Server-Side Web Development

JSP

Lecture #3 2007

1. Web Technologies
   - Internet
   - WWW
2. Protocols
   - TCP/IP
   - HTTP
3. Server-Side Web Development
   - Web Applications
   - Java Servlets

Web Technologies

- Markup & presentation (HTML, XHTML, CSS etc)
- Data storage & access (JDBC, XML etc)
- Network & application protocols (TCP/IP, HTTP etc)
- Programming & scripting languages (Java, JSP etc)

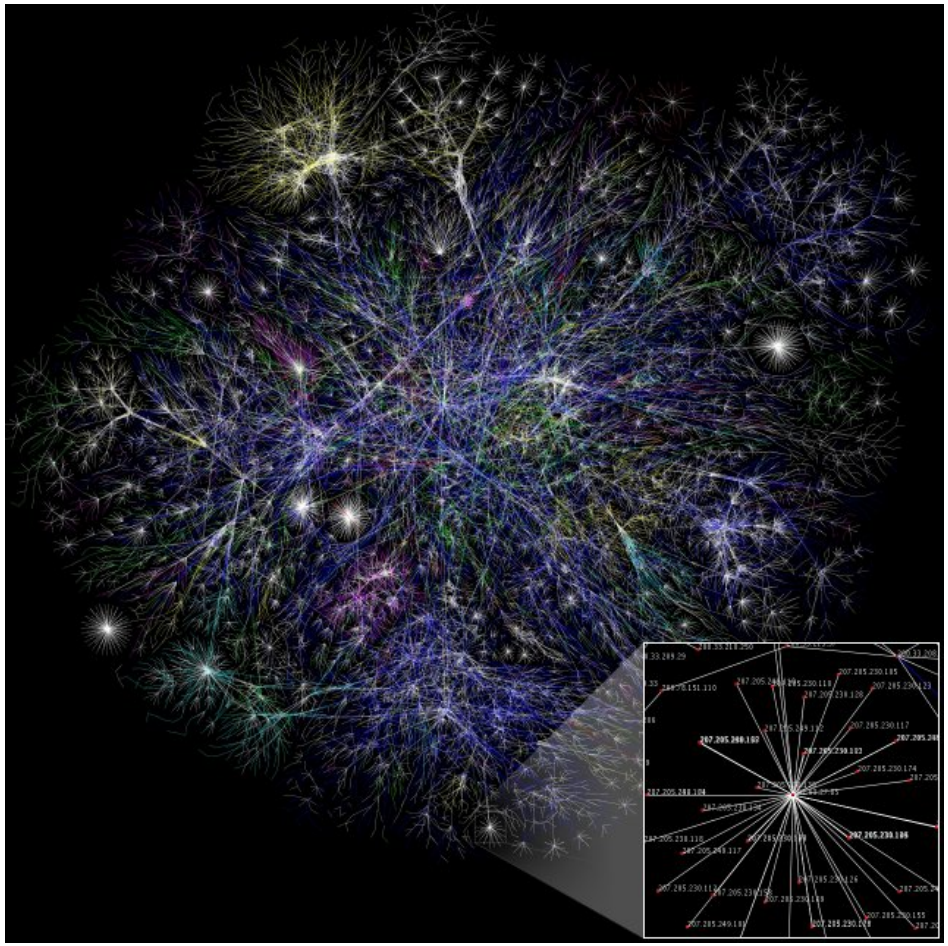
Internet

- Heterogenous network of networks
- More than 1 billion users
- Handles the web, e-mail, file transfers, instant messaging, telecommunication etc

Internet

- Internet Protocol (IP)
- Packet-oriented
- Uses IP-addresses (130.239.8.60)

Internet



World Wide Web (WWW)

- Web clients and servers
- More than 1 billion web pages on
- More than 100 million sites
- HTML, graphics and components
- Transfered via HTTP (over TCP/IP)

Hypertext Markup Language (HTML)

- Text-based
- Content + Markup
- Interpreted and visualized by browsers
- Constitutes a small part of the resulting network traffic

Extensible Hypertext Markup Language (XHTML)

- XML version of HTML
- Well-formed XML document
- Can be parsed by any XML tool
- Treated by browsers as a new version of HTML

Uniform Resource Locator (URL)

http://www.cs.umu.se:80/kurser/5DV076/SOM-07/index.html

Contains

- Protocol (http)
- Address (www.cs.umu.se)
- Port (80)
- Path (/kurser/5DV076/SOM-07/index.html)

Universally Unique Identifier (UUID)

20269f4c-9111-4778-8f78-249fea2b2e6e

- Generated from MAC-address, timestamps and random numbers
- Unique to a very high probability
- Used as unique filenames, database ids etc

Web Client

- Requests resources via HTTP
- Interprets and presents the data retrieved
- Can be anything that requests data via HTTP
- Usually a web browser
- Uses URLs to reference resources

Web Server

- Serves resources via HTTP
- Can be anything that serves data via HTTP
- Usually a dedicated server software
- May contain several pieces of software working together
- One server may serve several sites (and vice versa)
- Large sites often clustered for performance

Domain Name System (DNS)

- A network of DNS servers
- Resolves IP-addresses
- www.umu.se = 130.239.8.60
- 130.239.8.60 = www.umu.se

Proxy

Acts as an intermediary between clients and servers

A web proxy can, e.g.,

- Enforce access control
- Block, filter or alter information
- Cache information
- Anonymize traffic
- Perform network address translation

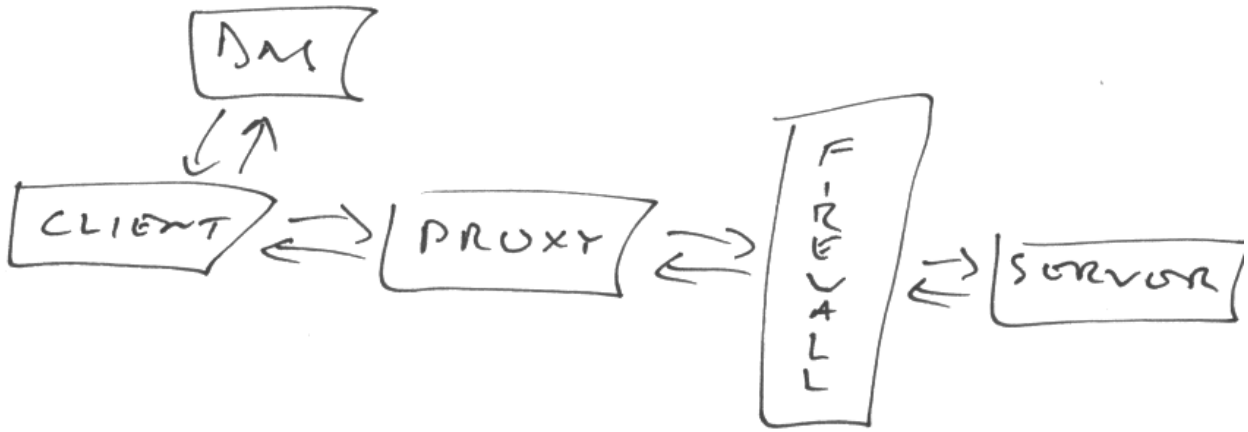
Firewall

Regulates traffic between networks

A firewall can, e.g.,

- Enforce access control
- Block, filter or alter connections
- Perform network address translation
- Be stateless, stateful or application-aware

Actors



Protocols

Specifies how information is exchanged in networks

- In which order
- In what format
- Transport level and application level protocols
- Built in protocol stacks

TCP/IP

- Packet-switched transport protocols
- Transport: Internet Protocol (IP)
- Reliable: Transmission Control Protocol (TCP)
- Unreliable: User Datagram Protocol (UDP)

- Each computer has (at least one) IP-address
- IP-addresses are resolved using DNS-systems
- Packets are routed from sender to receiver
- Packets can be lost, delayed or arrive in any order

Hypertext Transfer Protocol (HTTP)

- Text-based
- Application-level protocol (mostly used over TCP)
- Client-driven (requests and responses)
- Stateless (sessions are stored in cookies or rewritten URLs)
- Can handle text as well as binary data (encoded as text)

HTTP Request

- Request line (method + URI + protocol)
- Headers (request information)
- Body (optional)

HTTP GET Request Example

```
GET /kurser/5DV076/SOM-07/test.html HTTP/1.1
Host: www.cs.umu.se
```

HTTP Response

- Status line (protocol + status code + reason phrase)
- Headers (response information)
- Body (response data)

HTTP Response Example

```
HTTP/1.1 200 OK
Date: Tue, 15 May 2007 14:25:27 GMT
Server: Apache/2.0.54 (Unix)
Accept-Ranges: bytes
Content-Length: 50
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=ISO-8859-1

<html>
<head>
</head>
<body>
test
</body>
</html>
```

HTTP Request Methods

- **HEAD** - simulate a get request
- **GET** - retrieve resource
- **POST** - submit data to resource
- **PUT** - upload resource
- DELETE - delete resource
- TRACE - echo request
- OPTIONS - query server for supported methods
- CONNECT - create TCP/IP tunnel

HTTP HEAD Request

- Used to retrieve meta-information (headers)
- Simulates GET request
- No body (data) in response

HTTP GET Request

- Used to retrieve resources from web servers
- Parameters stored in query string (limited size)
- Default method for data retrieval

HTTP POST Request

- Used to submit (form) data to resources
- Data stored in request body
- Default method for data storage

HTTP PUT Request

- Used to upload resources
- Data stored (encoded) in request body

HTTP Response Status Codes

Directs client behavior

- 100-199 - informational
- 200-299 - request successful
- 300-399 - resource unavailable / moved
- 400-499 - client-side error
- 500-599 - server-side error

Server-Side Web Development

- Data access and logic rather than interfaces
- More technical than HTML/CSS design
- Several alternative technology platforms exists
- Dynamic content vs static content
- Content-driven solutions
- Focus on web applications
- Need for a structured programming model
- Need for a simpler way to do programming

Web Applications

- Applications with a web interface
- Ideal for thin-client solutions
- Suffers from limitations of the web media
- Usually combines techniques (e.g., JSP + AJAX)
- Several development frameworks available
- Clear trend towards generated web interfaces
- Usually session-oriented
- Usually deployed in WAR files

Web Archives (WAR)

- ZIP-file = compressed archive
- JAR-file = Java Archive (ZIP file with a manifest)
- WAR-file = JAR-file with web application information

Sessions

- Used to store data for a series of HTTP requests
- E.g., a shopping cart, user preferences, site history
- A session identifier is sent with each request
- The session identifier is used to locate the session
- Data is stored in the session context

Three Layer Architectures

1. Interface - web pages
2. Logic - software components (JavaBeans, EJB)
3. Data Access - databases

- Clean separation of concerns
- Scalable
- Support role-based development cycles
- Well suited for large sites and business logic integration

Two Layer Architectures

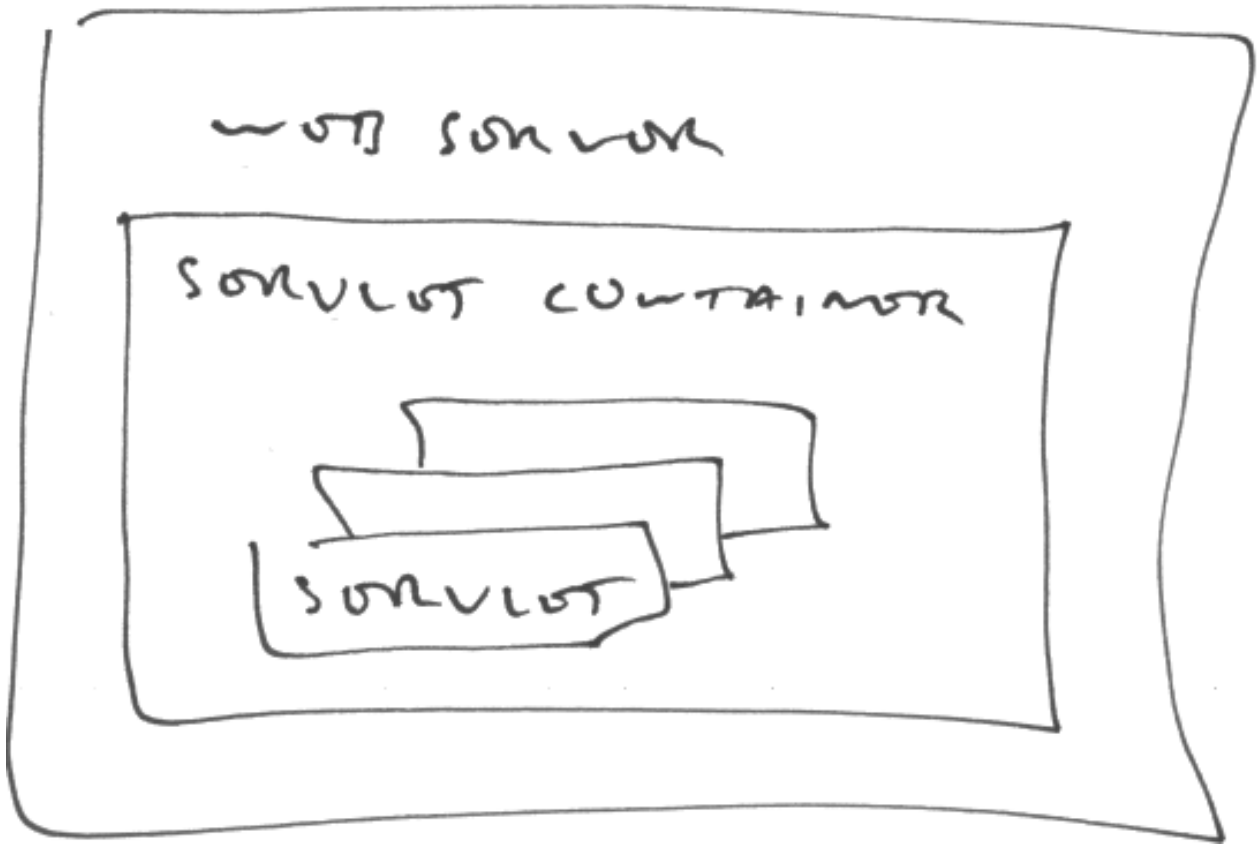
1. Interface (+ Logic)
2. Data Access + Logic

- Requires fewer software components
- Shorter development cycles for small development teams
- Better suited for smaller web sites

Java Servlets

- Java classes
- Implements the Java Servlet API interfaces (predates JSP)
- Receives a request and generates a response
- Can be written manually
- Must be thread-safe
- Usually generated automatically from JSP
- Hosted in a Servlet container

Java Servlets



The Servlet Lifecycle

1. init() - called on Servlet instantiation
2. service() - called for each request
3. destroy() - called on container shutdown

The Servlet service() method

- Part of a service-pattern
- Should not be implemented directly
- Inherit base class and implement handler methods
- Distinct handlers for each HTTP method (e.g., doGet())

1. service() parses request and determines HTTP method
2. service() calls appropriate handler method
3. Handler method processes request

Next Time

- JavaServer Pages (JSP)