



College of Information Studies

University of Maryland Hornbake Library Building College Park, MD 20742-4345

XML + Ajax

Week 13

INFM 603

Agenda

- Questions
- XML
- Web Services
- Ajax
- CSS?
- Drupalcon
- Project discussion

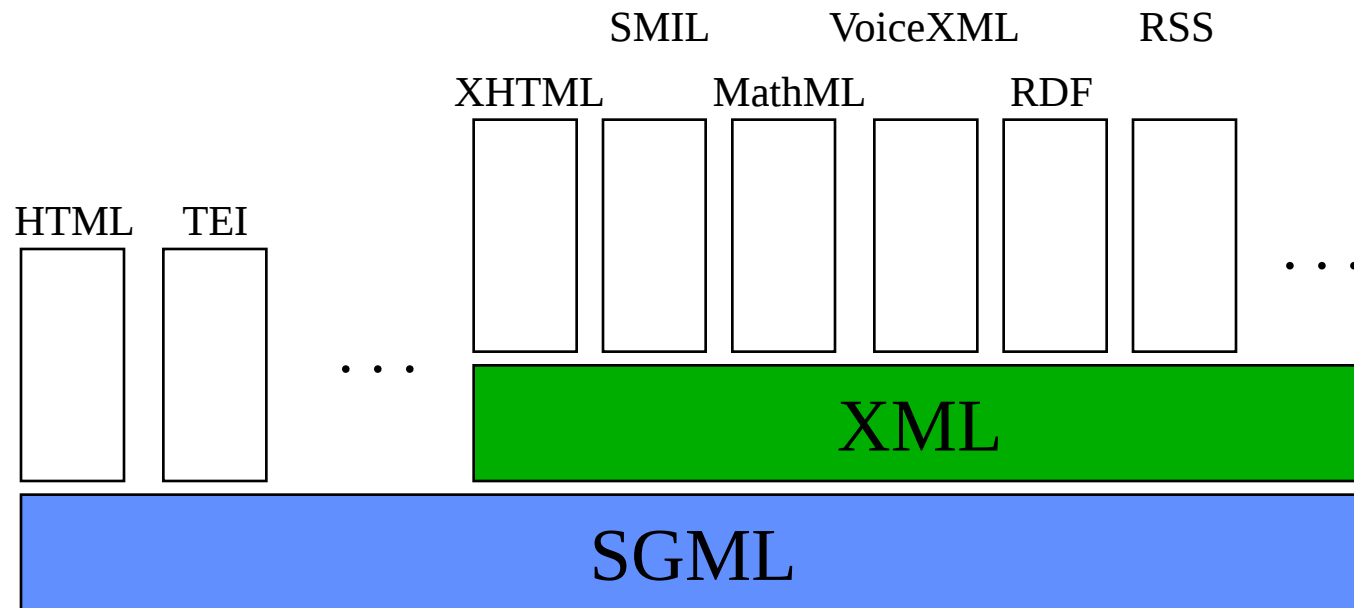
History of Structured Documents

- Early standards were “typesetting languages”
 - NROFF, TeX, LaTeX, SGML
- HTML was developed for the Web
- Specialized standards met other needs
 - Change tracking in Word, annotating manuscripts, ...
- XML seeks to unify these threads
 - One standard format for printing, viewing, processing

eXtensible Markup Language (XML)

- SGML was too complex
- HTML was too simple
- Goals for XML
 - Easily adapted to specific tasks
 - Rendering Web pages
 - Encoding metadata
 - “Semantic Web”
 - Easily created
 - Easily processed
 - Easily read
 - Concise

The XML Family Tree



Some XML Applications

- Text Encoding Initiative
 - For adding annotation to historical manuscripts
 - <http://www.tei-c.org/>
- Encoded Archival Description
 - To enhance automated processing of finding aids
 - <http://www.loc.gov/ead/>
- Metadata Encoding and Transmission Standard
 - Bundles descriptive and administrative metadata
 - <http://www.loc.gov/standards/mets/>

XHTML: Cleaning up HTML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<html xmlns="http://www.w3.org/TR/xhtml1" >
<head>
  <title> Title of text XHTML Document </title>
</head>
<body>
<div class="myDiv">
  <h1> Heading of Page </h1>
  <p> here is a paragraph of text. I will include inside this paragraph
    a bunch of wonky text so that it looks fancy. </p>
  <p>Here is another paragraph with <em>inline emphasized</em>
    text, and <b> absolutely no</b> sense of humor. </p>
  <p>And another paragraph, this one with an  image, and a <br /> line break. </p>
</div>
</body></html>
```

Really Simple Syndication (RSS)



```
<?xml version="1.0"?>
<rss version="2.0">
<channel>
  <title>Lift Off News</title>
  <link>http://liftoff.msfc.nasa.gov/</link>
  <description>Liftoff to Space Exploration.</description>
  <language>en-us</language>
  <pubDate>Tue, 10 Jun 2003 04:00:00 GMT</pubDate>
  <lastBuildDate>Tue, 10 Jun 2003 09:41:01 GMT</lastBuildDate>
  <docs>http://blogs.law.harvard.edu/tech/rss</docs>
  <generator>Weblog Editor 2.0</generator>
  <managingEditor>editor@example.com</managingEditor>
  <webMaster>webmaster@example.com</webMaster>
  <ttl>5</ttl>
  <item>
    <title>Star City</title>
    <link>http://liftoff.msfc.nasa.gov/news/2003/news-starcity.asp</link>
    <description>How do Americans get ready to work with Russians aboard the International Space Station? They take
      a crash course in culture, language and protocol at Russia's Star City.</description>
    <pubDate>Tue, 03 Jun 2003 09:39:21 GMT</pubDate>
    <guid>http://liftoff.msfc.nasa.gov/2003/06/03.html#item573</guid>
  </item>
</channel>
</rss>
```

See example at <http://www.nytimes.com/services/xml/rss/>

Atom Feeds

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>Example Feed</title>
  <subtitle>A subtitle.</subtitle>
  <link href="http://example.org/feed/" rel="self"/>
  <link href="http://example.org/" />
  <updated>2003-12-13T18:30:02Z</updated>
  <author>
    <name>John Doe</name>
    <email>johndoe@example.com</email>
  </author>
  <id>urn:uuid:60a76c80-d399-11d9-b91C-0003939e0af6</id>
  <entry>
    <title>Atom-Powered Robots Run Amok</title>
    <link href="http://example.org/2003/12/13/atom03"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <updated>2003-12-13T18:30:02Z</updated>
    <summary>Some text.</summary>
  </entry>
</feed>
```

Synchronized Multimedia Integration Language (SMIL)

- Window controls (in <head>)
 - Controlling layout: <region>, <root-layout>
- Timeline controls (in <body>)
 - Sequence control: <seq>, <excl>, <par>
 - Timing control: <begin>, <end>, <dur>
- Content types (in <body>)
 - <audio>, <video>, , <ref>

XML: A Family of Standards

- Definition: DTD or Schema
 - Known types of entities with “labels”
 - Defines part-whole and is-a relationships
- Markup: XML
 - “Tags” regions of text with labels
- Presentation: XSLT
 - Specifies how each type of entity should be rendered

Some Basic Rules for All XML

- XML is case sensitive
- XML declaration is the first statement
 - `<?xml version="1.0"?>`
- An XML document is a “tree”
 - Must contain one root element
 - Other elements must be properly nested
- **All** start tags must have end tags
- Attribute values must have quotation marks
 - `<item id="33905">`
- Certain characters are “reserved”
 - For example: < is used to represent `<`

XML Example

- View “The Song of the Wandering Aengus”
 - See related materials with this week’s slides
- Built from three files
 - yeats01.xml
 - poem01.dtd
 - poem01.xsl

XML Example

```
<?xml version="1.0"?>
```

```
<!DOCTYPE POEM SYSTEM "poem01.dtd">
```

```
<?xml-stylesheet type="text/xsl" href="poem01.xsl"?>
```

```
<POEM>
```

```
    <TITLE>The Song of Wandering Aengus</TITLE>
```

```
    <AUTHOR> <FIRSTNAME>W.B.</FIRSTNAME>
```

```
                <LASTNAME>Yeats</LASTNAME>
```

```
    </AUTHOR>
```

```
<STANZA>
```

```
    <LINE>I went on to the hazel wood,</LINE>
```

```
    <LINEIN>Because a fire was in my head,</LINEIN>
```

```
    <LINE>And cut and peeled a hazel wand,</LINE>
```

```
</STANZA>
```

```
</POEM>
```

Document Type Definition (DTD)

<!ELEMENT poem ((title, author, stanza)*)>

<!ELEMENT title (#PCDATA) >

<!ELEMENT author (firstname, lastname) >

<!ELEMENT firstname (#PCDATA) >

<!ELEMENT lastname (#PCDATA) >

<!ELEMENT stanza (line+ | linein+) >

<!ELEMENT line (#PCDATA) >

<!ELEMENT linein (#PCDATA) >

#PCDATA	span of text
<i>a,b</i>	<i>a</i> followed by <i>b</i>
<i>a b</i>	either <i>a</i> or <i>b</i>
<i>a*</i>	0 or more <i>a</i> 's
<i>a+</i>	1 or more <i>a</i> 's

Specifying Appearance: XSL(T)

```
<xsl:template match="POEM">
  <HTML>
  <BODY BGCOLOR="#FFFFCC">
    <xsl:apply-templates/>
  </BODY>
</HTML>
</xsl:template>
```

```
<xsl:template match="TITLE">
  <H1>
  <FONT COLOR="Green">
    <xsl:value-of/>
  </FONT>
</H1>
</xsl:template>
```

Multiple XML Namespaces

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rss="http://purl.org/rss/1.0/"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rss:channel rdf:about="http://www.xml.com/xml/news.rss">
    <rss:title>XML.com</rss:title>
    <rss:link>http://xml.com/pub</rss:link>
    <dc:description>
      XML.com features a rich mix of
      information and services for the XML community.
    </dc:description>
    <dc:subject>XML, RDF, metadata, information
      syndication services</dc:subject>
    <dc:identifier>http://www.xml.com</dc:identifier>
    <dc:publisher>O'Reilly & Associates, Inc.</dc:publisher>
    <dc:rights>Copyright 2000, O'Reilly &
      Associates, Inc.</dc:rights>
  </rss:channel>
</rdf:RDF>
```

Example from <http://www.xml.com/pub/a/2000/10/25/dublincore/>

XML DOM Parser (IE)

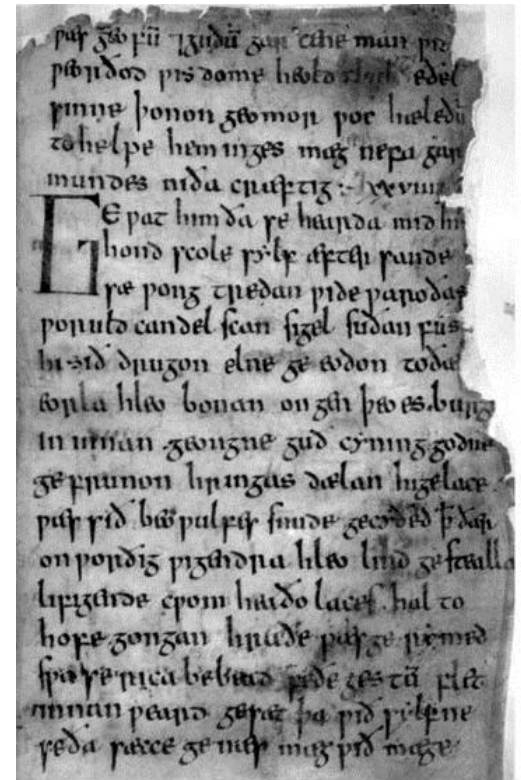
```
doc = new ActiveXObject("Msxml2.DOMDocument.6.0");  
doc.async = false;  
doc.load("sports.xml");  
result = doc.selectNodes ("/sports/game/name");  
for (i in result) {  
    document.writeln(result.item(i).text + "<br />");  
}
```

Resource Description Framework

```
<http://dbpedia.org/resource/Ayn_Rand> <http://dbpedia.org/ontology/deathDate>
    "1982-0306"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://dbpedia.org/resource/Ayn_Rand> <http://dbpedia.org/ontology/birthPlace>
    <http://dbpedia.org/resource/Russian_Empire> .
<http://dbpedia.org/resource/Ayn_Rand> <http://dbpedia.org/ontology/birthPlace>
    <http://dbpedia.org/resource/Saint_Petersburg,_Russia> .
<http://dbpedia.org/resource/Ayn_Rand> <http://dbpedia.org/ontology/birthDate>
    "1905-02-02"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://dbpedia.org/resource/Ayn_Rand> <http://purl.org/dc/elements/1.1/description>
    "novelist, philosopher, playwright, screenwriter"@en .
<http://dbpedia.org/resource/Ayn_Rand> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
    <http://xmlns.com/foaf/0.1/Person> .
<http://dbpedia.org/resource/Ayn_Rand> <http://xmlns.com/foaf/0.1/givenName>
    "Ayn"@en .
<http://dbpedia.org/resource/Ayn_Rand> <http://xmlns.com/foaf/0.1/surname>
    "Rand"@en .
<http://dbpedia.org/resource/Ayn_Rand> <http://xmlns.com/foaf/0.1/name>
    "Ayn Rand"@en .
```

Representing Complex Structures

- Concurrent hierarchies
- Standoff annotation

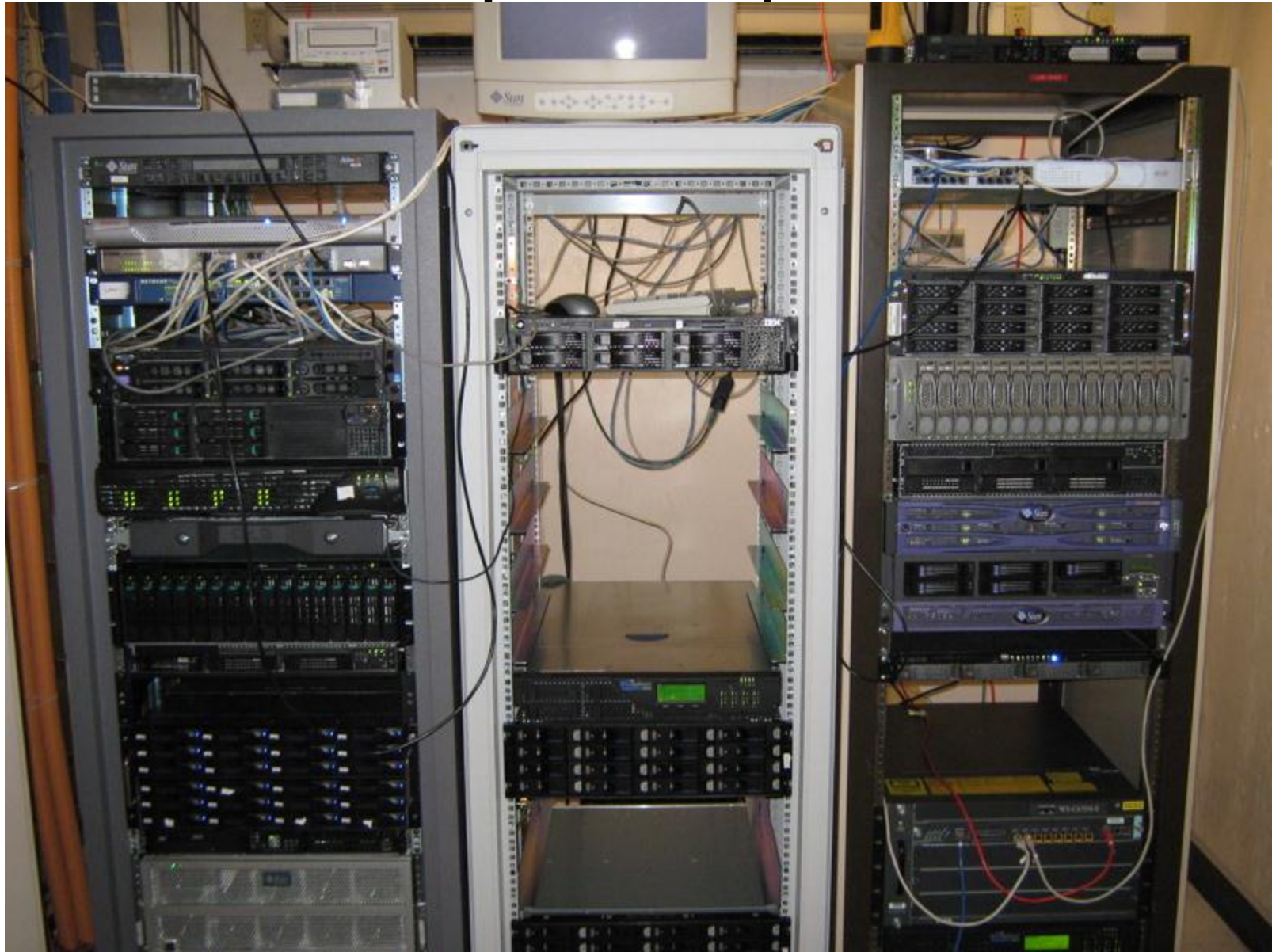


Beowulf, f.173r

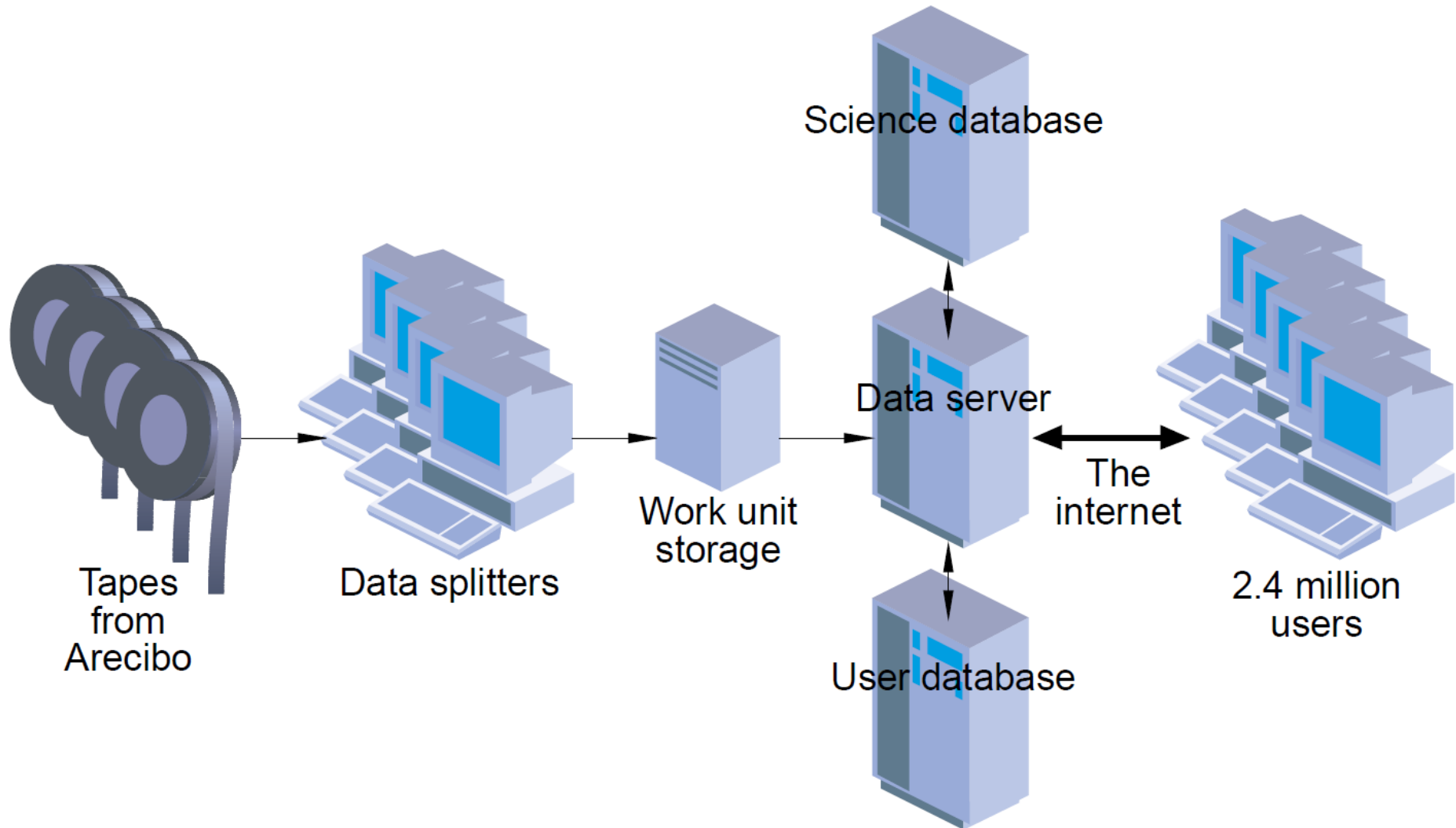
Networked Data Exchange

- Service-Oriented
 - Network of Workstations
 - Web API
 - Web services
 - Service Oriented Architecture
- Content-oriented
 - Web scraping
 - Microformats
 - Data repositories
 - Linked data
- Mashups

The World's Largest Supercomputer

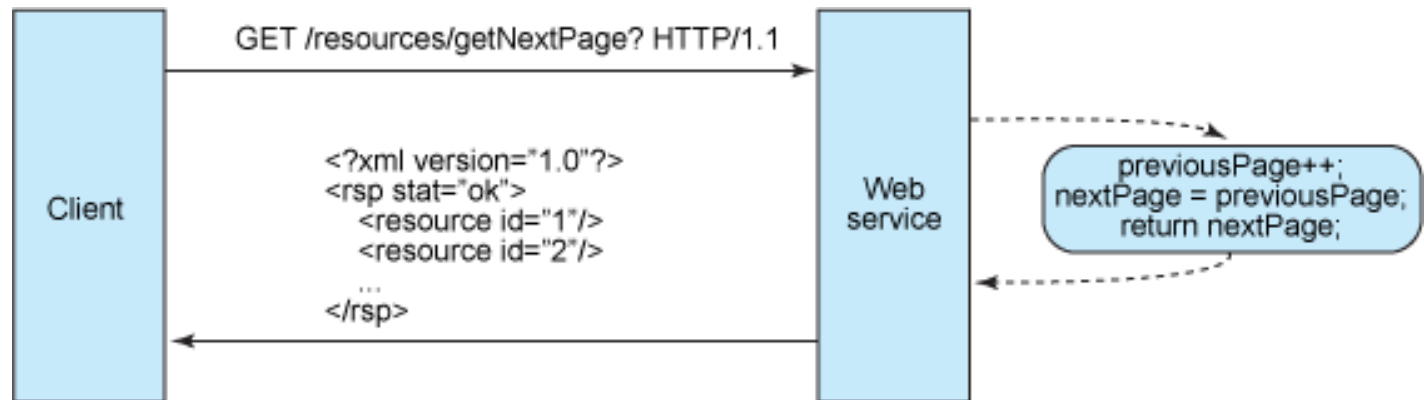


SETI@Home: Network of Workstations

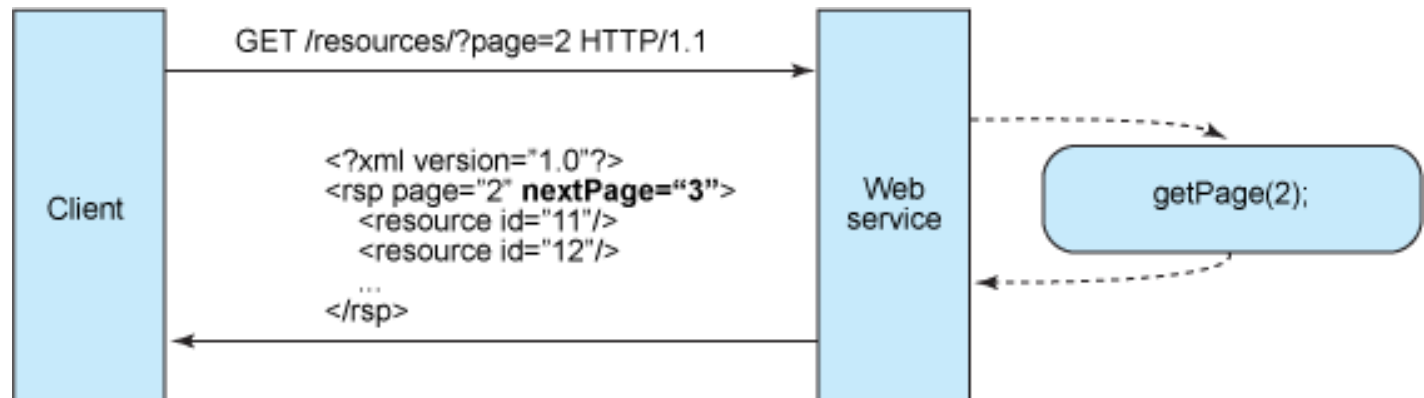


Web Services

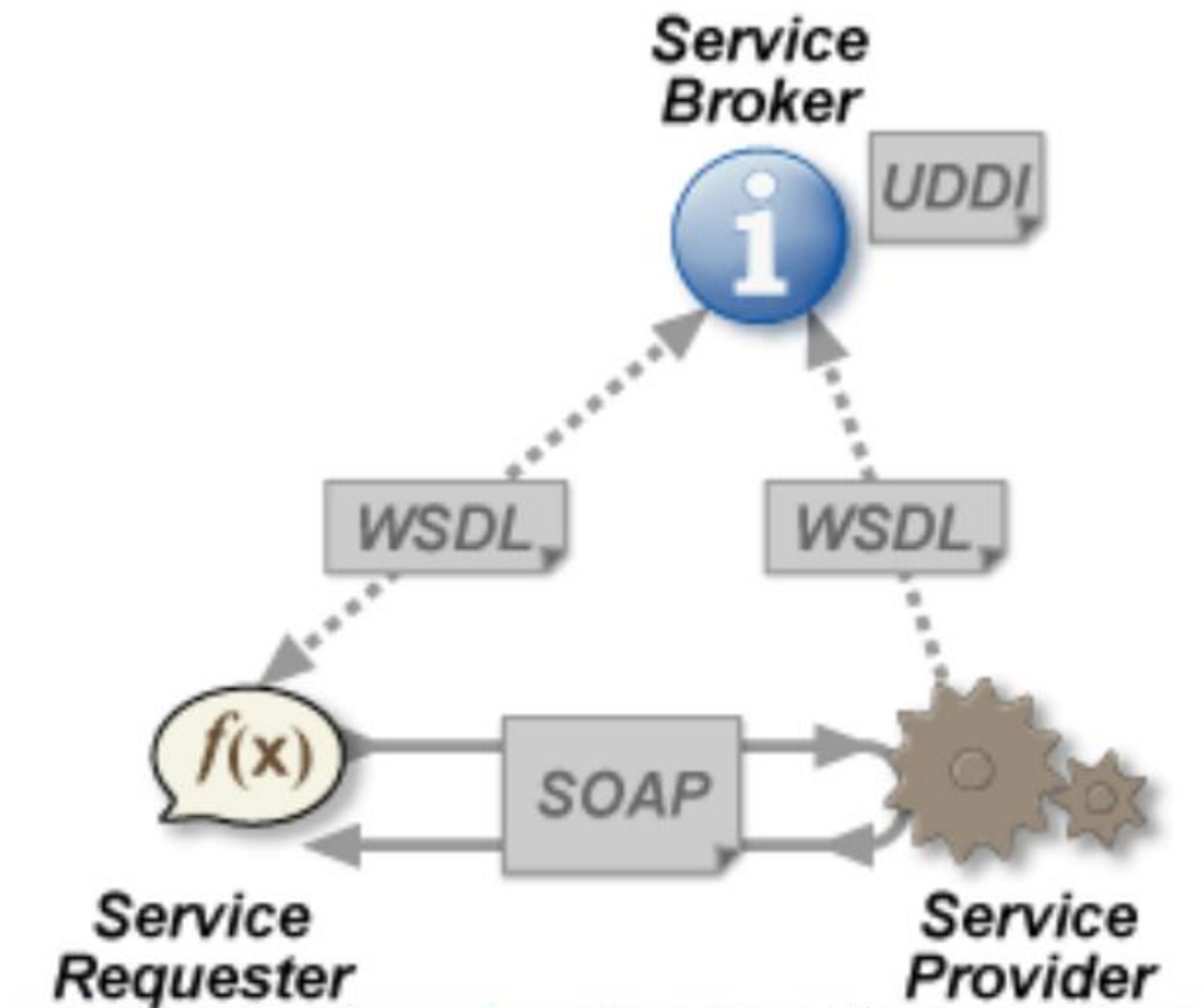
Stateful:



RESTful:

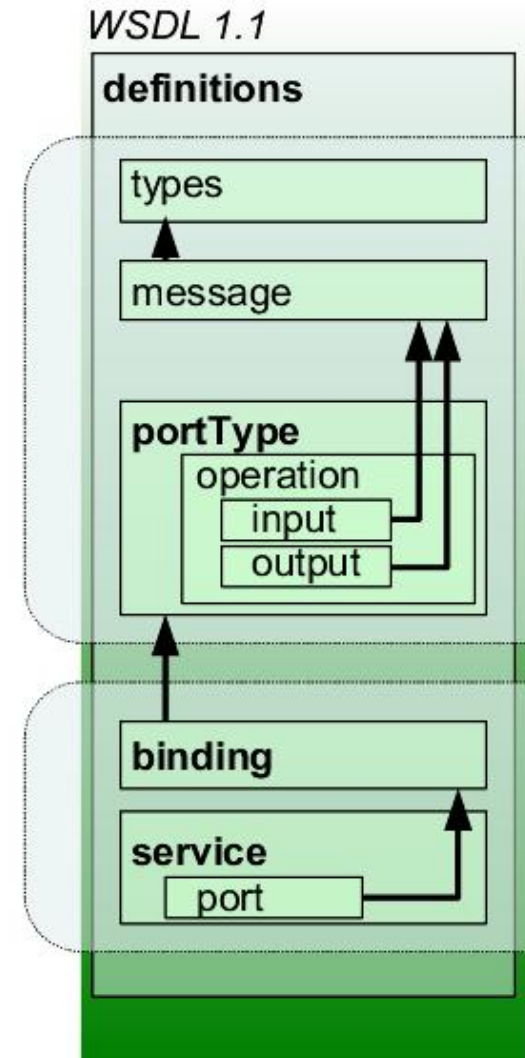


Web Services

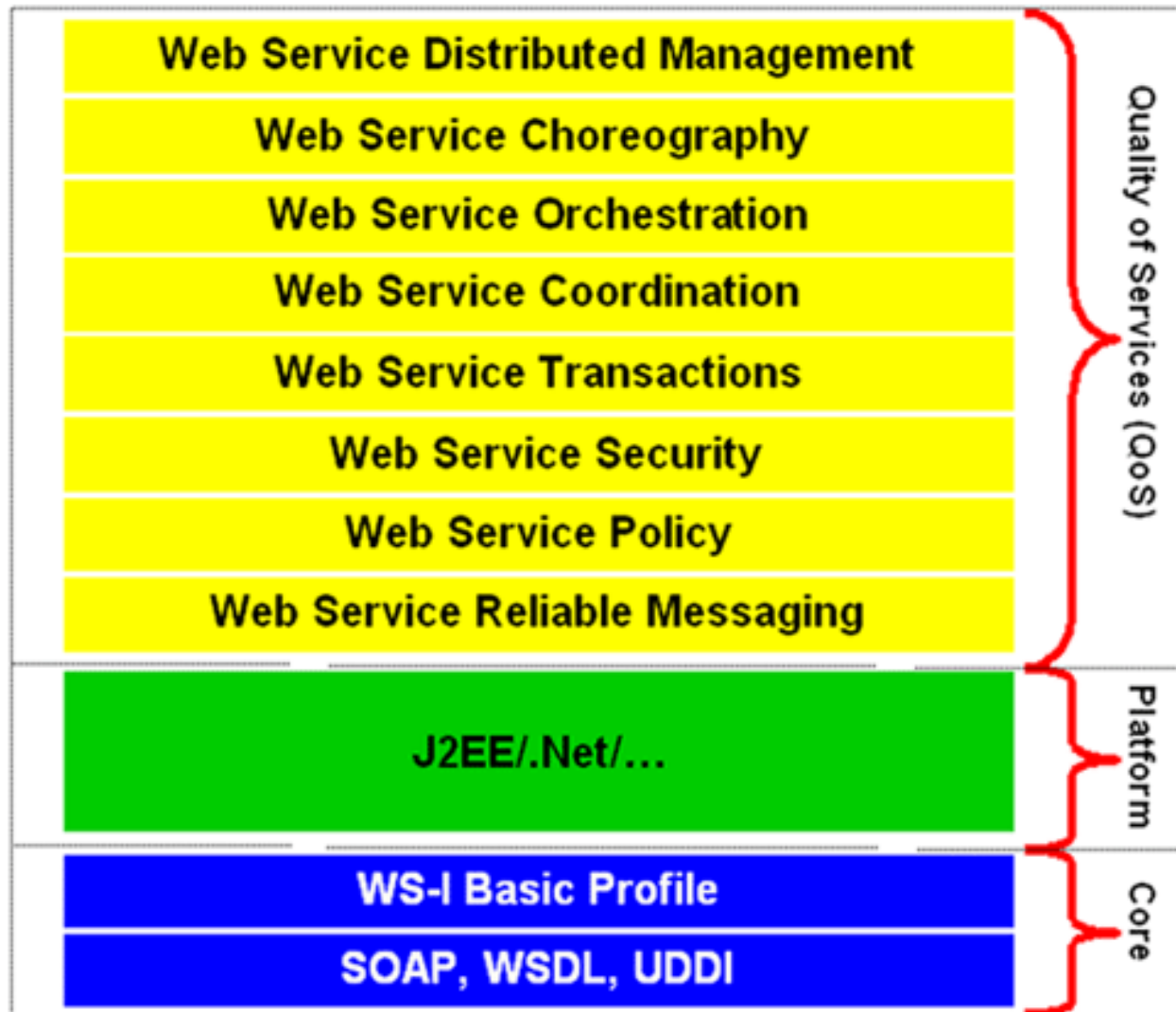


WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="HugeIntegerService"
  targetNamespace="http://hugeinteger.ch28.iw3htp4.deitel.com/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://hugeinteger.ch28.iw3htp4.deitel.com/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types><xsd:schema>
    <xsd:import xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
      schemaLocation="http://pjd:8080/HugeInteger/HugeIntegerService/
        __container$publishing$subctx/WEB-INF/wsdl/HugeIntegerService_schema1.xsd"
      namespace="http://hugeinteger.ch28.iw3htp4.deitel.com/"/>
    </xsd:schema>
  </types>
  <message name="add"><part name="parameters" element="tns:add"/></message>
  <message name="addResponse"><part name="parameters" element="tns:addResponse"/>
</message>
...
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="add"><soap:operation soapAction=""/>
<input><soap:body use="literal"/></input>
<output><soap:body use="literal"/></output></operation>
<operation name="equals"><soap:operation soapAction=""/>
<input><soap:body use="literal"/></input><output>
<soap:body use="literal"/></output>
</operation>
...
<service name="HugeIntegerService">
  <port name="HugeIntegerPort" binding="tns:HugeIntegerPortBinding">
    <soap:address xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
      location="http://pjd:8080/HugeInteger/HugeIntegerService"/>
  </port>
</service>
</definitions>
```



Services Oriented Architecture



Networked Data Exchange

- Service-Oriented
 - Network of Workstations
 - Web API
 - Web services
 - Service Oriented Architecture
- Content-oriented
 - Web scraping
 - Microformats
 - Data repositories
 - Linked data
- Mashups

Web Scraping



ScraperWiki

[About](#)

[Browse](#)

[Blog](#)

[Log in](#)

A community of programmers sifting information to give you the edge.

[Request data](#)

[Create a scraper](#)

Data Developer?

Refine and reuse data with Ruby, Python and PHP scripts that run maintenance-free in the cloud.

[« Find out more](#)



Data Requester?

Get data for predictability, scoops and better decisions. Find your surprises before they surprise you!

[Find out more »](#)

Microformats

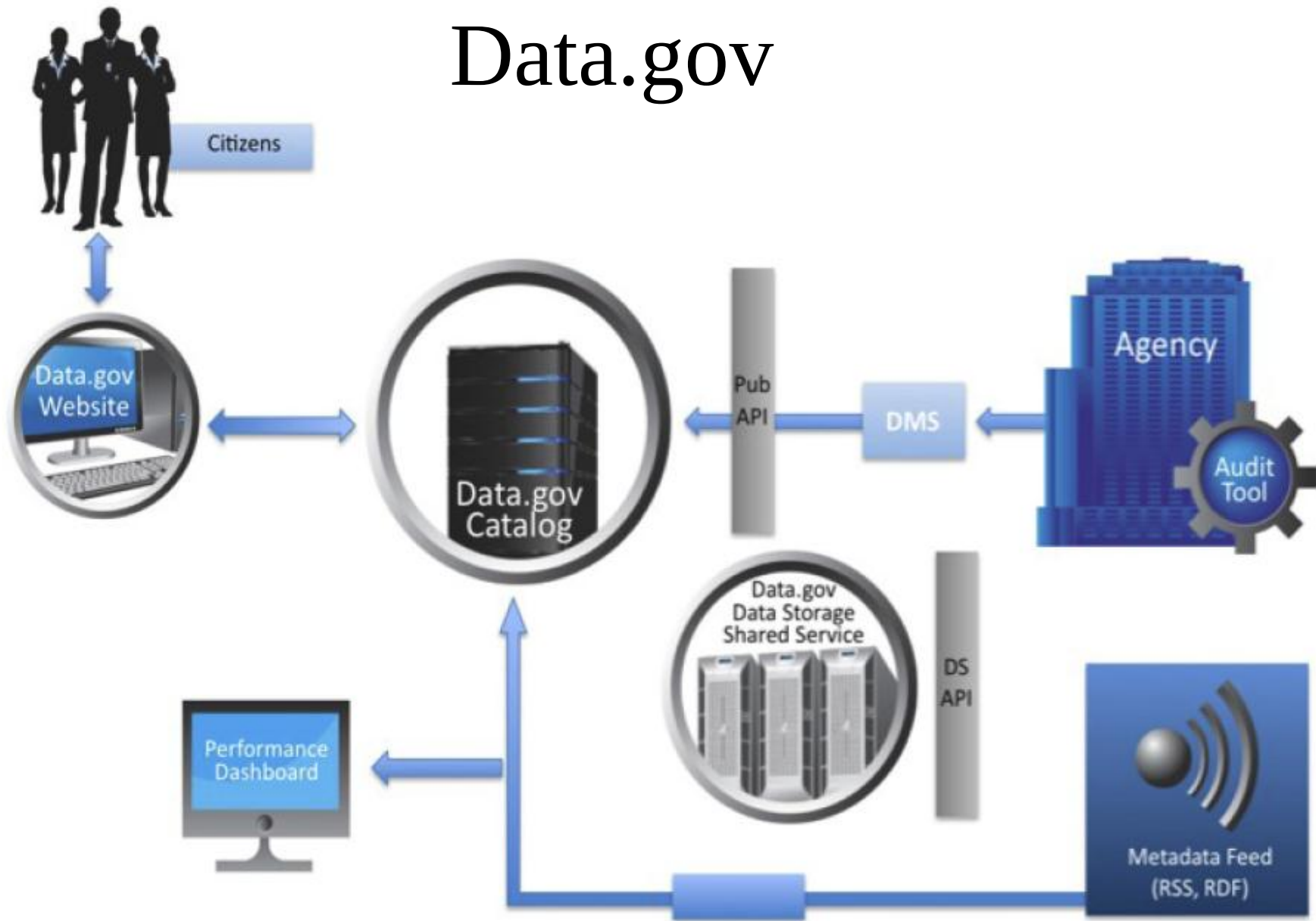
Human-Readable

```
<div id="contact">
<h2>Contact Me</h2>
<p> You can contact me via email to <a
href="mailto:jane@example.com">jan
e@example.com</a>, or send stuff to
me at the following address:</p>
<p>255 Some Street,<br />
Some Place,<br />
Some Town</p>
</div>
```

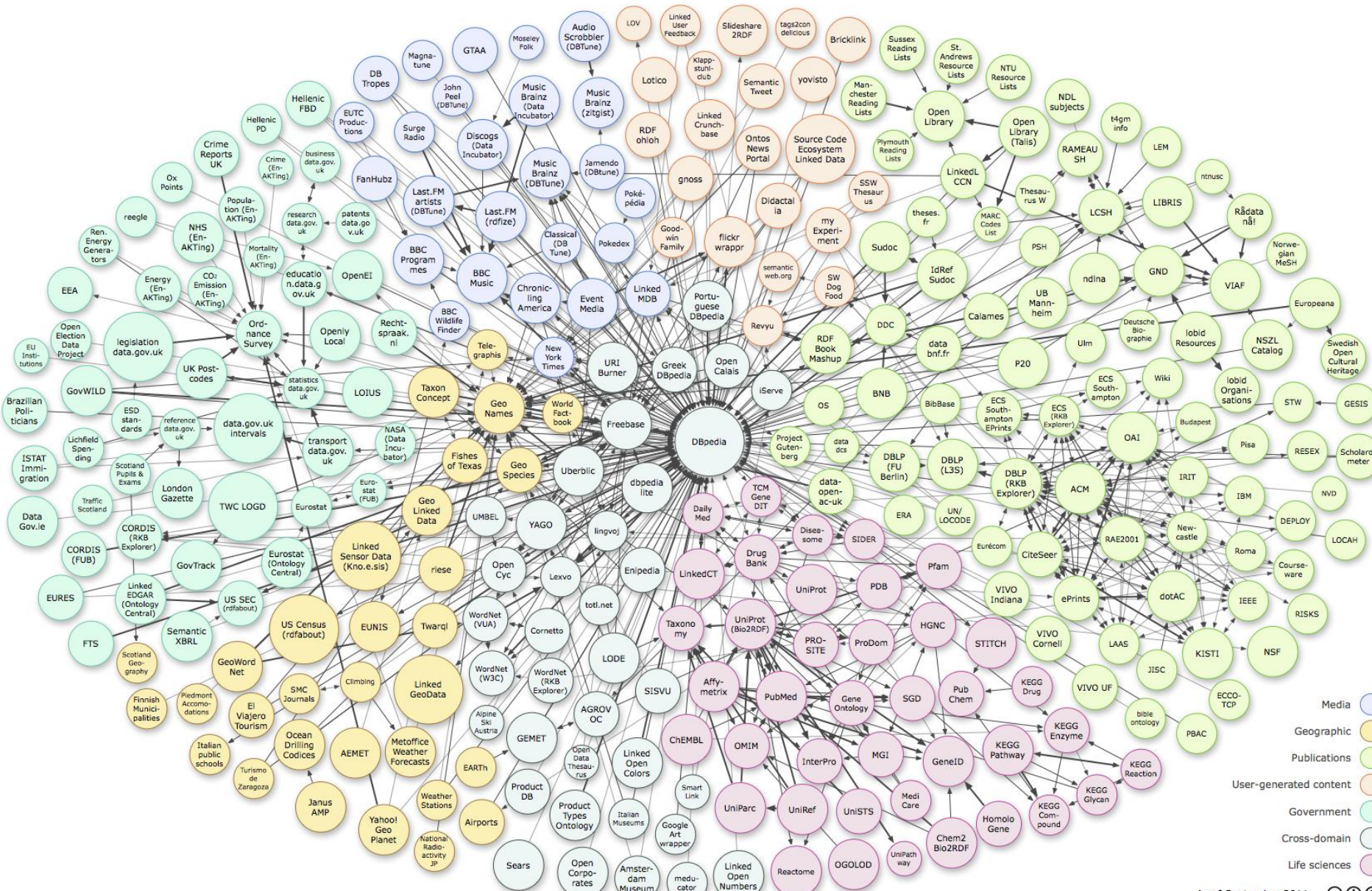
Human- and Machine-Readable

```
<div id="contact" class="vcard">
<h2>Contact Me</h2>
<h3 class="fn">Jane Doe</h3>
<p> You can contact me via email to <a
class="email"
href="mailto:jane@example.com">jane@
example.com</a>, or reach me at the
following address:</p>
<div class="adr">
<div class="street-address">255 Some
Street</div>
<div class="locality">Some Town</div>
<div class="region">Some Place</div>
</div> </div>
```

Data.gov



Linked Open Data



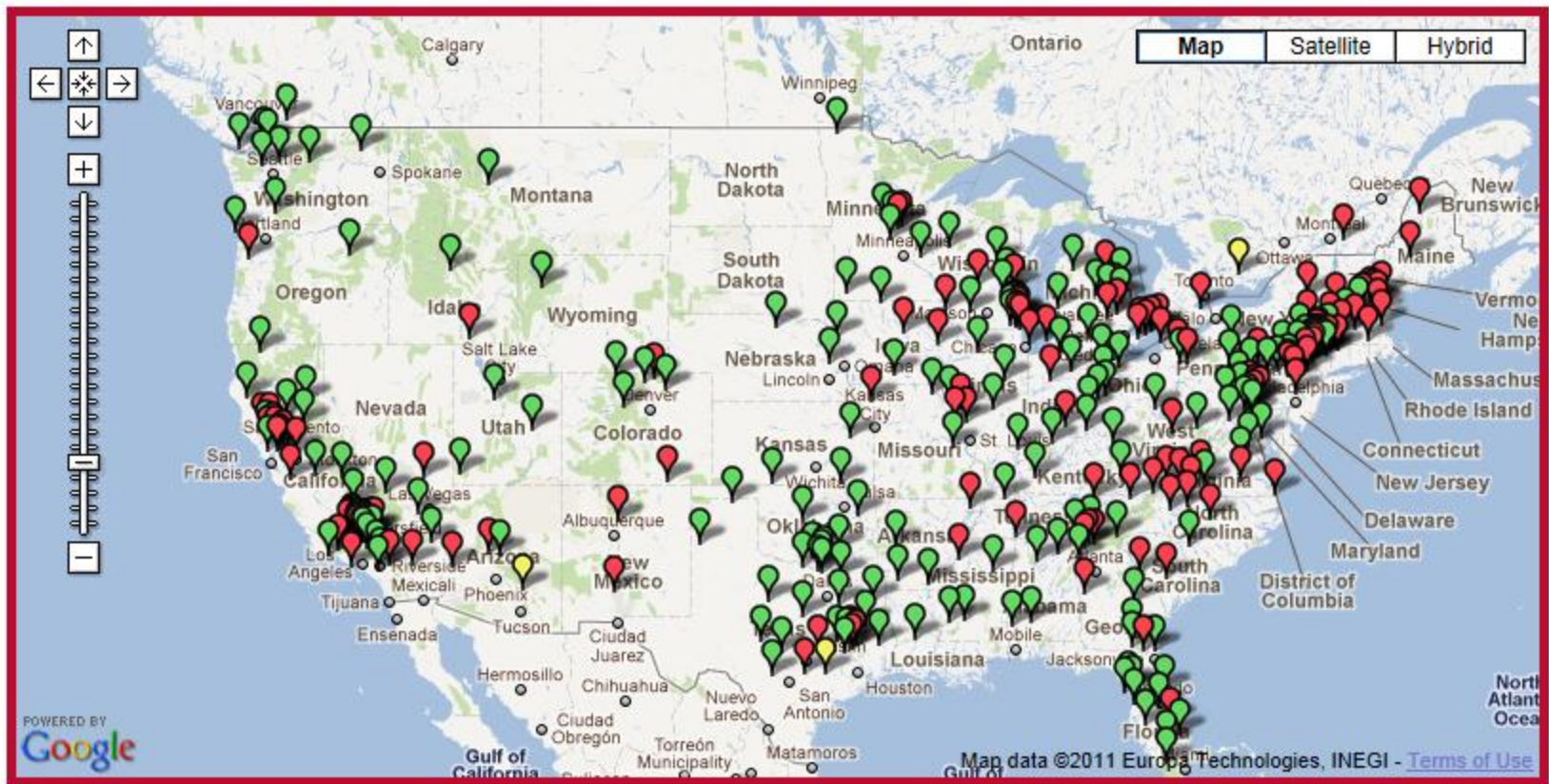
Networked Data Exchange

- Service-Oriented
 - Network of Workstations
 - Web API
 - Web services
 - Service Oriented Architecture
- Content-oriented
 - Web scraping
 - Microformats
 - Data repositories
 - Linked data

➤ Mashups

Mashups

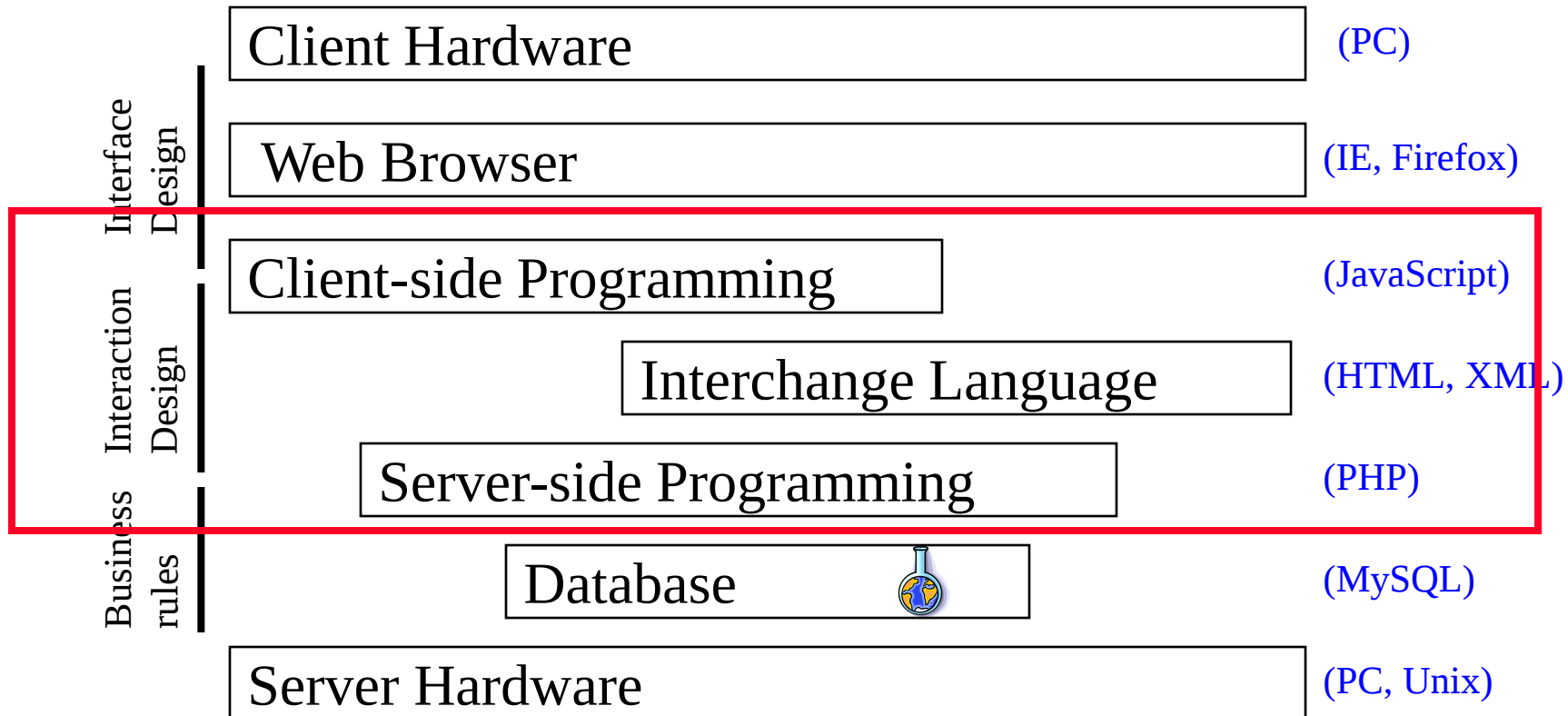
-Korea Free Trade Agreement Implementation Act



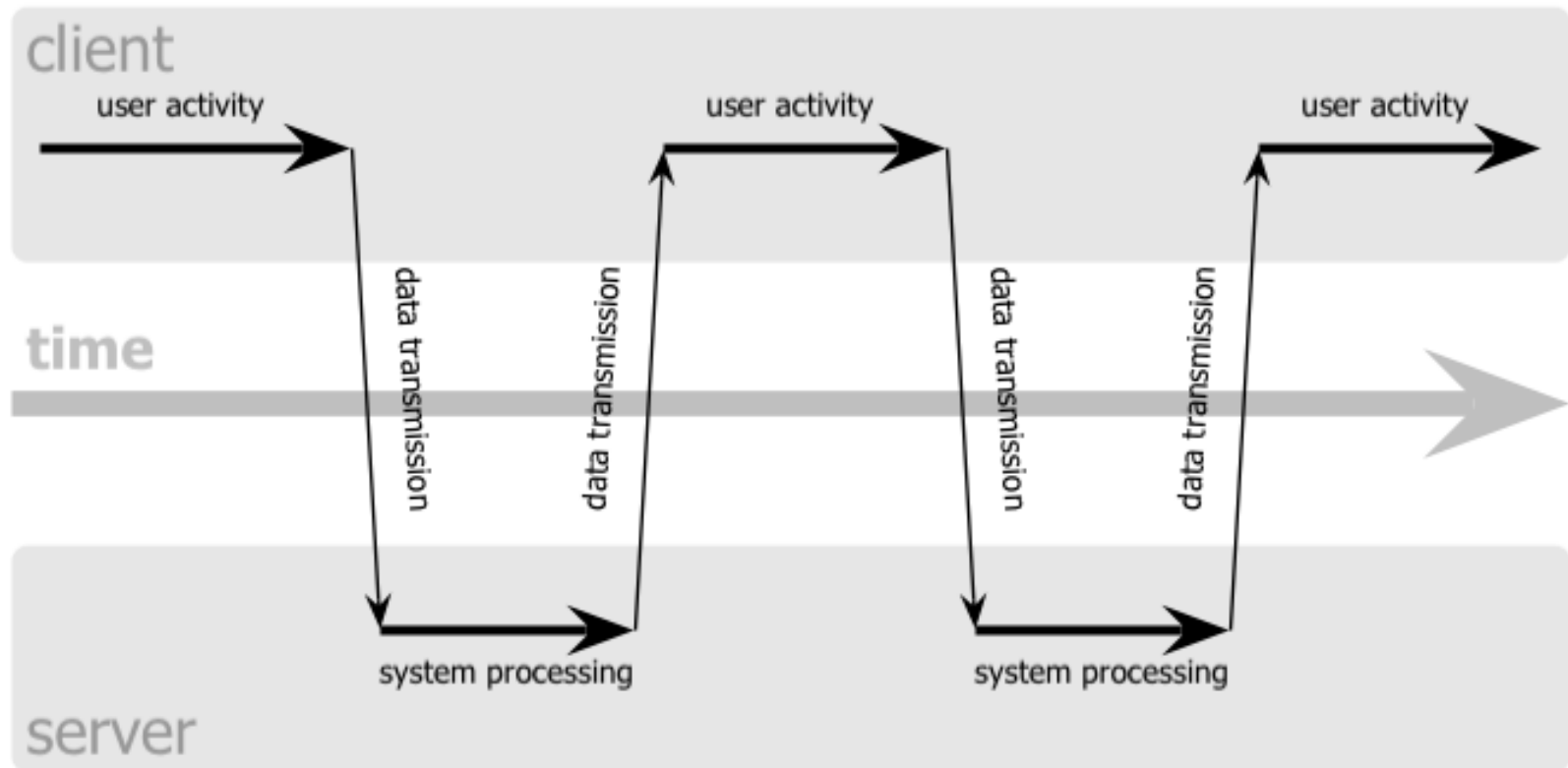
433 votes [s]

278 aye. 151 nay. 0 voting present. 4 not voting.

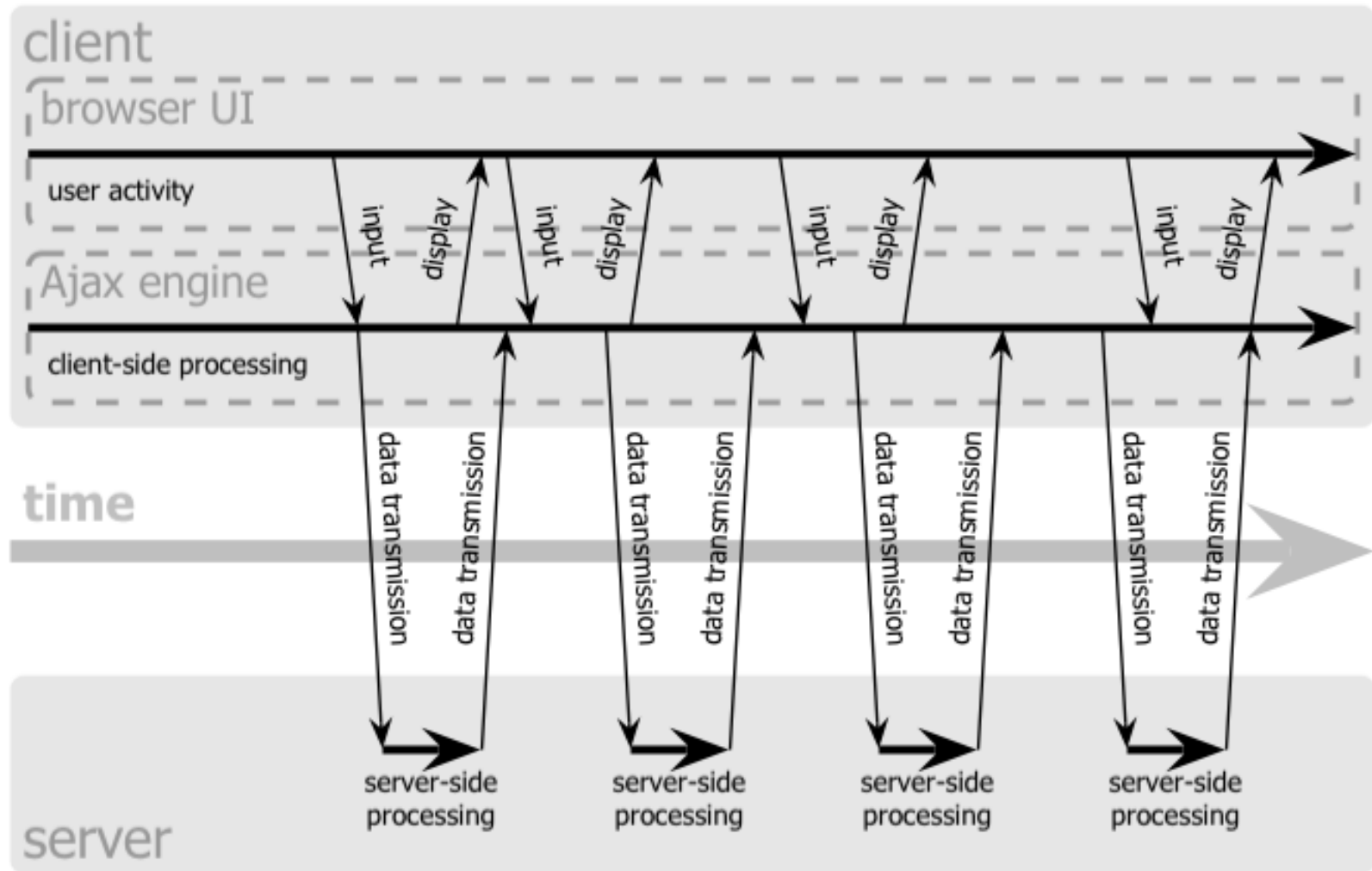
- Relational normalization
- Structured programming
- Software patterns
- Object-oriented design
- Functional decomposition



classic web application model (synchronous)



Ajax web application model (asynchronous)



Sajax: Simple Ajax for PHP

<http://www.modernmethod.com/sajax/>

```
<?
```

```
require("Sajax.php");
```

```
function multiply($x, $y) {  
    return $x * $y;  
}
```

```
sajax_init();
```

```
// $sajax_debug_mode = 1;
```

```
sajax_export("multiply");  
sajax_handle_client_request();
```

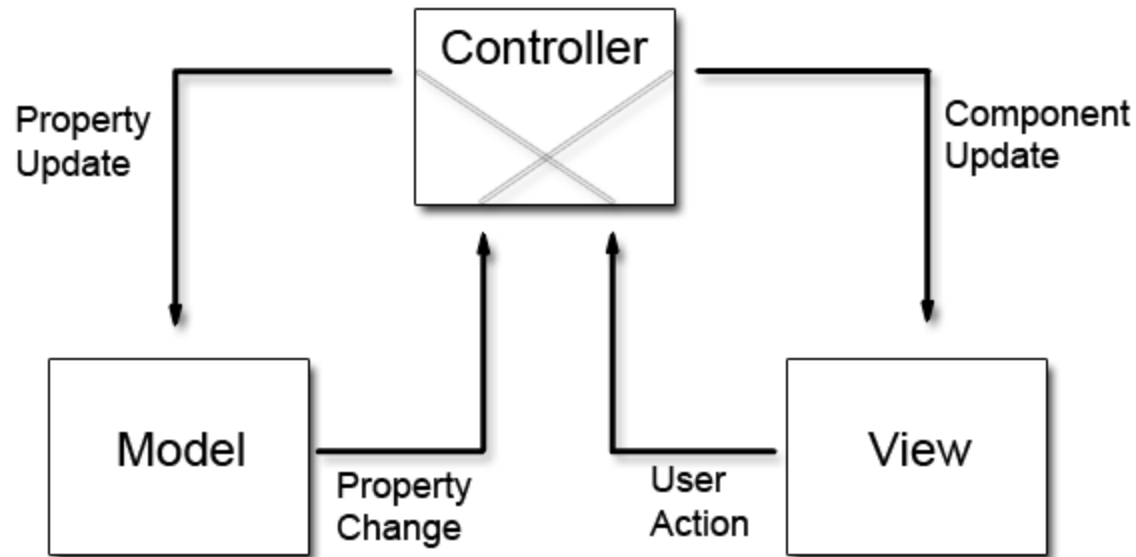
```
?>
```

```
<html><head>
  <title>Multiplier</title>
  <script>
    <?
    sajax_show_javascript();
    ?>
    function do_multiply_cb(z) {
      document.getElementById("z").value = z;
    }
    function do_multiply() {
      // get the folder name
      var x, y;
      x = document.getElementById("x").value;
      y = document.getElementById("y").value;
      x_multiply(x, y, do_multiply_cb);
    }
  </script>
</head>
<body>
  <input type="text" name="x" id="x" value="2" size="3">
  *
  <input type="text" name="y" id="y" value="3" size="3">
  =
  <input type="text" name="z" id="z" value="" size="3">
  <input type="button" name="check" value="Calculate"
    onclick="do_multiply(); return false;">
</body></html>
```

Communication methods

- GET/POST (client to server) – simple data
- XML (the “X” in AJAX)
 - Structured data between client and server.
 - Client responsible for updating page
- JSON – a light-weight alternative to XML
- HTML (server to client) – server sends rendered result
- Widget – GUI library hiding (some) AJAX details (e.g. DoJo)

Model-View-Controller



Object-relational model (ORM)

- In naïve webapp, data has three lives:
 1. Row in database
 2. Variable in program (field of object)
 3. Parameter in request (HTTP POST)
- ORM wraps database in object-oriented code (table rows look like objects)
- MVC allows near-transparent extension to Web forms, parameters.

CSS

- Separating content and structure from appearance
- Rules for defining styles “cascade” from broad to narrow:
 - Browser default
 - External style sheet
 - Internal style sheet
 - Inline style

Basics of CSS

- Basic syntax:

selector {property: value}

HTML tag you want to modify...

The property you want to change...

The value you want the property to take

- Example:

```
p { text-align: center;  
    color: black;  
    font-family: arial }
```

Causes

- Font to be center-aligned
- Font to be Arial and black

Different Ways of Using CSS

- Inline style:
 - Causes only this tag to have the desired properties
- Internal stylesheet:
 - Causes *all* tags to have the desired properties

```
<p style="font-family:arial; color:blue">...</p>
```

```
...  
<head>...  
  <style type="text/css" >  
    p { font-family:arial; color:blue}  
  </style>  
</head>  
<body>  
  <p>...</p>  
  ...
```

Customizing Classes

- Ability to define customized styles for standard HTML tags:

```
...  
<head>...  
  <style type="text/css">  
    p.style1 { font-family:arial; color:blue}  
    p.style2 { font-family:serif; color:red}  
  </style>  
</head>  
<body>  
  <p class="style1">...</p>  
  <p class="style2">...</p>  
  ...
```

External Style Sheets

- Store formatting metadata in a separate file

mystyle.css

```
p.style1 { font-family:arial; color:blue}  
p.style2 { font-family:serif; color:red}
```

```
...  
<head>...  
<link rel="stylesheet" href="mystyle.css" type="text/css" />  
</head>  
<body>  
<p class="style1">...</p>  
<p class="style2">...</p>  
...
```

