Introduction to Web Development & Web Services

27568-4578-01

Introduction to Web Development & Web Services

Written By

Rony Keren

Internet Team

John-Bryce

JOHN BRYCE
Leading in IT Education
a matrix company

Web Development

JOHN BRYCE
Leading in IT Education
a matrix company

- Intro to HTTP
- Web servers
- Dynamic content vs. static content
- CGI
  - Main Web components
  - (request, response, cookie, session)

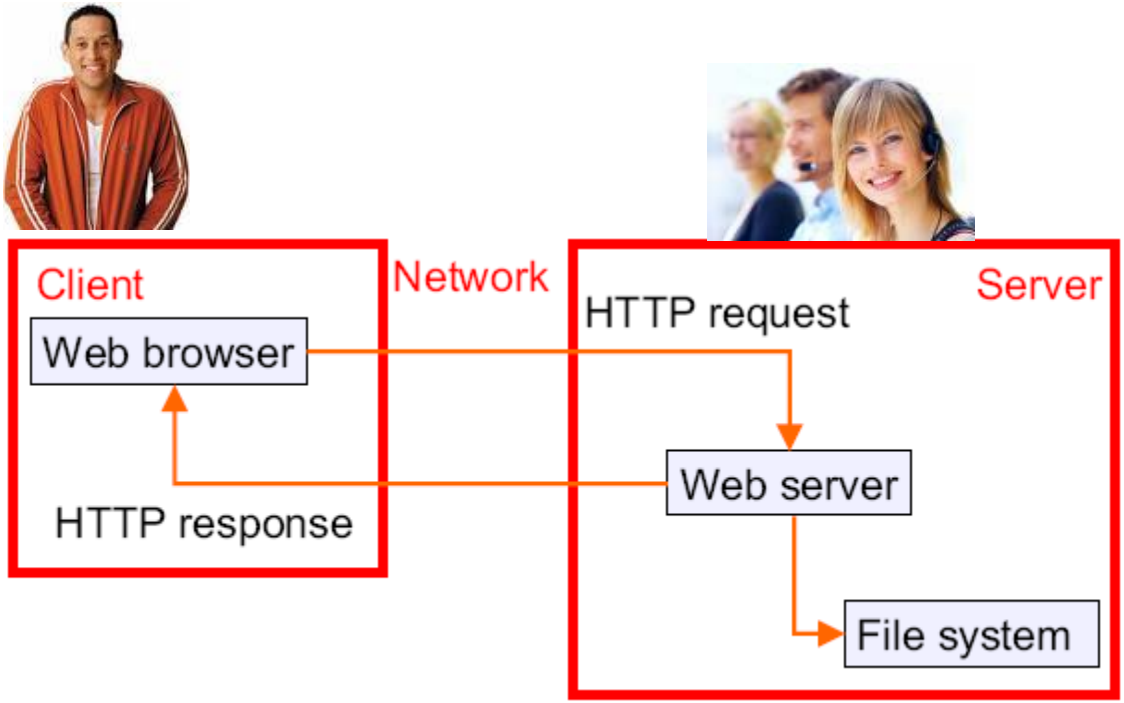
Getting Familiar With HTTP

JOHN BRYCE
Leading in IT Education
a matrix company

HTTP - Hyper **T**ext **T**ransfer **P**rotocol

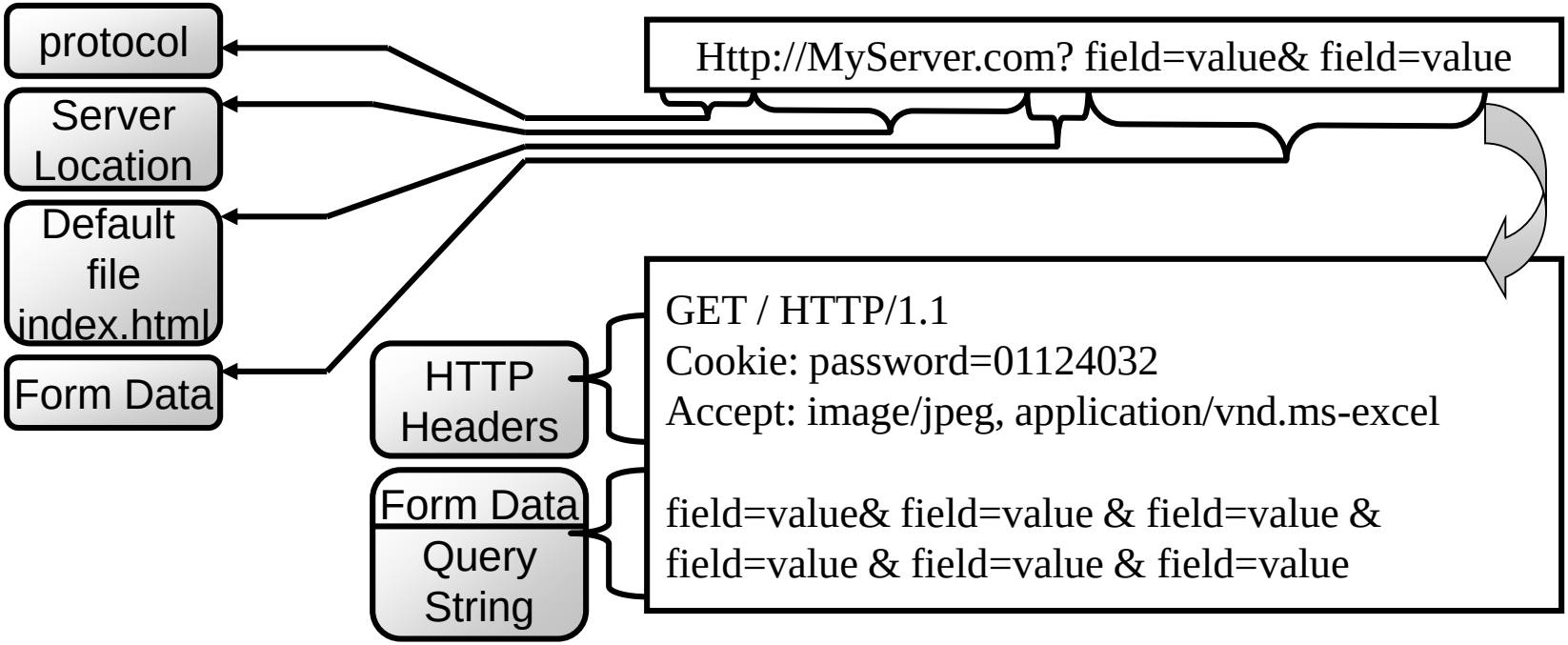
- define the data transmission during the socket life cycle
- works in a request-response environment
- The way browsers and web servers talk
- Is a W3C standard

Web application cycle



Getting Familiar With HTTP

HTTP **Request** structure:



Getting Familiar With HTTP

JOHN BRYCE
Leading in IT Education
a matrix company

HTTP Request structure:

First line - 'Request Line'
- Method (POST/GET)
- Server Root directory (/)
- Protocol HTTP (1.1/1.0)

Cookies
- Exist only by previous visits
- Invoked with the request
- Can have as much as you like
- Saved data: Field=Value

Accept
- The content type of the request
- Invoked with the request
- Specify the MIME type:
  text/plain | text/html | text/xml | image/gif

Form Data
- Contain request Data
- Filled by forms & scripts
- Converted into a QueryString by Jave Servlets-engine

GET / HTTP/1.1
Cookie: password=01124032
Accept: image.jpeg, application/vnd.ms-excel
….
Form Data

Getting Familiar With HTTP

JOHN BRYCE
Leading in IT Education
a matrix company

HTTP **Response** structure:

HTTP Headers

HTML Document

```
GET / HTTP/1.1 200 OK
Accept: image.jpeg, application/vnd.ms-excel
Date: Sun 18 Aug 2001 21:34:54
Server: TomCat/4.0-b7
Connection: keep-alive
Host: 127.0.0.1
Set-Cookie: pass=01a0Z  expires=Tue 22 Aug 2001
Content-Type: text/html; charset=ISO-8859-1

<html>…..
```

Getting Familiar With HTTP

JOHN BRYCE
Leading in IT Education
a matrix company

HTTP Response structure:

First line - 'Response Line'
- Method (POST/GET)
- Server Root directory (/)
- Protocol HTTP (1.1/1.0)
- HTTP Response Status:
  - 200-299 - success
  - 300-399 - file transmission
  - 400-499 - client error
  - 500-599 - server errors

Date
- the date & time server response occurred

Server
- the server details and version

```
GET / HTTP/1.1 200 OK
Date: Sun 18 Aug 2001 21:34:54
Server: TomCat/4.0-b7
Connection: keep-alive
Host: 127.0.0.1
Set-Cookie: pass=01a0Z  expires=Tue 22 Aug 2001
Content-Length=12344
Content-Type: text/html; charset=ISO-8859-1


<html>…..
```

Getting Familiar With HTTP

JOHN BRYCE
Leading in IT Education
a matrix company

HTTP Response structure:

Connection
•the kind of connection:
close - when there is no content-length
keep-alive - default

Host
•the IP address of the server

Set Cookie
•add cookie (field&value)
•can determine expire date

Content-Length
•the number of bytes being sent
•without Content-Length:
Connection=close

```
GET / HTTP/1.1 200 OK
Date: Sun 18 Aug 2001 21:34:54
Server: TomCat/4.0-b7
Connection: keep-alive
Host: 127.0.0.1
Set-Cookie: pass=01a0Z  expires=Tue 22 Aug 2001
Content-Length=12344
Content-Type: text/html; charset=ISO-8859-1

<html>…..
```

Getting Familiar With HTTP

JOHN BRYCE
Leading in IT Education
a matrix company

HTTP Response structure:

Content-Type [MIME]
•response content type can be:

Type	Meaning
application/msword	Word format
application/pdf	PDF format
application/x-gzip	gzip format
application/x-java-vm	Java class file
application/zip	Zip format
audio/x-wav	Wav file
image/jpeg	Image jpeg
text/html	Html document
text/xml	XML document
text/plain	Plain text
video/mpeg	MPEG clip

```
GET / HTTP/1.1 200 OK
Date: Sun 18 Aug 2001 21:34:54
Server: TomCat/4.0-b7
Connection: keep-alive
Host: 127.0.0.1
Set-Cookie: pass=01a0Z  expires=Tue 22 Aug 2001
Content-Length=12344
Content-Type: text/html; charset=ISO-8859-1

<html>…..
```

Response encoding:
• ISO-8859-1, ISO-8859-8
•UTF-8
•windows-1255….

Web Application Structure

JOHN BRYCE
Leading in IT Education
a matrix company

Web Application includes:

- Dynamic content [some server side languages - later]
  - Are calculated according to client request
  - Each client may get a custom response
- Static content [HTML, PDFs, documents, images etc.]
  - Static contents looks always the same – no matter who is the caller
- Client-side scripts [Applets, Java-scripts, flash apps.]
  - Also static content – since it is not generated per request
  - Browsers or other client devices uses these contents to execute some client side logic / interaction

Web Servers

- Web servers are containers of Web applications
- Each server may host multiple apps
- Each apps has its dedicated URL pattern as a prefix to all of its internal content (dynamic & static)
  - http://...../appA/.....
  - http://...../appB/.....

Web Servers

Web servers are responsible for

- delegating HTTP requests to the correct application
- sending responses generated by the application back to the called
- In other words: a platform that hosts HTTP based applications
  - Http based applications are using the HTTP synchronous request-response model

Dynamic vs. Static Content

JOHN BRYCE
Leading in IT Education
a matrix company

- Static content is already knows to you…
  - CSS, HTML, Java Script and all the data they use or present are mostly static
    - Welcome, error and many other types of pages may be static
  - These technologies are good for end-user interaction
- Dynamic content
  - Programming is done mostly for a request-response flow
    - Processing client requests
    - Generating server response
  - Done mostly according to CGI W3C standard
    - Common Gateway Interface

Dynamic vs. Static Content

JOHN BRYCE
Leading in IT Education
a matrix company

- Most leading dynamic web content languages
  - Java – Servlet technology
    - Provides platforms for Enterprise and IT as well
    - Solutions for both web & business tiers
  - .NET – ASP
    - Provides platforms for Enterprise and IT as well
    - Solutions for both web & business tiers
  - PHP
    - Focuses on web tier
    - Lately is going down due to the new superstar in the hood:
  - Java Script…… what ?! We said it a client technology…

Dynamic vs. Static Content

JOHN BRYCE
Leading in IT Education
a matrix company

- Java Script…… what ?! We said it a client technology…

- NODE.JS is a java script base package for developing web applications
- Java script is the only language today that is relevant for both dynamic and static web contents
- It is preferable simply since Java Script client developers are required for any web implementation anyway…
- So now, their knowledge in Java Script can be exploited on the server side for dynamic coding as well!!

CGI

JOHN BRYCE
Leading in IT Education
a matrix company

- Common Gateway Interface
  - A W3C standard
  - Specifies how to parse HTTP request & response into objects
  - Implemented and used by web servers:
    - Server take a client request
    - Server parses it into an object in the desired language
    - Server creates a response object
    - Server passes both request & response to the application addressed by the client call
    - The application read client submit data from the request
    - The application does its computations in order to calculate a result for the caller
    - The application puts the result in the response object
    - The server converts the object into HTTP and send it back

CGI

- Common Gateway Interface
  - Main components:
  - Request
    - Holds the following:
      - URL to handle the request
      - Client form data / link data

        May be in HTTP headers (GET) or body (POST) »
      - http req. headers
  - Response
    - Holds the following:
      - Status header (number between 200-599)
      - HTTP headers
      - Result data – usually placed in the HTTP body

CGI

JOHN BRYCE
Leading in IT Education
a matrix company

- Common Gateway Interface
  - Main components:
  - Cookie
    - A piece of data placed as a HTTP response header
    - Server may plant such contents in it responses
    - Client may use the cookie data in memory of store it
    - Browsers automatically send cookies back to the server as part of the client request as long as the cookie is valid
    - Server may plant data on client side between site visits
    - Example:
      - First time you purchase a product from a site – the site plants a cookie with details of what you bought
      - Next time you visit that site – you, un-knowingly send that cookie…
      - The site now can know that it is not your first

CGI

JOHN BRYCE
Leading in IT Education
a matrix company

- Common Gateway Interface
  - Main components:
  - Cookie
    - Example:
      - First time you purchase a product from a site – the site plants a cookie with details of what you bought
      - Next time you visit that site – you, un-knowingly send that cookie…
      - The site now can know:

        that it is not your first visit »

        What you purchased on that visit »
      - And the site can now adjust the “home-page” to be much more attractive for you…..

CGI

- Common Gateway Interface
  - Main components:
  - Session
    - Session is an object that can live along multiple requests
    - Each session has a unique ID
    - The ID is planted in memory as cookie on client side
    - Every-time the client sends a request – the ID is attached
    - This is how the server bind a client request to its exisiting session

CGI

- Common Gateway Interface
  - Main components:
  - Session
    - Session is an object that can live along multiple requests
    - Each session has a unique ID
    - The ID is planted in memory as cookie on client side
    - Every-time the client sends a request – the ID is attached
    - This is how the server bind a client request to its existing session

CGI

- Common Gateway Interface
  - Main components:
  - Session
    - Example:
      - User visits a web site and decides to purchase an item
      - When the first item is requested – a session is created on server-side and the item is placed inside it. The session ID is planted on client side now.
      - When the client adds another item – the ID is also sent to the server
      - The server uses the ID to track caller's session
      - The second Item is now added to the session
      - When client wants to complete the order than the total price will be calculated and than the session will get invalidated

Web Services

Web service is:

- A service available on the internet, that uses standard protocols for integration
  - Service
  - Internet [HTTP]
  - Standard protocols

Architecture

- Web App Architecture
  - What is a service
  - 2 tier model
  - 3 tier model
  - N tier model
- XML for transferring data
  - Well formed
  - Validation and types with Schema (XSD)
  - XML Binding - JAXB
  - XML vs. JSON
- MVC Model 2
- Moving to single page applications
  - The problem with views
  - AJAX for browsers
  - Future internet clients

Architecture

What is a service?

- A model, provided by vendor, that allows clients to communicate and interact

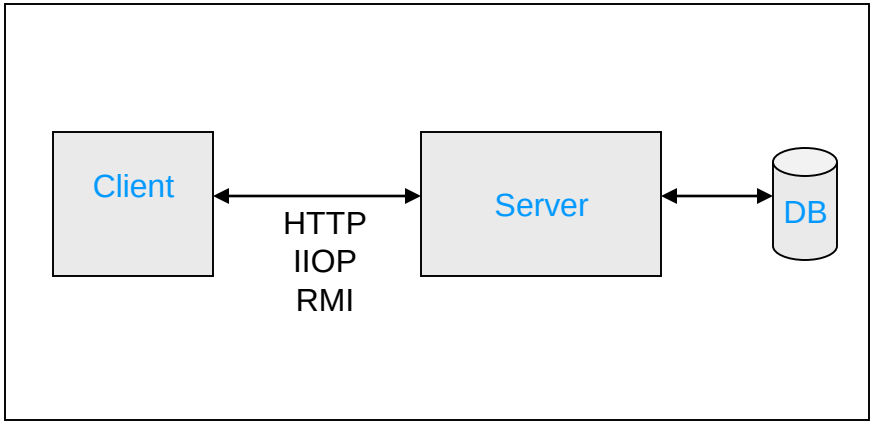
- May be self-descriptive since a contract is needed

Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

2 Tier Model

Tier 1 – container
Tier 2 – DB or any other 3rd party
Containers are focusing on communication
- Web containers – HTTP→ CGI
- RMI containers – Java connectors
- IIOP containers – IDL connectors



Disadvantages
- No infrastructure services
- Problematic when moving to large scales

Architecture

3 Tier Model

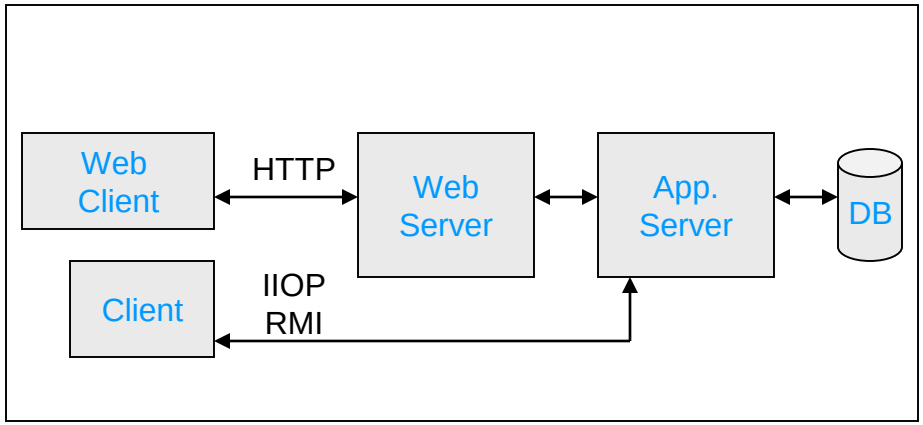
Tier 1 – Web server

Tier 2 – Business server

Tier 3 – DB or any other 3rd party

Business server provides infrastructural services

- Development focuses on service implementation
- Highly scalable
- Support various protocols



JOHN BRYCE
Leading in IT Education
a matrix company

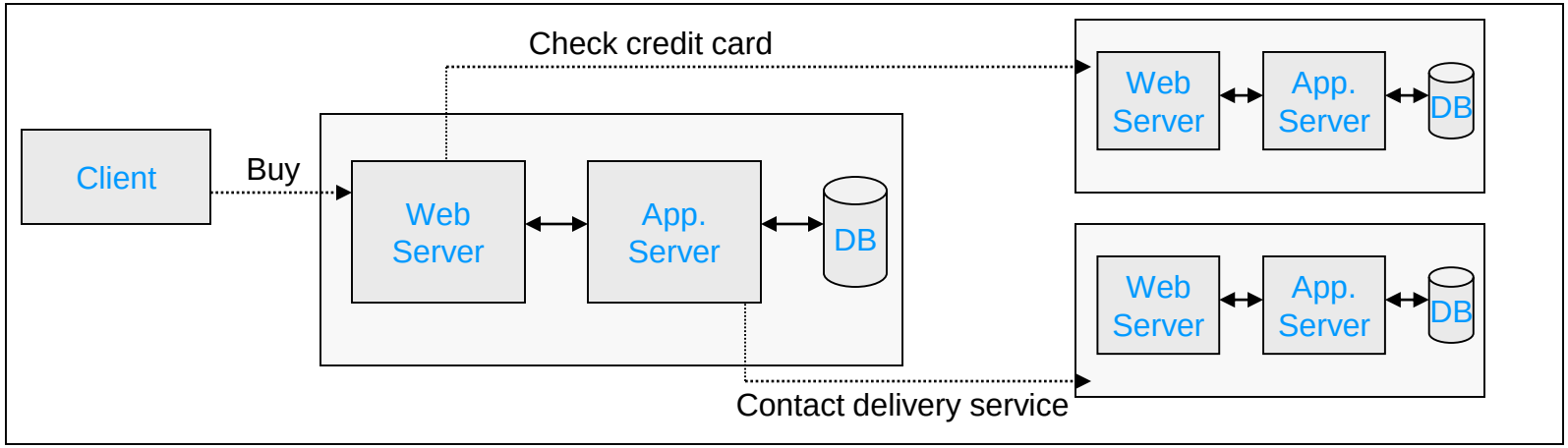
N Tier Model

2, 3 tier systems interaction

Client request might be handled by multiple systems

One system must effectively interact with another

B2B / EAI / SOA …



Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

XML for transferring data

HTML for applications. Describes plain data rather then how to present it

Application that ‘understands’ the data – can present it if needed…

Present and future devices will consume mostly data – not view

We can do much more with this

```
<people>
    <person>
        <name> David </name>
        <age> 20 </age>
    </person>
    ….
</people>
```

than we can do with that:

```
<table>
    <tr>
        <td> David </td>
        <td> 20 </td>
    </tr>
    ….
</table>
```

Architecture

- XML for transferring data
  - Well formed
    - Set of basic syntax rules
      - Including:
        - Closing tags
        - Attribute values inside quotes
        - Case sensitive
        - Correct element nesting…
    - Part of W3C XML standard
    - XML parsers must not parse any non well-formed data
    - Saves checks and manipulations for small & tiny devices
    - For browsers & micro-browsers - XHTML

Architecture

- XML for transferring data
  - Validation and types
    - XML structure is described via XSD (Schema)
    - W3C standard
    - XSD Schema defines:
      - Element name & content
      - Attributes
      - Simple and complex types
    - Since XSD defines primitives (xsd:integer, xsd:date….) – objects can be described as well..

Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

XML for transferring data

Schema example:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element name="People">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Person" type="PersonType" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:complexType name="PersonType">
        <xsd:sequence>
                <xsd:element name="Name" type="xsd:string"/>
                <xsd:element name="Age" type="AgeType"/>
                <xsd:element name="BirthDate" type="xsd:date" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="gender" type="GenderType" use="required"/>
    </xsd:complexType>
    <xsd:simpleType name="AgeType">
        <xsd:restriction base="xsd:nonNegativeInteger">
                <xsd:minInclusive value="0"/>
                <xsd:maxInclusive value="120"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="GenderType">
        <xsd:restriction base="xsd:string">
                <xsd:enumeration value="M"/>
                <xsd:enumeration value="F"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

XML for transferring data

XML example:

```
<?xml version="1.0"?>
<People xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:noNamespaceSchemaLocation="PeopleSchema.xsd">
                <Person gender="M">
                                <Name>Bill</Name>
                                <Age>35</Age>
                                <BirthDate>1984-04-13</BirthDate>
                </Person>
                <Person gender="F">
                                <Name>Dana</Name>
                                <Age>47</Age>
                                <BirthDate>1961-11-03</BirthDate>
                </Person>
                <Person gender="F">
                                <Name>Amy</Name>
                                <Age>23</Age>
                                <BirthDate>1991-04-15</BirthDate>
                </Person>
                <Person gender="M">
                                <Name>David</Name>
                                <Age>13</Age>
                                <BirthDate>2000-07-02</BirthDate>
                </Person>
</People>
```

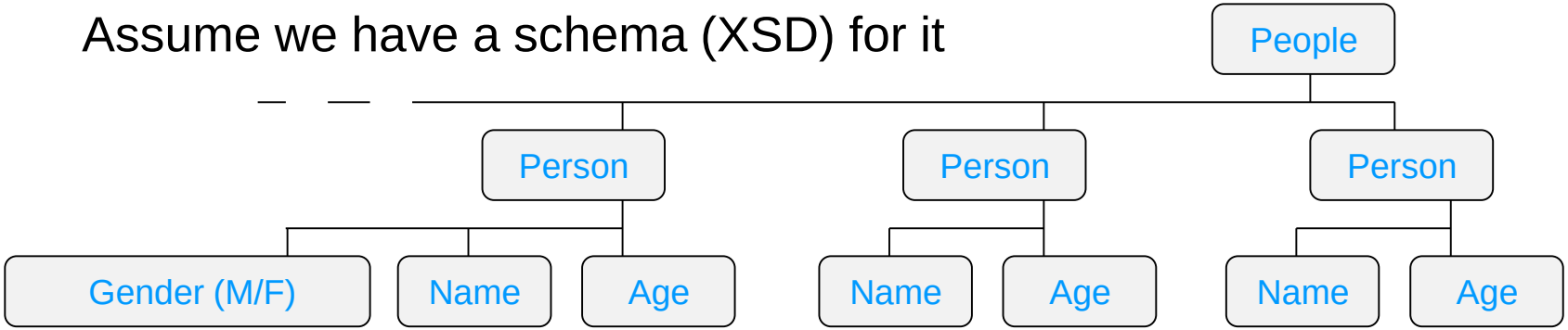
Architecture

- XML Binding
  - Useful when developing application that integrates via XML
  - Complex data structures are mapped to classes
    - Better approach then DOM

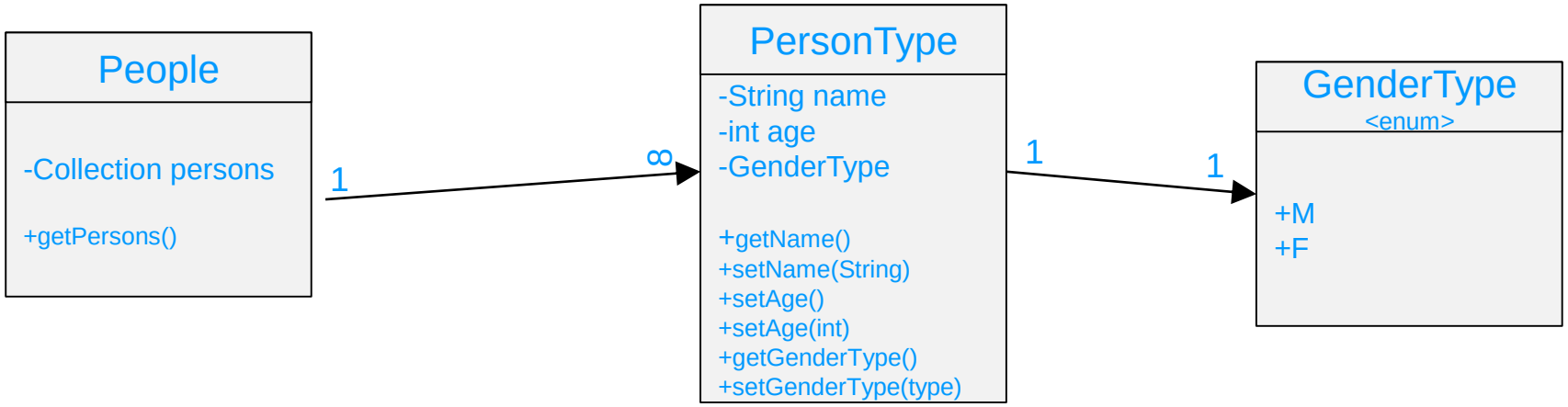
JOHN BRYCE
Leading in IT Education
a matrix company

Binding

Assume we have a schema (XSD) for it



JAXB generated classes



Architecture

With Binding we get:

- Generated classes according to schema
- Lightweight Java objects
- Auto Marshalling and Un-marshalling

Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

Un-marshalling

- Convert from XML to objects
- From text to Memory

Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

Marshalling

- Export objects from memory into XML stream
- From memory to text

Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

XML vs. JSON

What is JSON ?

- Java Script Object Notation
- Also self-descriptive text based protocol
- Used for marshalling and un-marshalling Jscript objects

```
{
 “people”: [
            { “name": “David“,  “age": “20“},
            { “name": “Dana“,  “age": “25“},
            { “name": “Eve“,  age": “30“},
 ]
}
```

Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

XML vs. JSON

Why is it an alternative for XML ?

- Better for small applications (like client side apps)
  - No parsers are needed
  - Contracts are less critical
  - Light integration
- Jscript and Android developers prefers it
  - Got popular APIs for binding, handling & presenting JSON based data

Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

XML vs. JSON

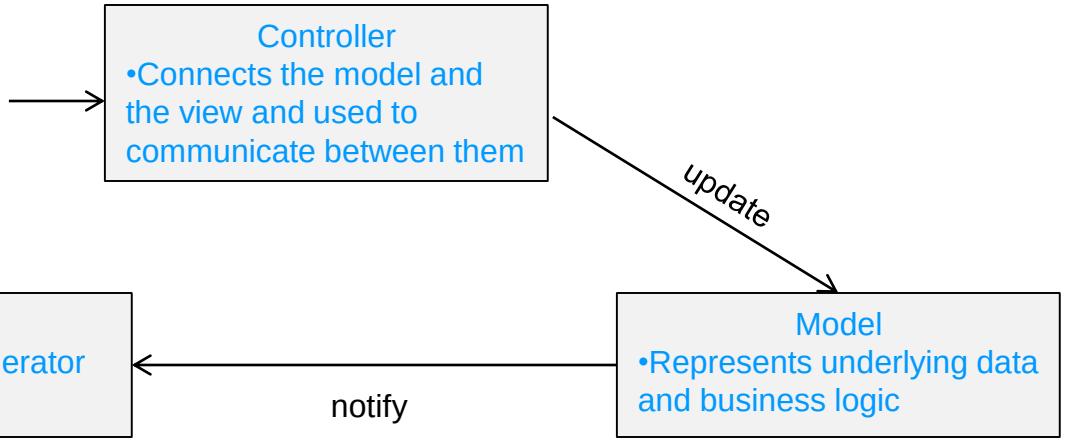
Marshalling and un-marshalling can be done with JSONs

Allows to use narrower protocol for object description

Got its native support in Java Script

Architecture

MVC Model 2



Classic MVC

Designed to separate client flow and interaction from business model that serves the client request and from the final output.

Traditional MVC as used in web modules – MVC Model 2

JOHN BRYCE
Leading in IT Education
a matrix company

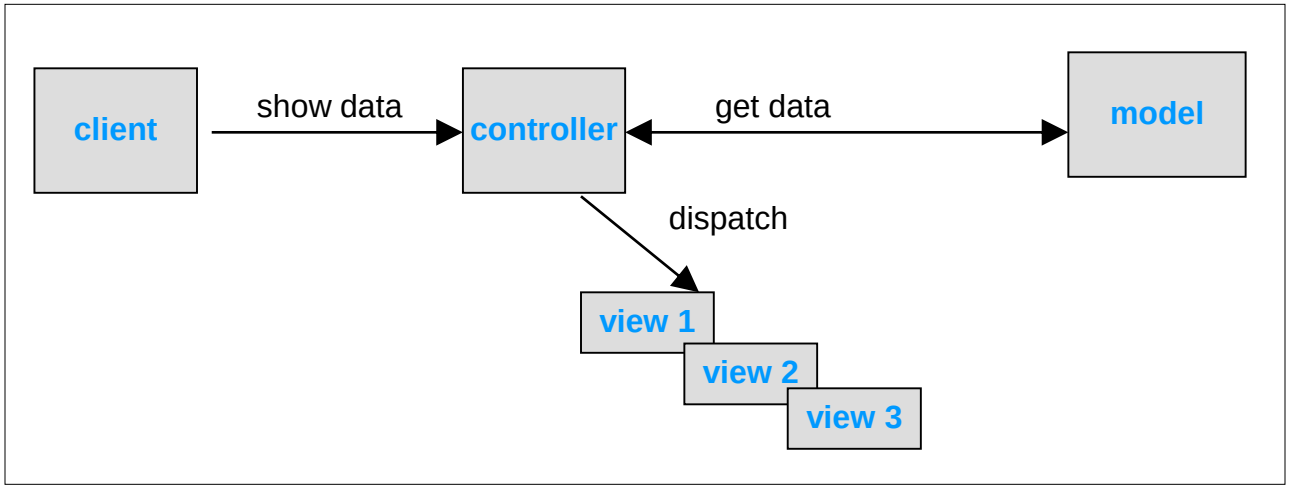
MVC Model 2

In web development:

- Controller – Web object that manages and handles client submit
- View – the result that is presented to the client by the end of the interaction
- Model – the calculation and computing when handling client request (business logic)

Architecture

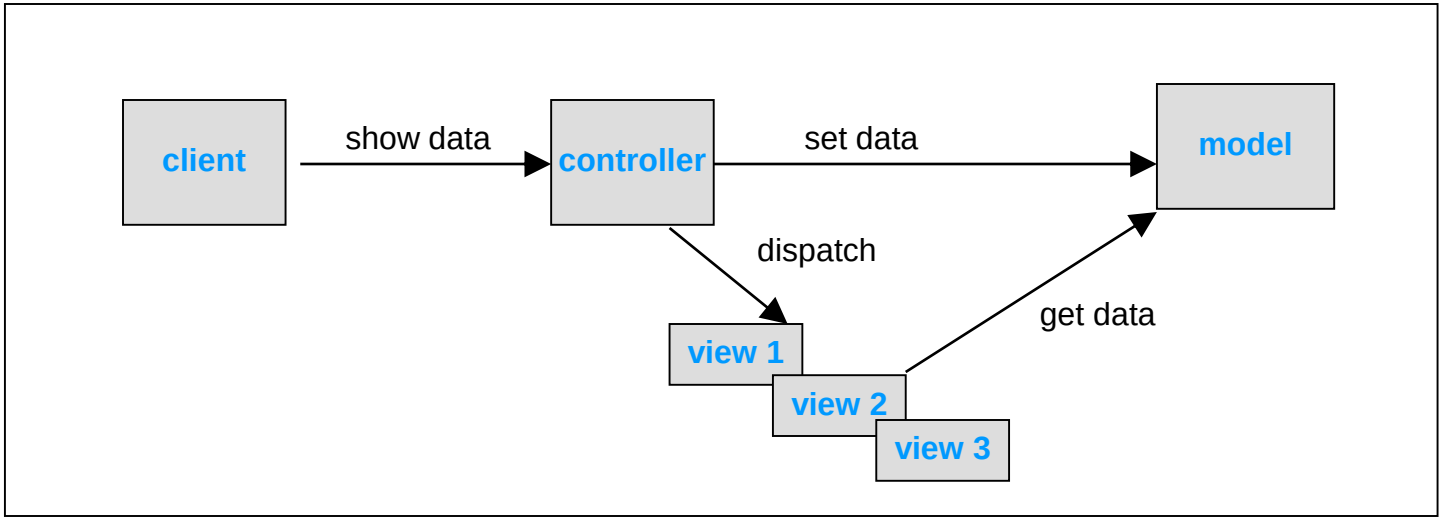
- Service To Work - MVC Model 2



- Better since:
  - View and model communicated through value objects
  - Trivial server side code is embedded in views
- But, views are still a mix of server & client code...

JOHN BRYCE
Leading in IT Education
a matrix company

Dispatcher View model



- Here, views are used also for ‘controlling’.
- Tightly coupling between views and model
- Can be considered for very simple modules

Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

The problem with views

Mixing server side code in view causes some serious problems:

- Value objects embedded in HTML
- It is never just HTML…(CSS, Jscript…)
- What if client requires something else than HTML ??

Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

The problem with views

List of web frameworks that should help:

Echo, Cocoon, Millstone, OXF, Struts, SOFIA, Tapestry, WebWork, RIFE, Spring MVC, Canyamo, Maverick, Jpublish, JATO, Folium, Jucas, Verge, Niggle, Bishop, Barracuda, Action Framework, Shocks, TeaServlet, wingS, Expresso, Bento, jStatemachine, jZonic, OpenEmcee, Turbine, Scope, Warfare, JWAA, Jaffa, Jacquard, Macaw, Smile, MyFaces, Chiba, Jbanana, Jeenius, Jwarp, Genie, Melati, Dovetail, Cameleon, Jformular, Xoplon, Japple, Helma, Dinamica, WebOnSwing, Nacho,

Cassandra, Baritus, Stripes, Click, GWT, Apache Wicket

So many… means that:

- none is really good enough…
- maybe problems can't be solved with MVC model 2

Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

The problem with views

- AJAX – bigger than it seems…
- AJAX technology encourages web modules to ‘talk’ using XML / JSON rather than HTML

Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

Introduction to AJAX

- **A**synchronous **J**script **A**nd **X**ML
- AJAX is based on *XMLHttpRequest* Object
  - Is an interface implemented by a scripting engine
  - Allows scripts to perform HTTP client functionality
  - W3C standard

Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

Introduction to AJAX

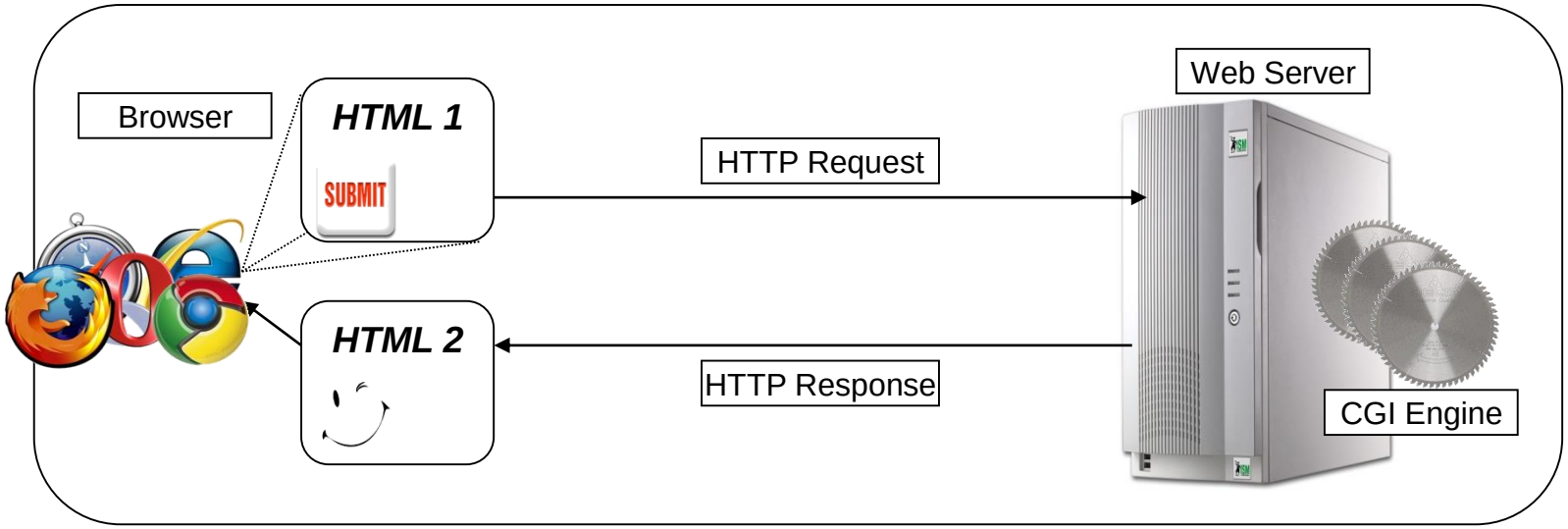
Classic way of interacting in web applications:

- Page by page
- Each page links or submits to another
- Static or dynamic content produced by the server
- Client side manipulation are done on downloaded data
- Most of client state is kept on server side

Architecture

Introduction to AJAX

View of classic architecture



Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

Introduction to AJAX

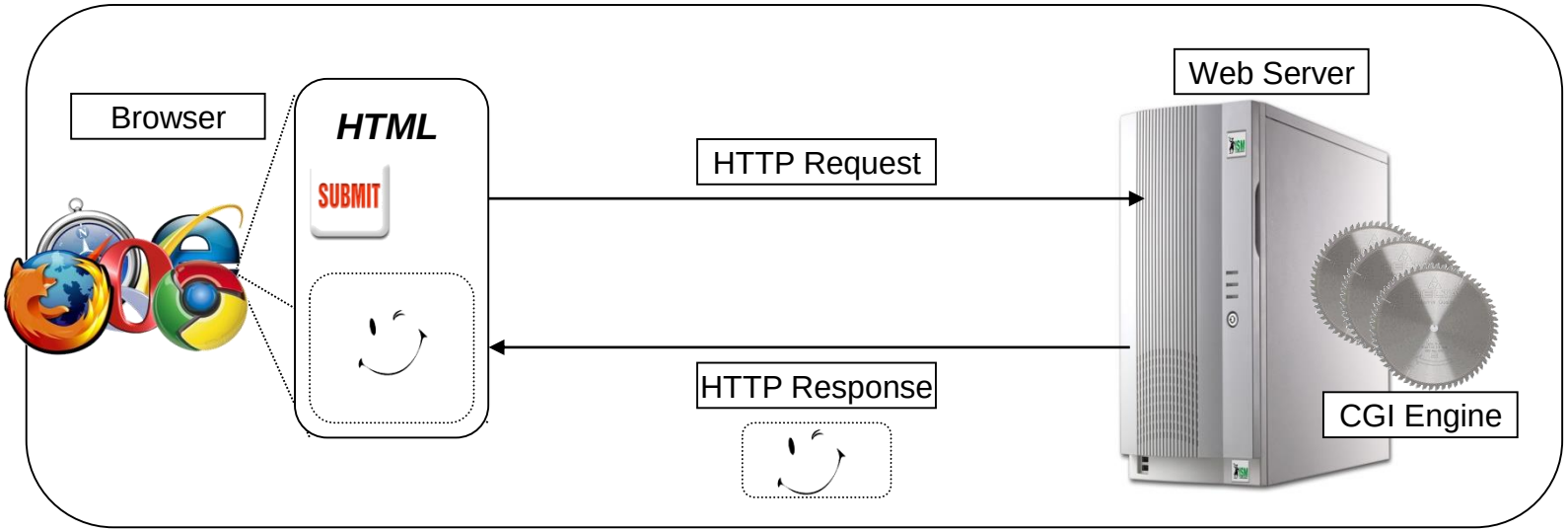
AJAX way of interaction:

- Same page generates request(s) & processes responses
- Dynamic content handled also by the client
- Client downloads only the data he needs
- Client is notified asynchronously regarding data receiving

Architecture

Introduction to AJAX

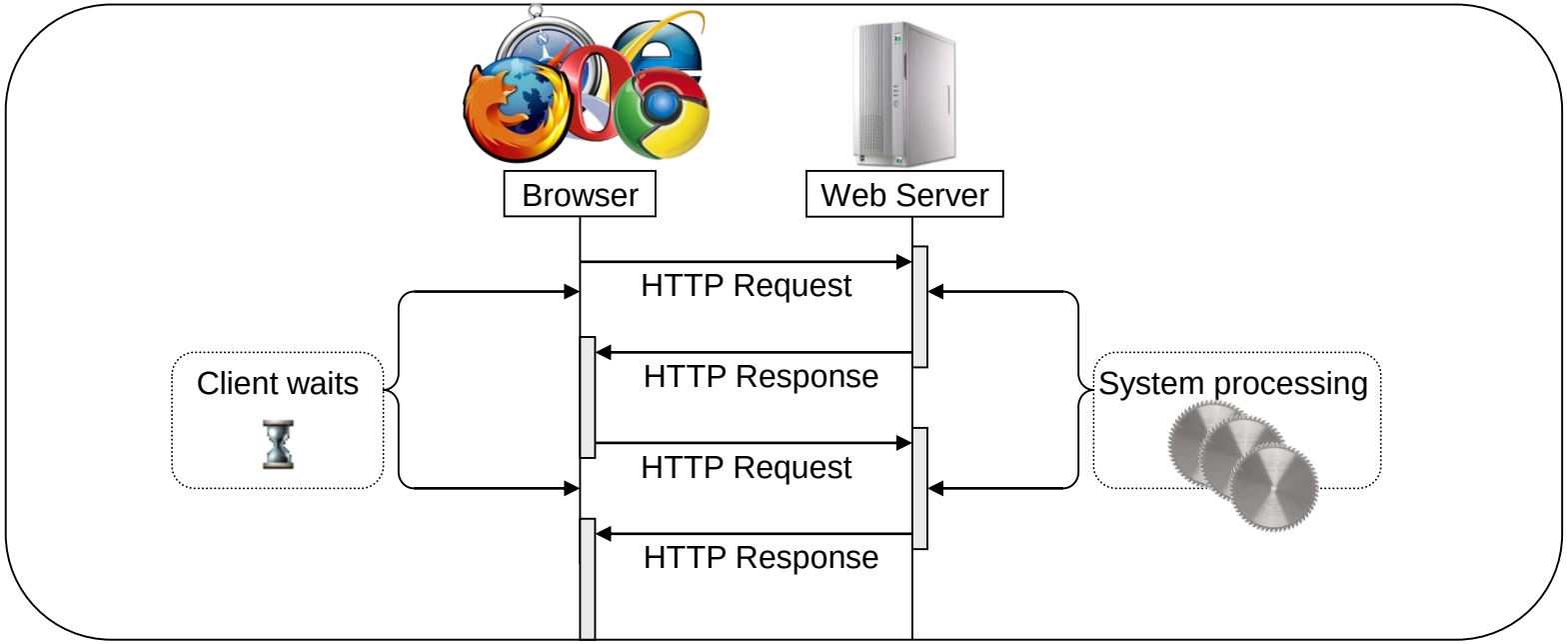
View of AJAX architecture



JOHN BRYCE
Leading in IT Education
a matrix company

Introduction to AJAX

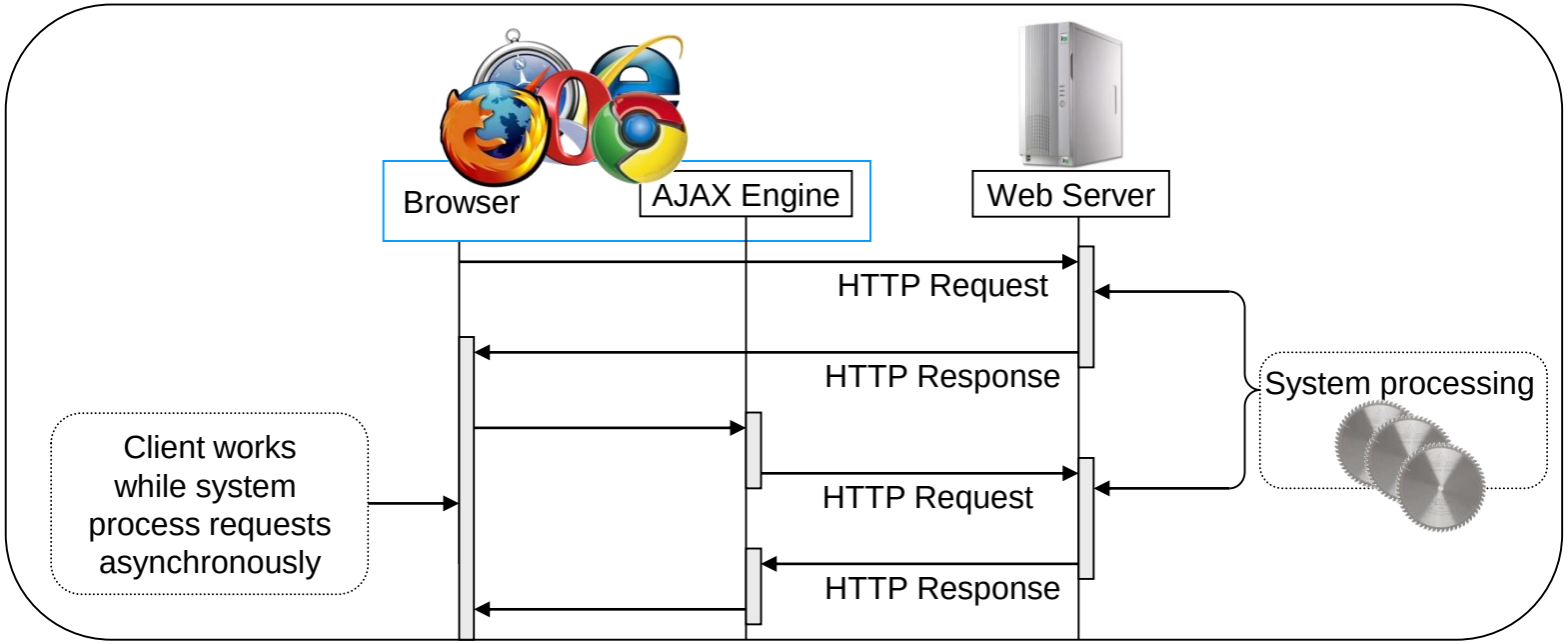
Classic interaction flow:



JOHN BRYCE
Leading in IT Education
a matrix company

Introduction to AJAX

AJAX interaction flow:



Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

Introduction to AJAX

url – the address of this ajax call. May target a Servlet or JSP. May send parameters just like any HTTP request

When a response is received stateCange() function will be asynchronously called

open() - Setting request data format:
- method (GET/POST)
- url
- asynchronous call – true enables it & is the default value

Send method takes a DOM Object.
DOM object hosts XML documents and Fragments available via DOM API.
Null value is also permitted, usually when string values are sent as a request header.

stateChange() method will be called asynchronously. Will be explained later.

```
function generateRequest()
{
    var url="http://localhost:8080/ajax";
    url+="?command=doIt";
    xmlhttp.onreadystatechange=stateChange;
    xmlhttp.open("GET",url,true);
    xmlhttp.send(null);
}

function stateChange()
{
    if (xmlhttp.readyState==4)
        // ...some code here...
    else
        alert("Problem retrieving XML data")
 }
…
```

Architecture

- Moving to single page applications

  - Using AJAX, web modules can focus on transferring data rather than view
    - Client receives a single HTML loaded with Jscript functions & callbacks
    - Jscript caller functions sends request data
    - Jscript callback processes response and renders it to page

    - Finally !
      - web modules input & output can be based on structured, self descriptive text formats
      - Future non-HTML clients may use the same modules & data

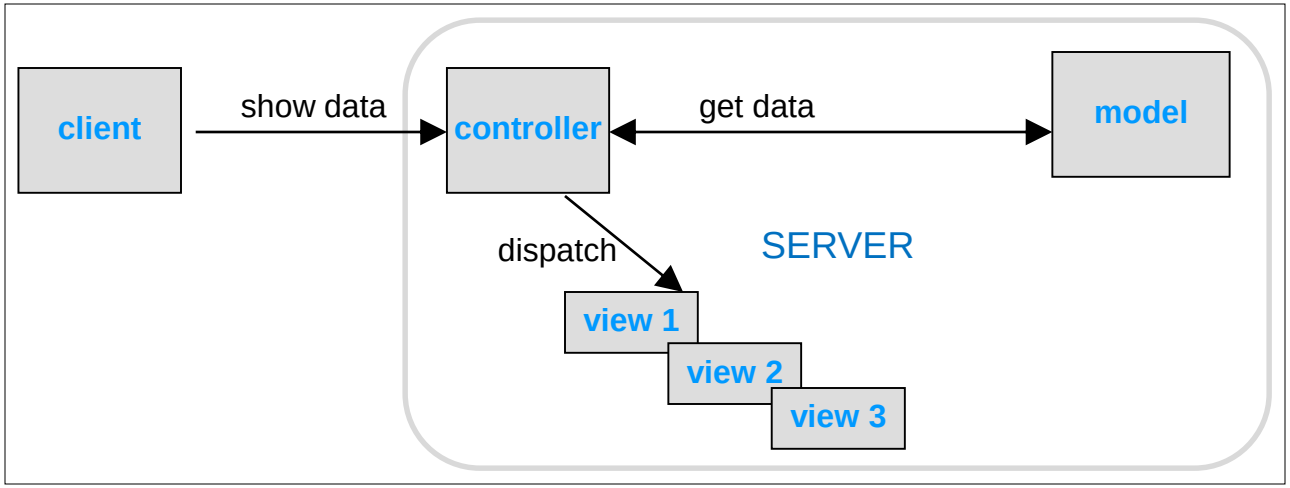
Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

Moving to single page applications

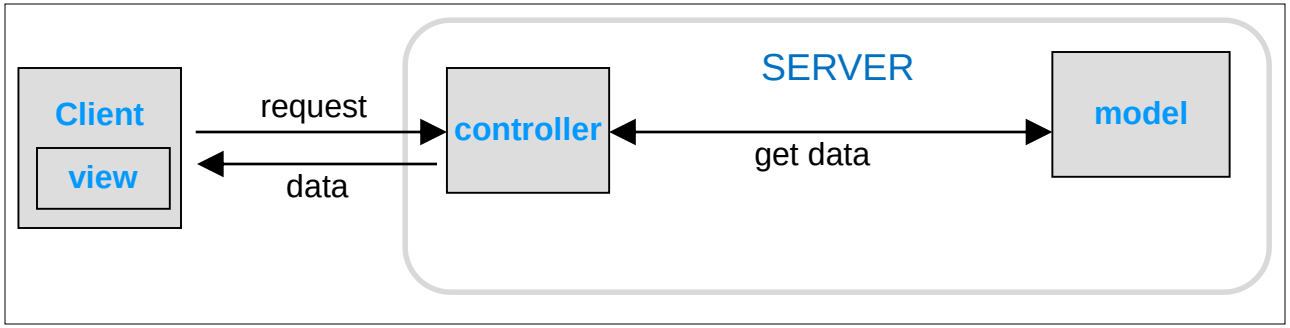
MVC Model 2

Views are generated on server side



The 'new' MVC

Views are handled by the client. Communication between client and server is based on data



Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

Future internet clients

- Why is it so important to ‘talk’ via XML/JSON and not ‘draw’ HTMLs ?

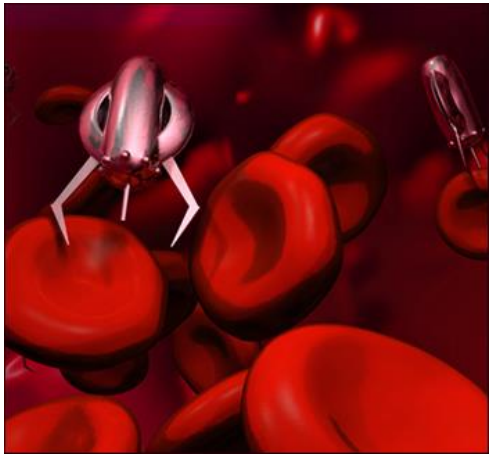
- Internet is much more than visiting web-sites…
- Future client of the internet are not going to use keyboards and screens… HTML might be irrelevant

Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

Future internet clients

- Phones & voice over IP networks
- Smart cards
- Chips
- Nanotechnology



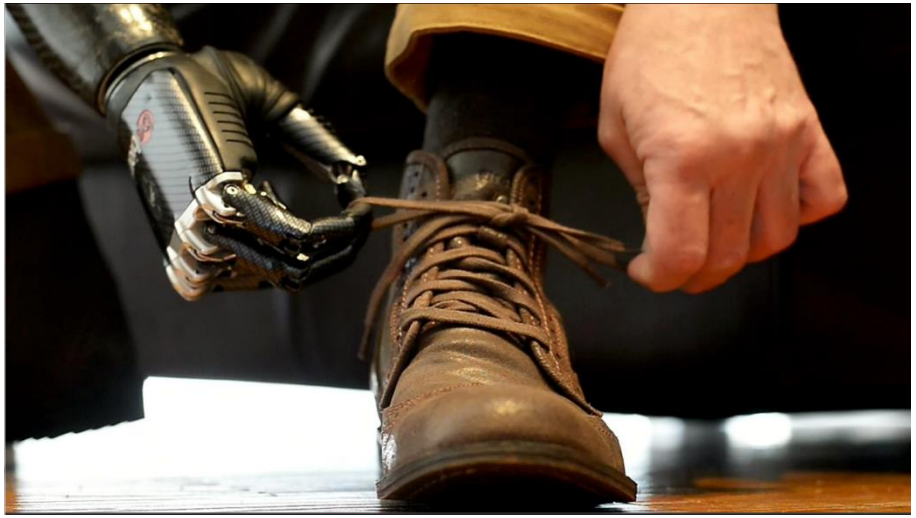
Architecture

JOHN BRYCE
Leading in IT Education
a matrix company

Future internet clients

But the new ultimate client is US

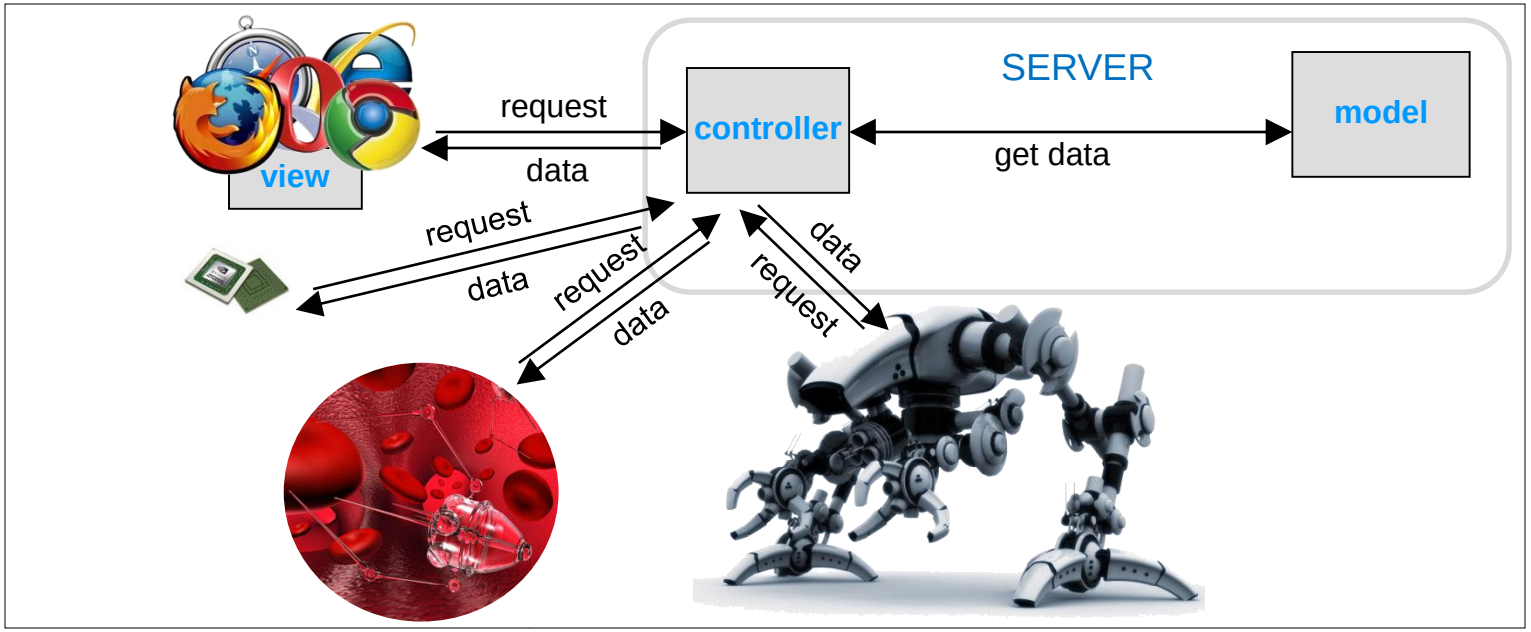
- No hardware, no UI – just us
- Ability to share data directly from & to our brains



Architecture

Future internet clients

The ‘new’ MVC



JOHN BRYCE
Leading in IT Education
a matrix company

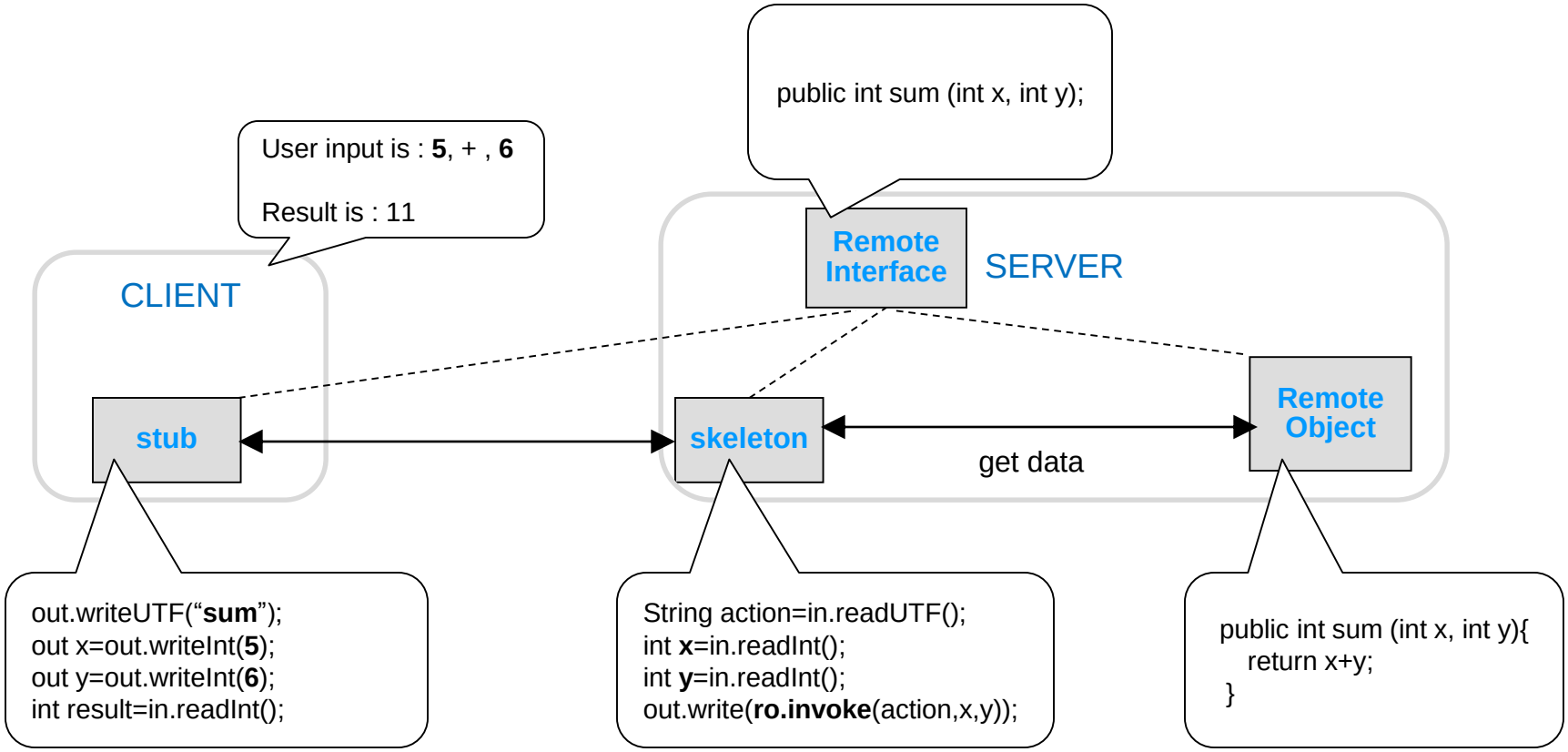
Architecture & Terms

- Remote Procedure Call
  - Client invokes method on a Remote Object over a network
  - Client obeys a contract which is the Remote Interface
  - Remote object is a resource
  - Remote method is a service

  - In order to communicate both client & server uses sockets
    - Socket communication is determined according to the remote interface
    - Stub - Client side socket
    - Skeleton – server side socket that is used as a proxy to the remote object

JOHN BRYCE
Leading in IT Education
a matrix company

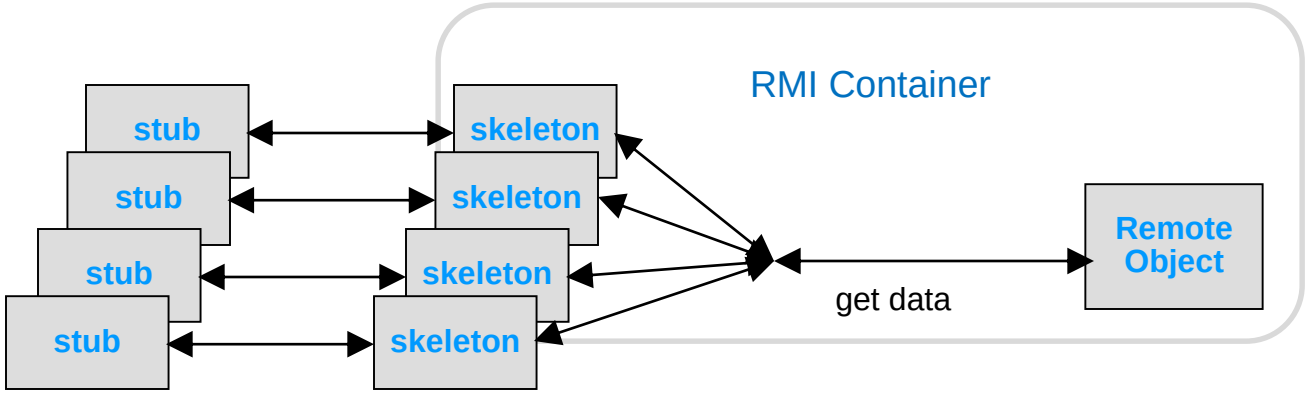
Architecture & Terms



RPC

RPC in JEE

basic RPC



No on-demand services…

- Remote object remains in memory even when there are zero clients…
- There is always only one Remote Object – no matter how many clients request it..

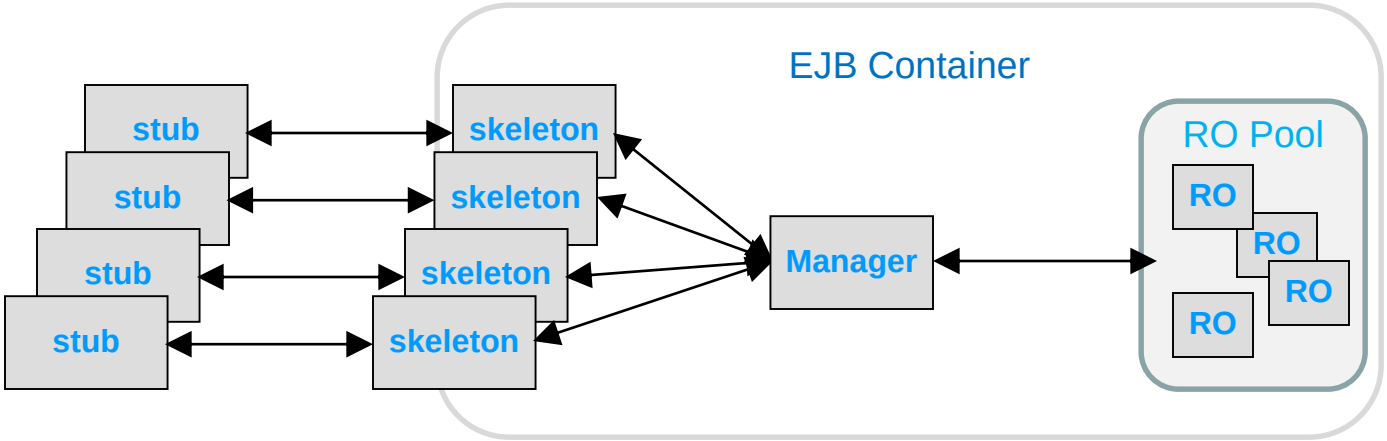
Good for limited concurrencies and complexities

For small / medium developments

JOHN BRYCE
Leading in IT Education
a matrix company

RPC in JEE

- RPC in IT/Enterprise



- Here we got a server implementation for a remote object manager
- This manager is responsible for all real-time decision making
  - Goal is to keep an optimized number of remote object for maximizing
    - Performance
    - Memory usage
    - Response time

JOHN BRYCE
Leading in IT Education
a matrix company

RPC Framework Requirements

- For server development
  - Map services to generate a contract
  - Expose the contract
  - Instantiate & publish the resource (or resource pool)
  - Create skeleton when requested
- For client development
  - Generate stub according to a given contract

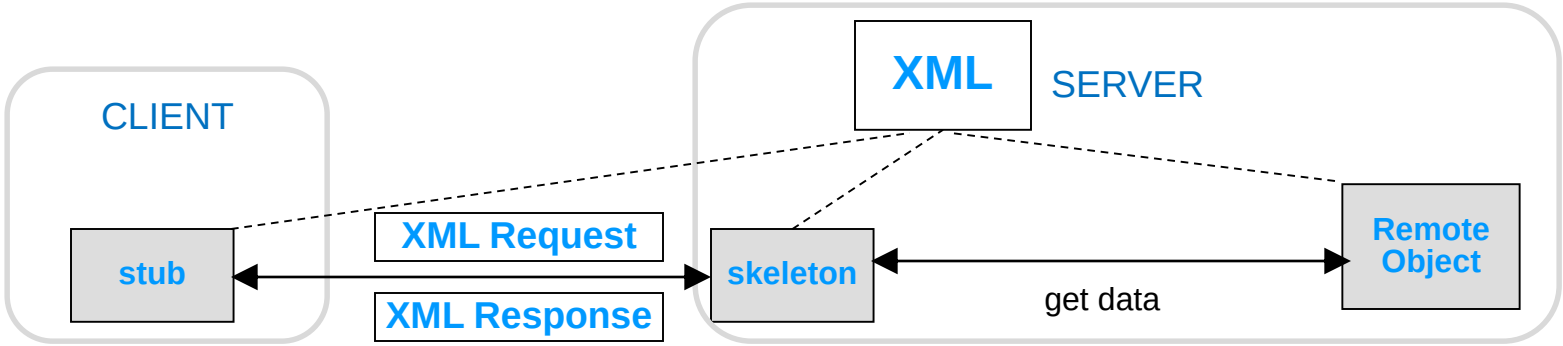
JOHN BRYCE
Leading in IT Education
a matrix company

XML for RPC – Web Services

- Main goal of XML is for application integration
- If the contract is in XML format it can be:
  - describing services written in any language
  - used by any client

- If the stubs & skels will ‘talk’ via XML:
  - each may be written in a different language
  - xsd types can be used to describe primitives & objects

XML for RPC – Web Services



XML based Web Services

- XML based RPC
- WSDL
  - Role
  - Structure
- SOAP
  - Role
  - Structure
  - SOAP over HTTP
  - SOAP action
- JAX-WS
  - Creating a Java service
  - Publishing & testing
  - Using *wsimport* for generating clients
  - JEE support

XML based Web Services

JOHN BRYCE
Leading in IT Education
a matrix company

XML based RPC

- Uses XML standard for contracts
- Uses XML to call remote objects and get response
- XML may be transferred over HTTP
- XML may be passed trough TCP/IP directly
- For asynchronous services (service that result with void)
  - XML can be sent as JMS text message

XML based Web Services

JOHN BRYCE
Leading in IT Education
a matrix company

- WSDL - stands for Web Services Description Language
  - Describes a resource & its services
  - Specifies the location
  - Details types and structure used to interact with the services
  - Provides information regarding binding style for generating stubs
    - Inner classes
    - Separate classes

XML based Web Services

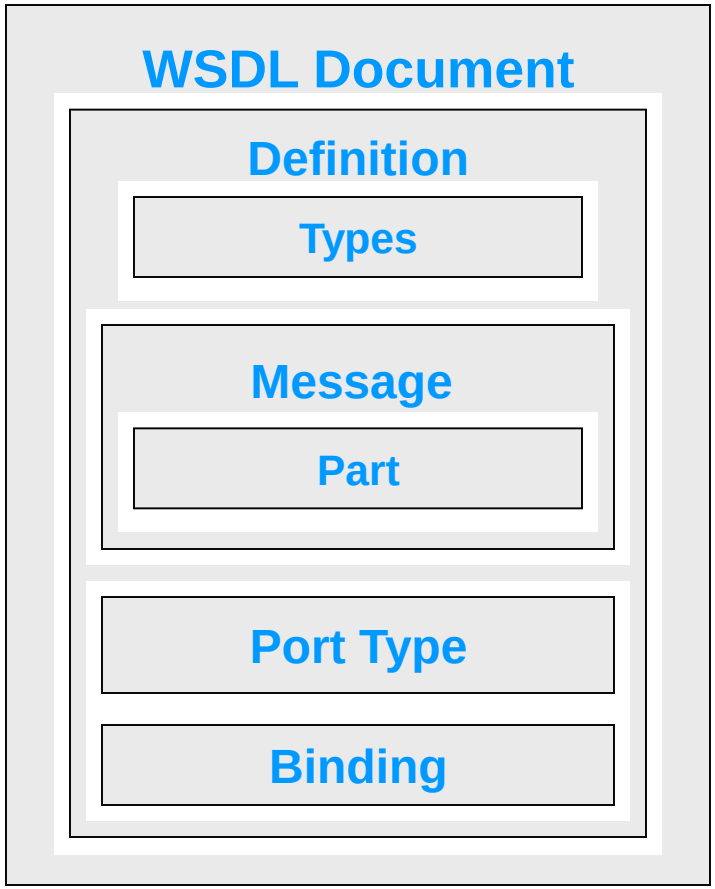
JOHN BRYCE
Leading in IT Education
a matrix company

WSDL Structure

Types	• types are used to specify complex parameters format
Part Message	• parts are used to specify the message parameters • message is an operation signature. for input - output mode, two messages are required
Port Type Operation	• port is used for defining message flow (operation) • operation defines input & output messages
Binding Service	a link between a service and the SOAP message that generated. specifies: • SOAP Action name • input & output encoding • SOAP version in use

JOHN BRYCE
Leading in IT Education
a matrix company

WSDL Structure



JOHN BRYCE
Leading in IT Education
a matrix company

WSDL Example

This WSDL defines the following service:
- client sends a stoke symbol in string format
- client gets the stoke value in float format

```
<?xml version="1.0"?>
<definitions name="StockQuote"
          targetNamespace="http://example.com/stockquote.wsdl"
          xmlns:tns="http://example.com/stockquote.wsdl"
          xmlns:xsd1="http://example.com/stockquote.xsd"
          xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
          xmlns="http://schemas.xmlsoap.org/wsdl/">

< types>
   <schema targetNamespace="http://example.com/stockquote.xsd"
         xmlns="http://www.w3.org/2000/10/XMLSchema">
       <element name="TradePriceRequest">
           <complexType>
             <all> <element name="tickerSymbol" type="string"/> </all>
           </complexType>
       </element>
       <element name="TradePrice">
           <complexType>
              <all><element name="price" type="float"/></all>
           </complexType>
        </element>
    </schema>
</types>…
```

String

Float

Types – defines the request & response parameters format

XML based Web Services

JOHN BRYCE
Leading in IT Education
a matrix company

WSDL Example

```
<message name="GetLastTradePriceInput">
    <part name="body" element="xsd1:TradePriceRequest"/>
</message>
```

String

```
<message name="GetLastTradePriceOutput">
    <part name="body" element="xsd1:TradePrice"/>
</message>
```

Float

Message – defines a message infrastructure (parts)

```
<portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
        <input message="tns:GetLastTradePriceInput"/>
        <output message="tns:GetLastTradePriceOutput"/>
    </operation>
</portType>
```

Port Type – defines operation as a messages flow

JOHN BRYCE
Leading in IT Education
a matrix company

WSDL Example

```
<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="GetLastTradePrice">
            <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
            <input>  <soap:body use="literal"/>  </input>
            <output> <soap:body use="literal"/> </output>
        </operation>
</binding>

<service name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
        <soap:address location="http://example.com/stockquote"/>
    </port>
</service>

</definitions>
```

Binding – sets up the SOAP structure for a pre-defined operation

Service – specifies the WS name and address for SOAP messaging

XML based Web Services

JOHN BRYCE
Leading in IT Education
a matrix company

SOAP

- Simple Object Access Protocol
- W3C Standard
- Defines a standard way to wrap RPC requests & responses
- Supports exceptions description (Faults)
- Will usually be sent over HTTP
- Can be unidirectional & bidirectional
- Can be synchronous & asynchronous
- SOAP Gateway is needed (skeletons in WEB tier)

XML based Web Services

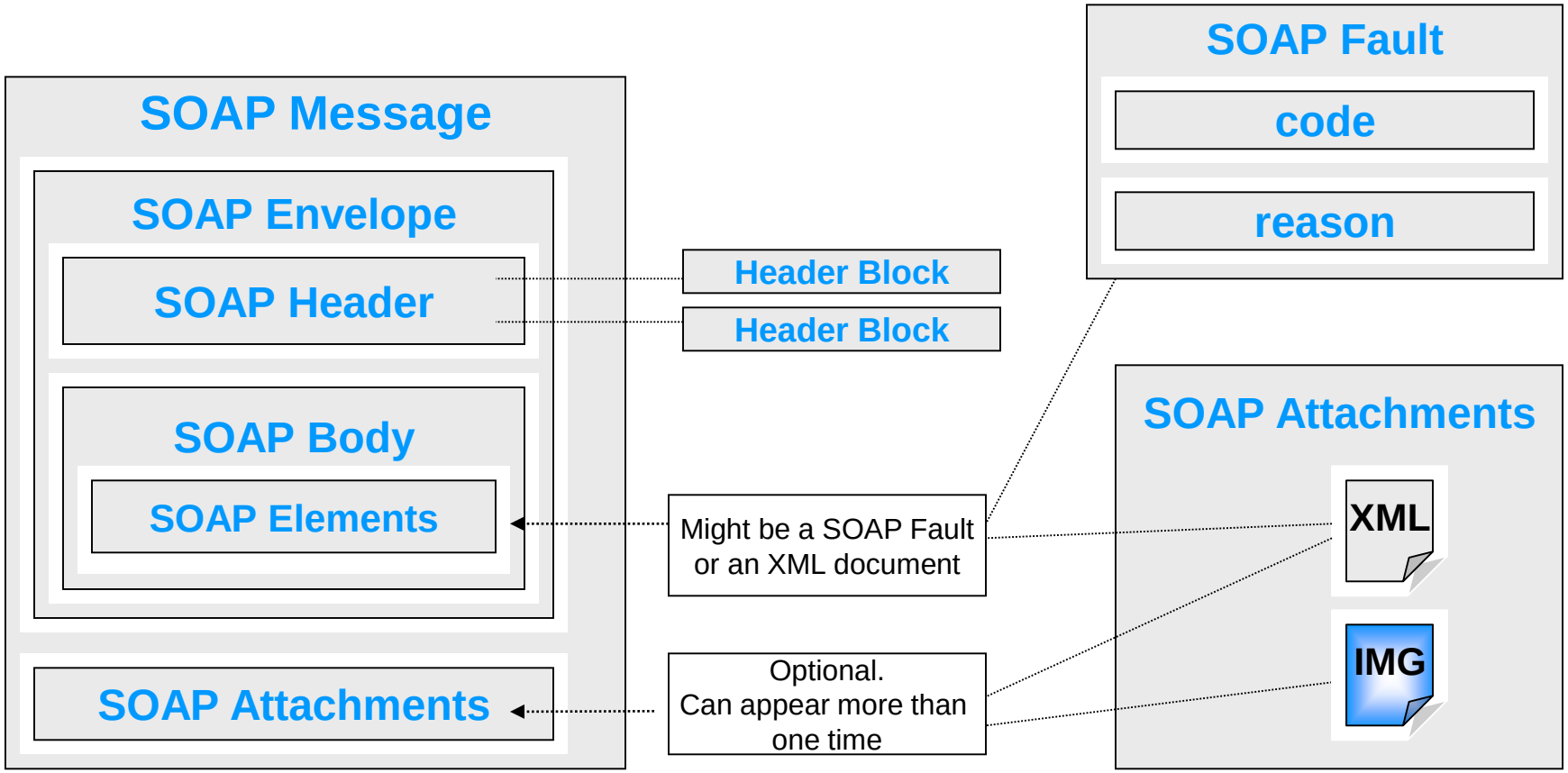
JOHN BRYCE
Leading in IT Education
a matrix company

Soap structure

- SOAP Envelope
- SOAP Head
- SOAP Body
- SOAP Element
- SOAP Fault
- SOAP Attachment

XML based Web Services

Soap structure schema



XML based Web Services

JOHN BRYCE
Leading in IT Education
a matrix company

SOAP over HTTP Request example

```
HTTP 1.0 POST
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI“
```

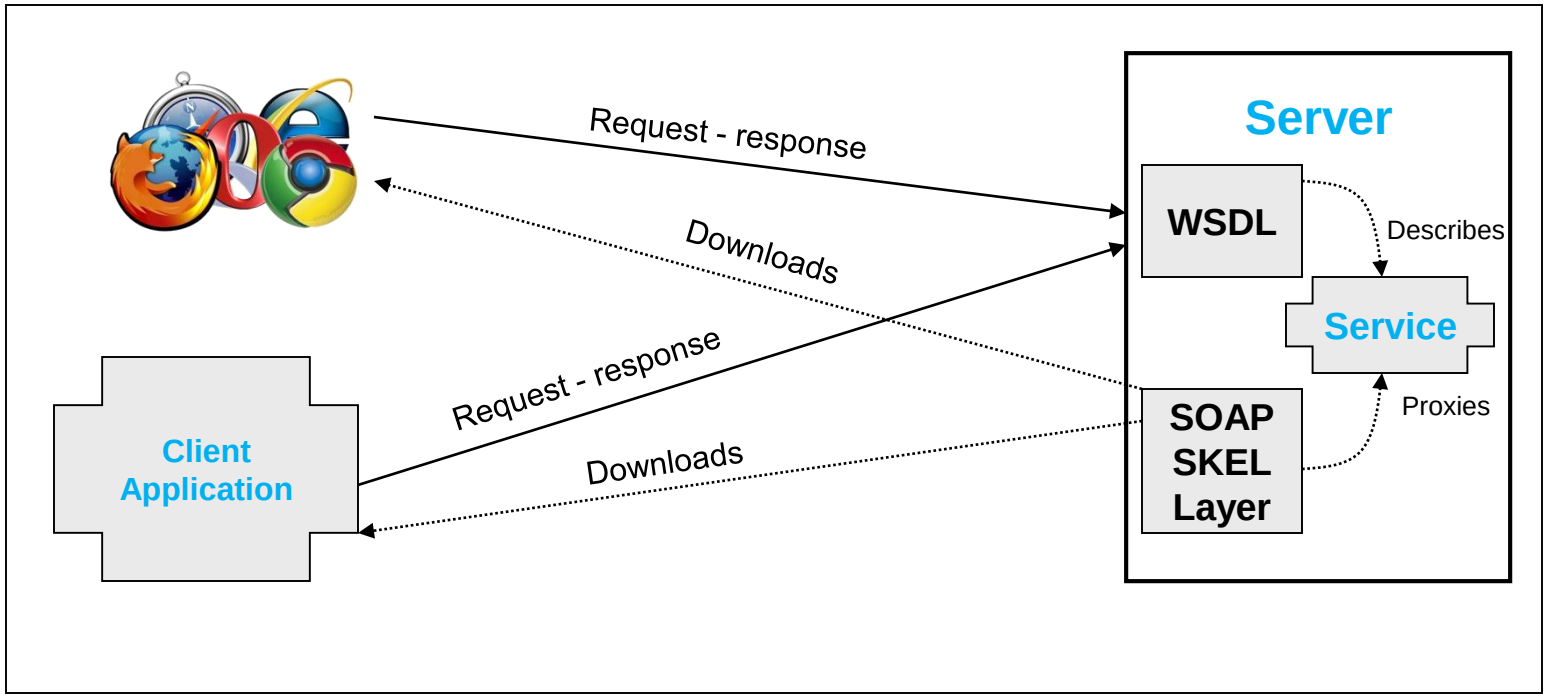
POST method - allows great amount of data upload

SOAP Header that provides another checking mechanism

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <SOAP-ENV:Body>
        <m:GetLastTradePrice xmlns:m="Some-URI">
            <symbol>DIS</symbol>
        </m:GetLastTradePrice>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

XML based Web Services

XML based RPC



REST based Web Services

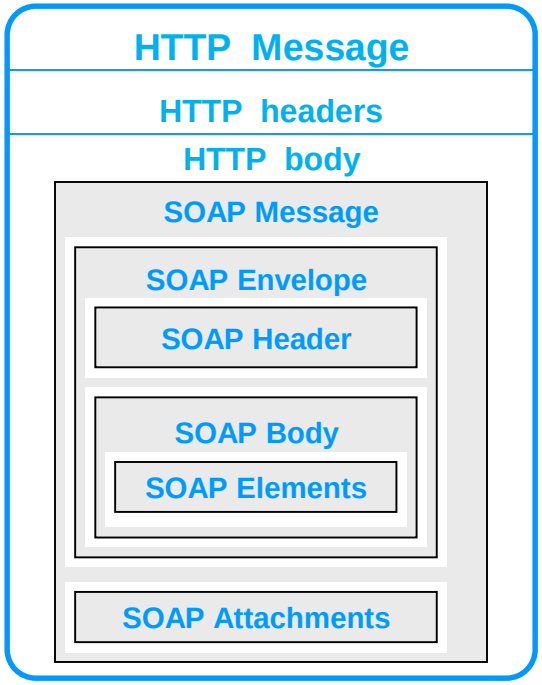
- Big data – the challenge
  - Parallel computing
  - NoSQL DBs
  - Saving bandwidth & faster response
- Introduction to REST
- HTTP for RPC
- RESTful
- WADL

REST based Web Services

JOHN BRYCE
Leading in IT Education
a matrix company

Big data – the challenge

- From single & concurrent to parallel & cloud computing
- Using NoSQL DB in addition to the classic relational DB
- Saving bandwidth & performance
  - XML is a very inefficient protocol uses tags to wrap data
  - XML forces the use of parsers
  - SOAP is an additional protocol on top of HTTP



REST based Web Services

- Introduction to REST
- REST – REpresentational State Transfer
  - is an HTTP ‘enrichment’ that provides advanced RPC
    - passing data in any format including XML, JSON and binary data
- REST can be counted as part of HTTP unlike SOAP which is a separate protocol

REST based Web Services

JOHN BRYCE
Leading in IT Education
a matrix company

HTTP for RPC

- Client may use the following HTTP features in order to invoke a service:
  - URI – path can determine the endpoint class and even method
  - ACCEPT header – used by the client to specify response MIME type
    - service methods may result in different MIME types
    - client call can be delegated to method that produces the MIME type it expects
  - METHOD – GET, POST, DELETE, PUT, HEAD
    - each method can be mapped to several HTTP-methods
    - client call is delegated to the method matches client HTTP request method

REST based Web Services

JOHN BRYCE
Leading in IT Education
a matrix company

HTTP for RPC

RESTful – a methodology of mapping HTTP methods to actual business logic

Suggested way to implement business according to HTTP method

HTTP Method	Single element	Collection
GET	Fetch an element from a collection	Fetch the whole collection
PUT	Replace or create new element in a collection	Override one collection with a new one
POST	Assign a value to an object	Add new value to a collection
DELETE	Delete a specific element from a collection	Delete the entire collection

REST based Web Services

- WADL
- Main elements
  - <application> - the root element
  - <grammars> - includes *.xsd which may describe response content (if it
  - is XML based) – optional
  - <resources> - contains all the resources described in the document
  - <resource> - describe the resource itself, holds the path to it
    - <method> - describes a method for the invocation (GET/POST….)
      - <request> - describes the request and response structure
        - <param> <option> - describe a parameter name, type (xsd:) and optional values
        - <representation> - specifies request body MIME-TYPE
      - <response> - describes the response status code and its MIME type
        - <representation> - specifies response body MIME-TYPE

REST based Web Services

JOHN BRYCE
Leading in IT Education
a matrix company

WADL – simple example

```
<application xmlns="http://wadl.dev.java.net/2009/02">
        <resources base="http://example.co.il/rest">
                <resource path="employees">
                                <method name="GET"/>
                                <method name="POST"/>
                </resource>
        </resources>
 </application>
```

Calling to http://example.co.il/rest/employees is available via POST & GET
POST – probably add a new employee
GET – fetch employee list

REST based Web Services

JOHN BRYCE
Leading in IT Education
a matrix company

WADL - placeholder for path-param

```
<application xmlns="http://wadl.dev.java.net/2009/02">
        <resources base="http://example.co.il/rest">
                <resource path="employees">
                                <resource path="{empId}">
                                                <param  required="true" name="empID"/>
                                                <method name="GET"/>
                                </resource>
                </resource>
        </resources>
 </application>
```

Calling to http://example.co.il/rest/employees/1234
is available via GET
empId will get the value 1234 and delegates it to the empID
method parameter

JOHN BRYCE
Leading in IT Education
a matrix company

WADL - placeholder for query-param

```
<application xmlns="http://wadl.dev.java.net/2009/02">
        <resources base="http://example.co.il/rest">
                <resource path="employees">
                        <method name="GET">
                                <request>
                                                <param  required="false" default="1" name="empID"/>
                                </request>
                        </method>
                </resource>
        </resources>
 </application>
```

Calling to http://example.co.il/rest/employees?1234
is available via GET
empId will get the value 1234 and delegates it to the empID
method parameter