Web Server Architecture

Types of Web servers

- Static site
  - Web was originally about downloading static content from Web servers
    - Q: What should a Web site do to serve a static page for a request?
  - HTTP server (say, Apache) + Filesystem
    - Configure the mapping from a URL to a file
      - E.g., "DocumentRoot /var/www/html/" in Apache
- Dynamic site
  - A large part of Web content is dynamically generated now
    - Q: What happens when Google gets a query? What does it do?
    - Q: What happens when you log in Amazon?
    - Q: How does a server identify which process to run to generate the corresponding content given a request?
      - e.g., web.xml file on Tomcat

Four layers of Web site

- Encryption layer: encrypt transport
- HTTP layer: interpret request and serve response
- Application layer: generate dynamic content
- Storage/Data layer: store and retrieve data

Generating dynamic pages

How can we generate a dynamic Web page?

- Example: Hello, John! at http://oak.cs.ucla.edu/classes/cs144/examples/hello.html

  1. Programmatic approach: Write a program that prints out the HTML page!

     Example: Java Servlet for "Hello, John!"

```
protected void doGet(HttpServletRequest request,
                     HttpServletResponse response)
    throws ServletException, IOException
{
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head><title>Hello</title></head>");
    out.println("<body>Hello, " + request.getParameter("
       first_name") + "!");
    out.println("</body>");
    out.println("</html>");
    out.close();
}
```

2. Template approach: Write an HTML page that allows simple "variable substitution"!

   Example: Java ServerPages (JSP)

```
<html>
<head><title>Hello</title></head>
<body>Hello, <%= request.getParameter("first_name") %>!
</body>
</html>
```

- Q: What are the problems of the two approaches?
- Notes:
  - Even for template approach, once complex code gets embedded inside, the page gets ugly and becomes difficult to maintain
  - Code "ownership"
    - Often, page design is done by designers, while app coding is done by developers.
    - Who "owns" the above pages?
    - When multiple people "own" the same page, "conflicts" arise
      - Can we separate page design from programming logic?