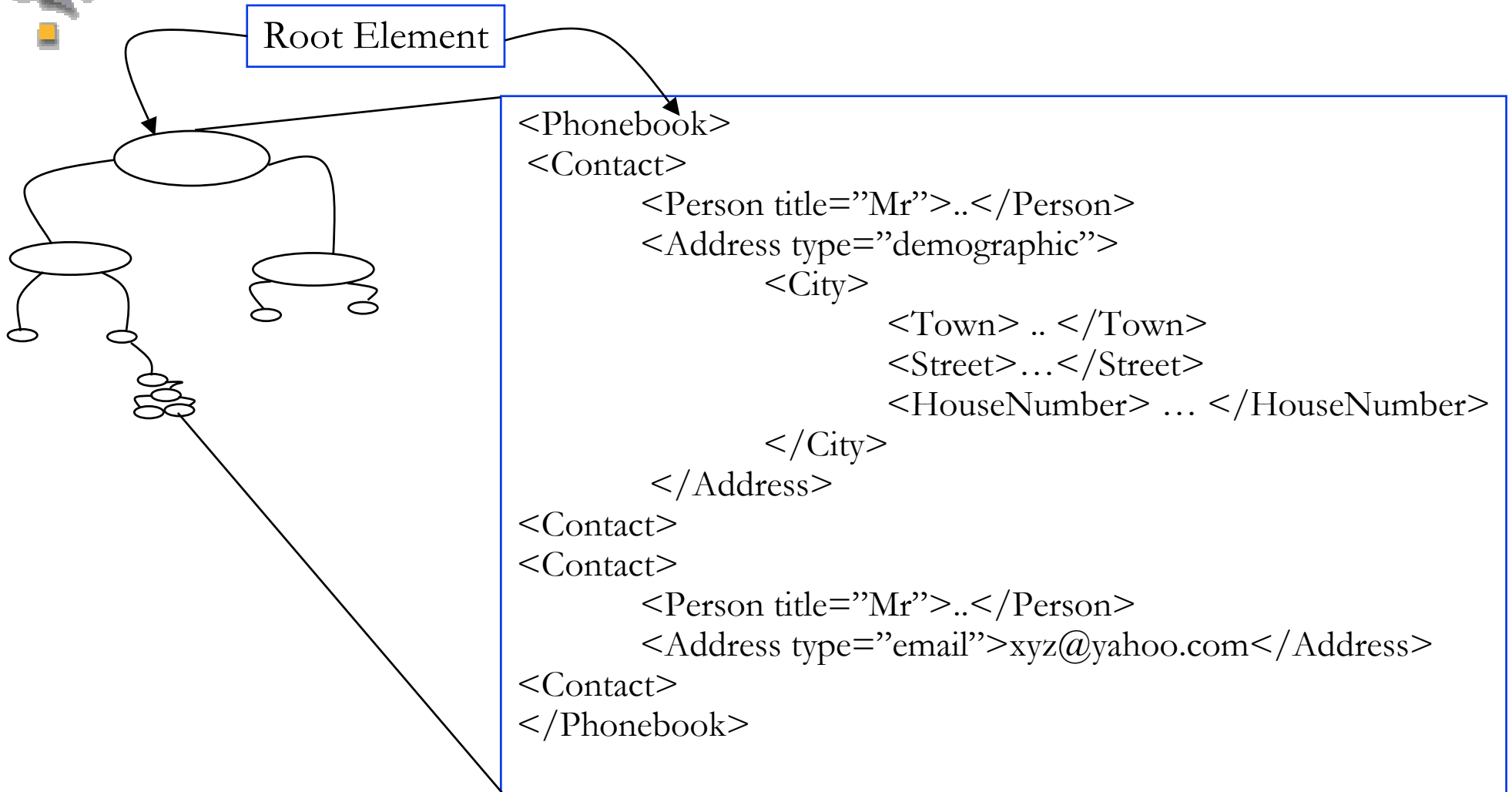
- **Element:**

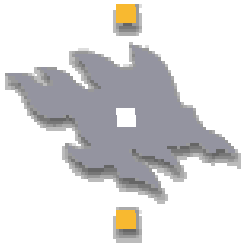
- `<H1 color="red">This is a title</H1>`



XML Document

Root Element





XML – Quick Summary [Contd]

- **Entity References:** To represent XML documents using markup tags, some characters are made reserved, meaning that the documents can not contains such characters in thier normal character form. For the need to represent these characters Entities must be used for such purposes and they begin with an ”&” and end with a ”;”, for example: <, representing ”<”.
- Entities are also used to represent repeated text / content from external files



XML – Quick Summary [Contd..]

- **Comments:** `<!-- Any free text here -->`
- **Processing Instructions** `<?name pidata?>` : These are instructions meant to be passed to the applications for processing. The name is called the PI target is used to identify the PI to the applications.
- **CDATA Section** instructs the parser to ignore all parts that occur in between the start of the CDATA Section , namely: `<![CDATA[" and the end tag"]]>`
- **Document Type Declaration(DTD):** This is used to declare the structure of the document and rigid rules and syntax of such a structured document. This is represented a meta information which is passed onto a parser which can ensure and validate that the XML documents follows such rigid rules, syntax and the defined structure.



XML – Quick Summary [Contd..]

- The DTD consists of declaration of the following four different aspects:
 - **Element Type declaration:** Specifies the Name of the element and specifically the internal structure of an element.
 - For ex: `<!ELEMENT address (name, phone+,(street, city, state?)?, country)>`
 - **Attribute Declaration:** Specifies the attributes a specific Element can have and particularly the name and sort of value that attribute can hold
 - For e.g.: `<!ATTLIST phone mobile CDATA #REQUIRED
state (using|no_used) 'using'>`
 - **Type of Attributes:** CDATA, ID, IDREF, ENTITY/S, NMTOKEN/S
 - **Default VALUES :** #REQUIRED, #IMPLIED, #FIXED, “value”



XML – Quick Summary [Contd..]

- **Entity Declarations:** Used to associate a name with a fragment of document

- For e.g.: `<!ENTITY % localaddress “(street,city,state?)”>`

- `<!ELEMENT address (name,phone+,%localaddress;,country)>`

- Three types of Entity references

- **Internal Entities :** user defined, `<`, `>`, `'`, `&`, `"`;

- **External Entities :** used to include external files into the referenced locations

- **for e.g.:** `<! ENTITY personal_preferences SYSTEM “/userid/1234/profile.xml”>`

- **Parameter Entities:** Used only in DTDs, for associating with fragment of document Type declarations

- **Notation Declarations:** used to identify specific types of external binary data

- For e.g.: `<!NOTATION MPEG7 SYSTEM “mpg”>`



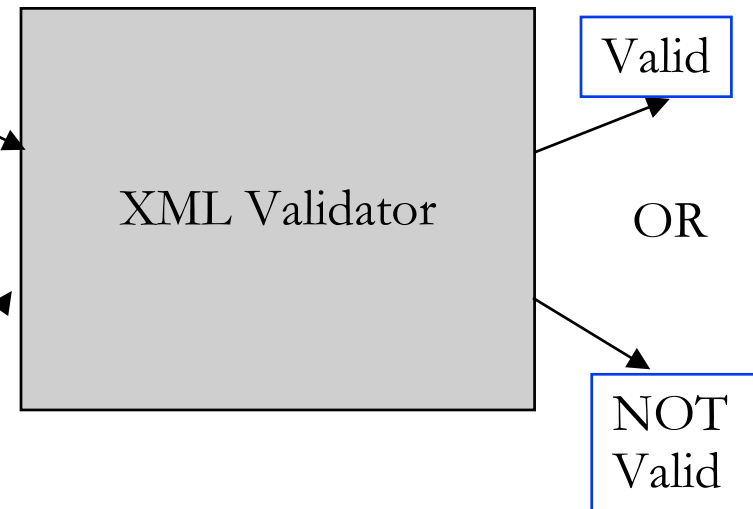
DTD based XML Structure & validations

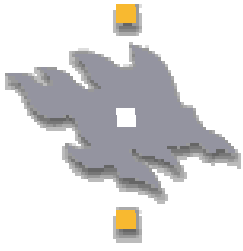
XML Document

```
<?xml version="1.0" ?>
<!DOCTYPE address SYSTEM
    "address.dtd">
<address>
    .....
</address>
```

```
<!ENTITY % localaddress "(street,city,state?)">
<!ELEMENT address
(name,phone+,%localaddress;,country)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT city (#PCDATA)>
.....
<!ELEMENT country (#PCDATA)>
```

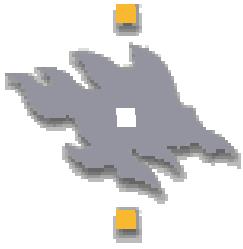
DTD Declarations





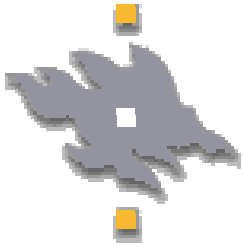
Limitations of DTD

- DTD's were something that was carried on from the SGML world which was made for the purpose of SGML validation. Maybe not the best suitable for XML data which was meant for application level data.
- DTD's are hard to parse compared to XML files and required parsers to utilise alternative means of parsing DTD for validations purposes
- Are not modular enough
- Lack of support for XML namespaces
- Weak Data Typing: Do not provide rich data typing usually required for application level data representations
- Not suitable for data oriented information, but better suitable for text-oriented information



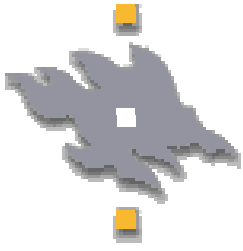
Limitations of DTD [contd..]

- DTDs do not allow complex information models
- DTDs are not namespace aware



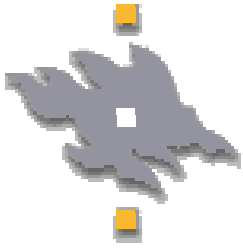
Good Links and References

- <http://www.w3.org/TR/1998/REC-xml-19980210#ISO8879>
- <http://www.zvon.org>
- <http://www.xml.org>



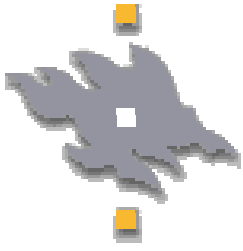
XML Namespaces

A brief recap of our understanding



XML Namespaces

- XML namespaces provide a simple method for qualifying element and attribute names used in Extensible Markup Language(XML) documents by associating them with namespaces identified by URI references[W3C]
- Purpose:
 - Several different application modules process XML documents. They process them based on the knowledge they are encoded with about specific XML document structures.
 - These purpose of the well known XML document structures are better reused than reinventing in the cases where the similar effects are desired.



XML Namespaces

- The XML structures can be reused if the constructs of the document (elements, attributes, etc) are described in a universal and referencable manner.
- XML Namespace addresses this exactly
- [W3C] It is a collection of names, identified by a URI reference [\[RFC2396\]](#), which are used in XML documents as element types and attribute names



Defining an XML Namespace

- The names in the XML Namespaces should appear as :

Prefix: LocalName

- Where "*Prefix*" is a name which refers to a URI, which selects the namespace to which the LocalName belongs to. The "*LocalName*" is the name of the construct which is used within the XML document structure

NOTE: The URI contains characters which are not allowed in the names of the constructs and hence this has to be proxied by a prefix.



Defining an XML Namespace [Contd..]

- The URI's are proxied by the "Prefix" by defining it as shown below:

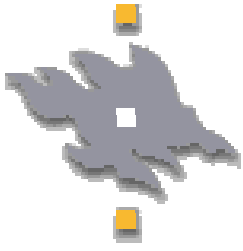
```
<name xmlns:prefix="URI"> .....</name>
```

Where **xmlns** identifies the value as an XML namespace while "Prefix" can be any qualified name. URI refers to a universally unique namespace.

NOTE: It is not necessary that the URI contains the XML Schema which defines the element construct

- The Element as well as the attributes can be Namespace qualified, such as shown in the example below:

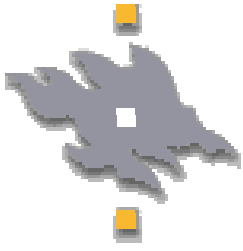
```
<my_message xmlns:m="http://www.mobile.com/mms_schema">  
  <m:to>+358504837281</m:to>  
  <m:from>chande@cs.helsinki.fi</m:from>  
  <m:message>  
    <m:object m:type="text"> This is a test message </m:object>  
    <m:object m:type="binary_attachment"> ..... </m:object>  
  </m:message>  
</my_message>
```



Namespace Scoping

- The namespace declaration is considered to be applicable to the element construct in which it is declared and with the sub element constructs.
- This can be overridden if the another namespace is declared within the element construct

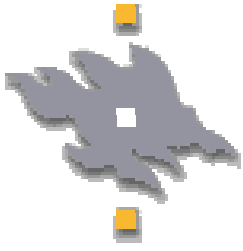
```
<my_message xmlns:m="http://www.mobile.com/mms_schema"
             xmlns:p="urn:www.mobile.com:presenceschema">
  <m:to>+358504837281</m:to>
  <p:status>meeting</p:status>
  <m:from>chande@cs.helsinki.fi</mfrom>
  <m:message>
    <m:object m:type="text"> we need to talk urgently</m:object>
    <m:object m:type="binary_attachment"> ..... </m:object>
  </m:message>
</my_message>
```



Defaulting Namespace

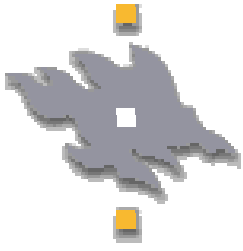
- The namespace applicable can be defaulted to the XML document structure if the prefix is not specified:

```
<my_message xmlns="http://www.mobile.com/mms_schema">
  <to>+358504837281</to>
  <from>chande@cs.helsinki.fi</from>
  <message>
    <object type="text"> This is a test message </object>
    <object type="binary_attachment"> ..... </object>
  </message>
</my_message>
```



To Summarise

- Web Evolution
- Emergence of Web Services
- Brief Overview
 - URI, URN, URL
 - XML Overview
 - XML Namespace



All References

History of Internet & WWW: <http://www.netvalley.com/intval1.html>

A little History of WWW: <http://www.w3.org/History.html>

Web design Issues: <http://www.w3.org/DesignIssues/Overview.html>

Web Addressing : <http://www.w3.org/Addressing>

URI : <http://www.ietf.org/rfc/rfc2396.txt>

URI, URL, URN Clarifications: <http://www.w3.org/TR/uri-clarification/>

Guidelines for defining new URL Schemas: <http://www.ietf.org/rfc/rfc2718.txt>

Definition of URN namespaces: <http://www.ietf.org/rfc/rfc2611.txt>

W3C's XML Recommendation : <http://www.w3.org/TR/1998/REC-xml-19980210#ISO8879>

Zvon – A good source of tutorials : <http://www.zvon.org>

XML.Org: Good XML source of information: <http://www.xml.org>