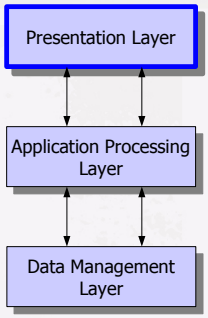
Presentation Layer

Dr. Alfredo Teyseyre - Dr. Alejandro Zunino
ISISTAN – UNICEN

Layered Application Architecture

- **Presentation layer**
  - Concerned with presenting the results of a computation to system users and with collecting user inputs
- **Application processing layer**
  - Concerned with providing application specific functionality ex., in a banking system, banking functions such as open account, close account, etc.
- **Data management layer**
  - Concerned with managing the system databases



Requirements (Web)

- Easy to change
- Separation of roles:
  - programmer
  - Web designer
- WYSIWYG
- As easy as desktop UI
- UI components
- MVC
  - model: data
  - view: how data is displayed
  - controller: user commands

Technological Alternatives (Java)

- J2EE standards:
  - Servlets: Java with HTML
  - JSP: HTML with Java
  - Taglibs: pre-built JSP components
- Almost standard:
  - Struts: command oriented
  - JSF: web framework (still JSP)
- Others:
  - GWT, Wicket, Vert.X, Spring MVC, Tapestry
  - Express, Meteor, Ionic, React, AngularJS
  - ...

Introduction to Servlets and JSP

Introduction

- Servlet Architecture Overview
- Servlets in Context
  - Other options for server side development
- Advantages of Servlets
- Introduction to Java Server Pages (JSP)
- Servlets vs. JSP

Architectural Overview

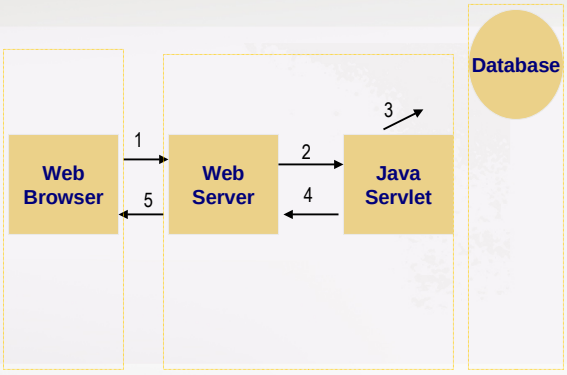
What is a Servlet?

- Java's answer to the Common Gateway Interface (CGI).
- **Servlet**: a java program that runs within the web server.
- The standard Java way for building web applications.

Life of a Servlet

- Regardless of the application, servlets usually carry out the following routine:
  1) Read any data sent by the user
     - Capture data submitted by an HTML form.
  2) Look up any HTTP information
     - Determine the browser version, user, cookies, etc.
  3) Generate the Results
     - Connect to databases, connect to legacy apps, etc.
  4) Format the Results
     - Generate HTML on the fly
  5) Set the Appropriate HTTP headers
     - Tell the browser the type of document being returned or set any cookies.
  6) Send the document back to the client

Life of a Servlet



Servlets in Context

Server Side Options

- There are many options for creating server side applications.
- We will examine some of these options briefly.
- This better enables us to understand servlets within the broader context of web development.
- Also enables us to better understand the advantages and disadvantages of servlets.

Server Side Options

- Common Gateway Interface (CGI)
- PHP
- Java: Servlets/JSP, JSF, ...
- ASP.NET
- NodeJS
- ...

Common Features

- All server side frameworks share a common set of features:
  - Read data submitted by the user
  - Generate HTML dynamically based on user input
  - Determine information about the client browser
  - Access Database systems
  - Exploit the HTTP protocol

Decision Points

- Ease of development:
  - How easily can you build new applications?
- Performance:
  - How fast can the framework respond to queries?
- Scalability:
  - Can the framework scale to thousands, millions of users?
- Security:
  - Are there any inherent security vulnerabilities?
- ...

Option: CGI

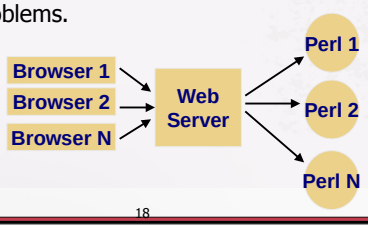
- Represents one of the earliest, practical methods for generating web content.
- Primarily written in the Perl programming language.
- Unfortunately, traditional CGI programs suffer from scalability and performance problems.
- Let's examine these two problems…

CGI Architecture

1) Browser initiates request
2) Web server receives the request.
3) For each request, web server spawns a new operating system process to execute the CGI/Perl Program.

CGI Architecture

- For each browser request, the web server must spawn a new operating system process:
  - takes time and memory
  - traditional CGI programs have performance and scalability problems.
  - Every other server architecture tries to address these problems.



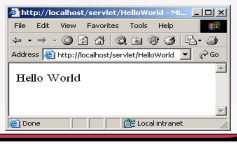
Option: PHP

- An open source project written entirely by volunteers
- Provides simple, but powerful database access.
- Also great for rapid development.
- For additional information: http://www.php.net
- No OO...

A Servlet That Generates Plain Text

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    out.println("Hello World");
  }
}
```



Java Server Pages

- Complementary to Java Servlets
- Can be used alone or in conjunction with servlets
- Represent (yet) another method for creating server side applications
- Servlets
  - code looks like a regular Java program.
- JSP
  - embed Java commands directly within HTML
- Let's examine a Servlet program next to a JSP program…
  - Each of these prints, "Hello, World!"

A Servlet: Looks like a regular Java program

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
  public void doGet(HttpServletRequest req, HttpServletResponse res)
   throws ServletException, IOException {
    res.setContentType("text/html");
    PrintWriter out = res.getWriter();
    out.println("<HTML>");
    out.println("<HEAD><TITLE>Hello World</TITLE></HEAD>");
    out.println("<BODY>");
    out.println("<BIG>Hello World</BIG>");
    out.println("</BODY></HTML>");
  }
}
```


A JSP Page: Looks like a regular HTML page

```
<html>
<head>
<title>Hello, World JSP Example</title>
</head>
<body>
 <h2> Hello, World!
The current time in milliseconds is
<%= System.currentTimeMillis() %>
</h2>
</body>
</html>
```


Summary

- Servlet: a java program that runs within the web server.
- Servlets have lots of advantages over other server side scripting options.
- Servlets look like regular Java code with some HTML.
- Java Server Pages look like HTML with some Java.