

# Advanced Web Techniques

Incorporating Dynamic Content into  
Web Presentations

Paco Hope

# Goal of this Series

- Provide a clear **framework** to categorize existing technologies
- Provide clear **differentiation** between technologies
- Introduce the **basic structure** and syntax of three key technologies
- Provide **foundation** which can be supplemented by existing documentation

# Dynamic versus Static

## Static

- File contents served **as is**
- **No processing** done
- **Fastest** delivery (13 system calls in apache under Solaris)
- Benefits from caching

## Dynamic

- **Reacts to stimuli** (date, time, database, weather, user input)
- **Sometimes slower**
- Requires **computation** (sometimes an issue)
- Reduces effectiveness of caching

# Two Classes of Dynamism

## Client-side

(handled by web browser or software on local machine)

- Java
- JavaScript
- ActiveX
- Shockwave
- RealAudio (video, etc)

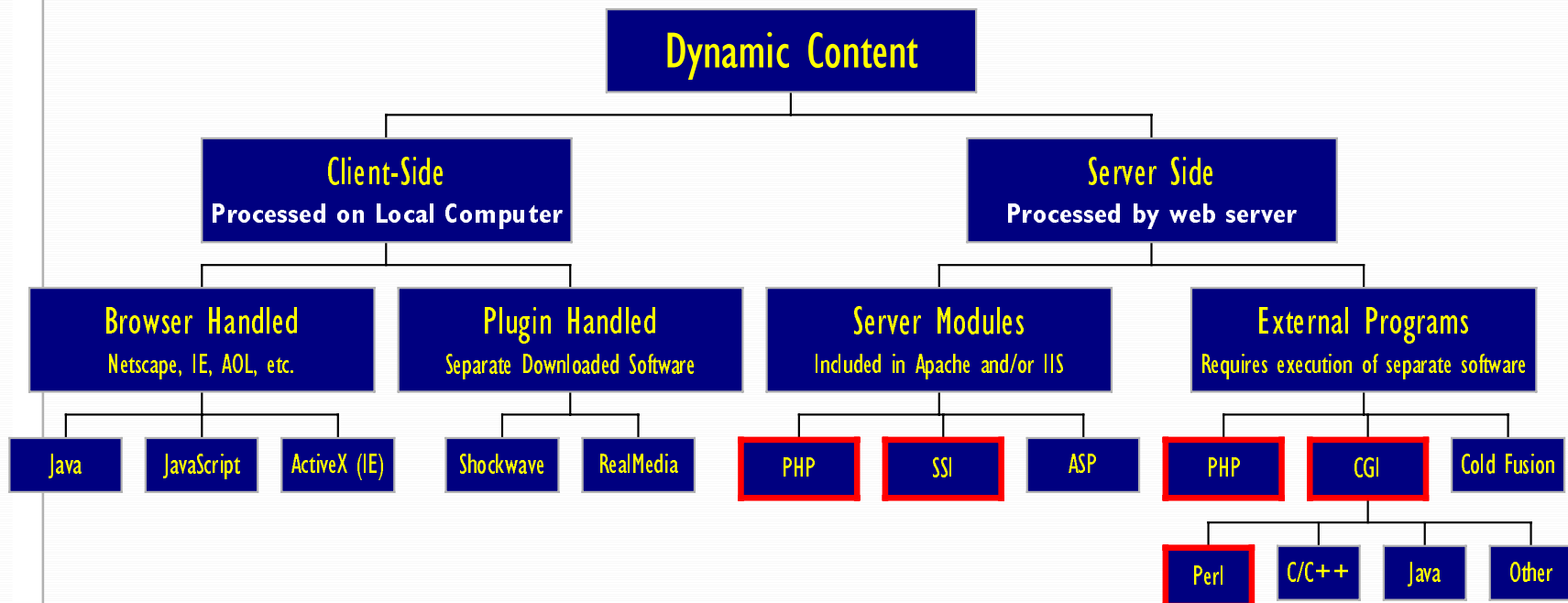
## Server-side

(handled by web server)

- CGI
- SSI
- PHP
- ASP
- Cold Fusion
- Domino

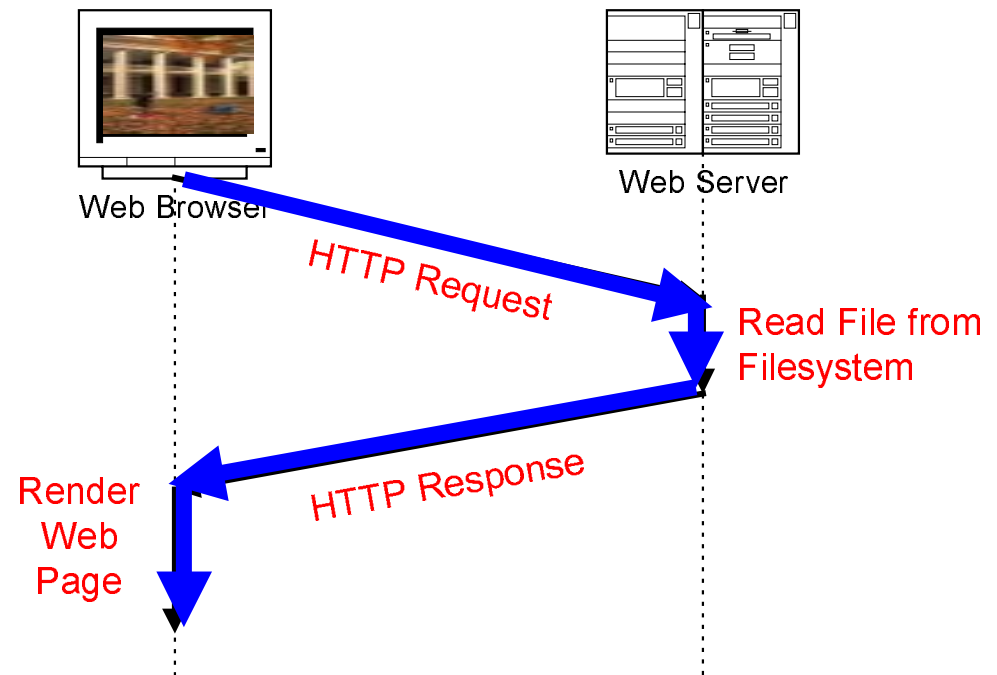
# Taxonomy of Technologies

## Classes of Dynamic Content



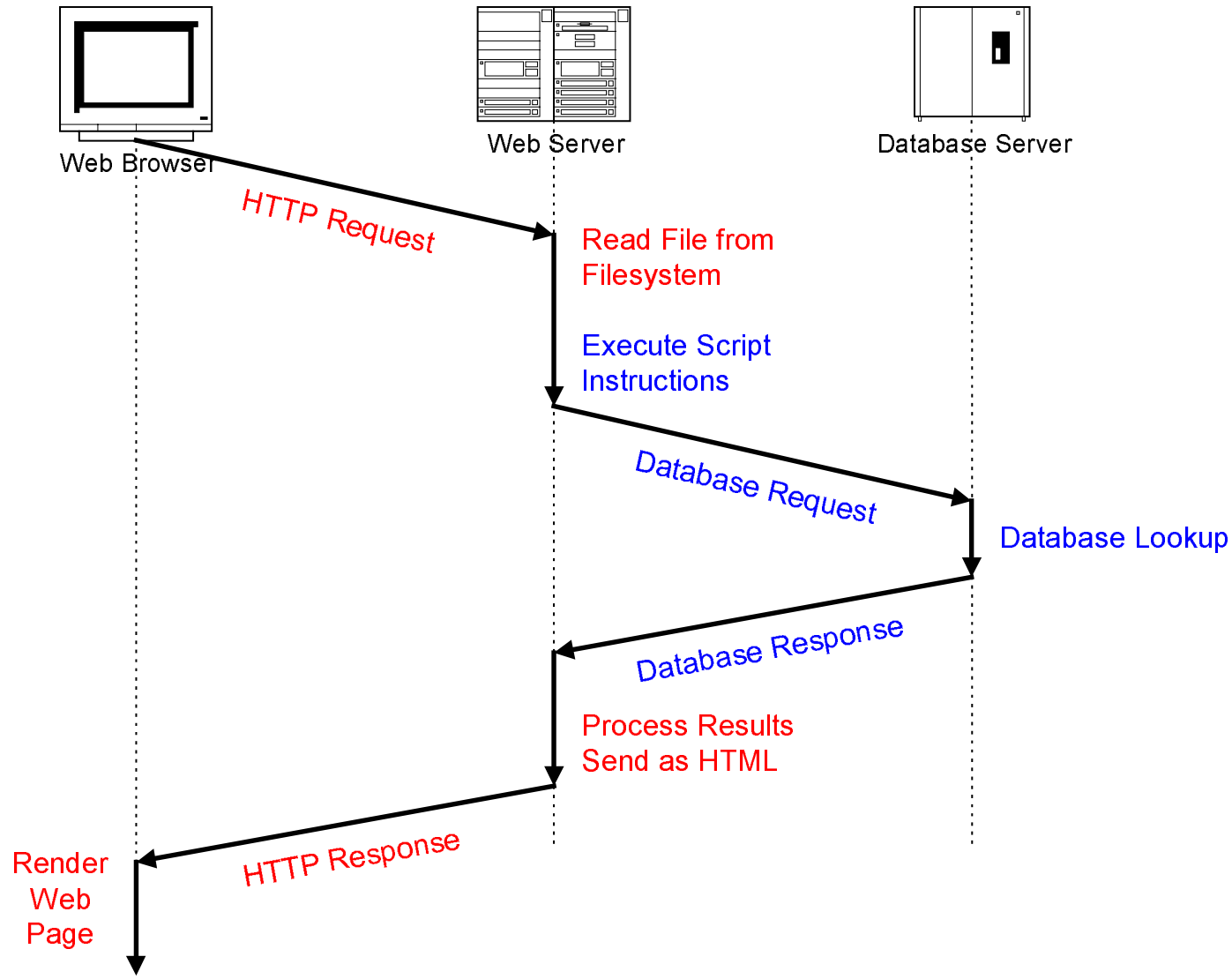
# Static Web Content Illustrated

1. Web browser makes HTTP request
2. Web server matches URL against file system
3. Web server finds corresponding file
4. Web server reads file off disk
5. Web server sends file contents to browser in HTTP response
6. Web browser renders HTML graphically

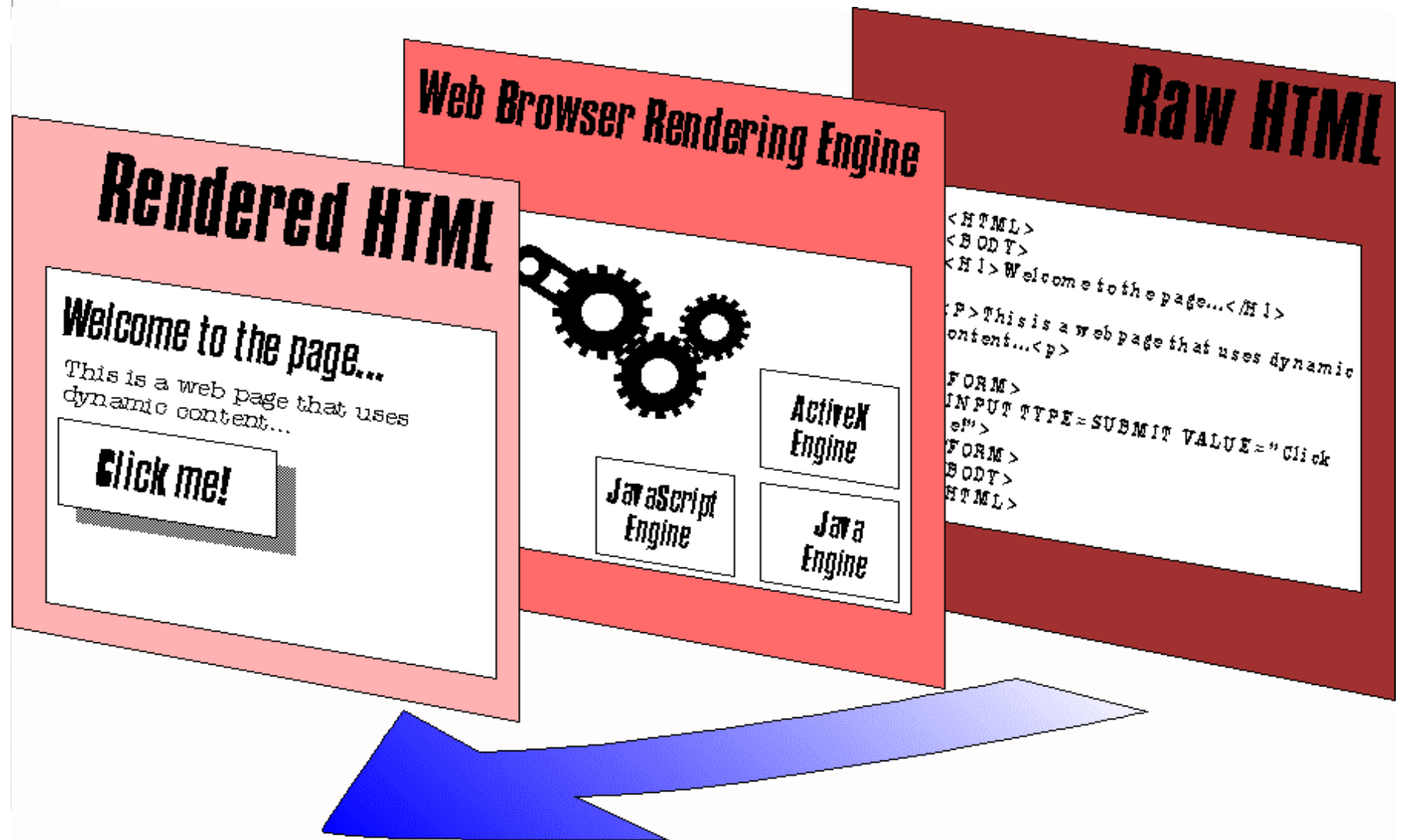


Client-side dynamic content is handled the same way

# Server-side Dynamic Example



# The Rendering Process



# Tour of Technology

- JavaScript
- Java
- Active X
- ASP
- SSI
- CGI
- PHP

---

**We will  
answer:**

1. What is it good for?
2. What are its drawbacks?
3. How hard is it to develop?

# JavaScript: What is it?

```
<script language="JavaScript">
  ➤ Scripting language
  ➤ Interpretted by browsers
  ➤ Can do useful housekeeping and interactivity tasks
  ➤ Not standardized
  ➤ Good example: http://www.cdw.com/
  ➤ Bad example: http://www.belk.com/
</script>
<!-- Fix Netscape CSS bug -->
<script language="JavaScript">
```

# JavaScript

```
<script language="JavaScript">
```

## Good

- Adds **spice**
- Readily **available**
- **Improves usability** of web pages
- **Many examples** and cookbooks available

## Bad

- Makes pages **larger**
- Not universal
- **Can't be searched**, indexed, translated, printed
- **Limited**. Can't do advanced things like database connections or file access

```
</script>
```

```
<!-- Fix Netscape CSS bug -->
```

```
<script language="JavaScript">
```

# Java: What is it?

- Programming language not necessarily web-related
- Interpreted by browsers
- Powerful and extendable
- Standards are shaky

- Good example: <http://fundevaluator.fidelity.com/>

```
* @exception IOException if an I/O error occurs.
* @see      java.io.InputStream#read()
*/
public int read(byte b[], int off, int len) throws IOException {
    if (b == null) {
        throw new NullPointerException();
    } else if ((off < 0) || (off > b.length) || (len < 0) ||
               ((off + len) > b.length) || ((off + len) < 0)) {
        throw new IndexOutOfBoundsException();
    } else if (len == 0) {
        return 0;
    }
}
```

# Java


## Good

- **Powerful**
- Flexible, extensible
- **Offloads computation**
- Can perform **complex tasks**
- **Robust security**
- **Good development environments**

## Bad

- Users often download **large code**
- Not universal
- **Can't be searched, indexed, translated, printed**
- **Standards fluctuate**
- Usually **slow**

# ActiveX

- Client-side, Windows, IE only (not even Mac)
  - Provides complete program functionality
  - Full access to client system
- 
- The logo for ActiveX, featuring a cluster of purple and blue 3D rectangular blocks on the left, and the word "ActiveX" in a stylized, italicized red font on the right, all set against a light red background.
- Can be developed in several languages (Basic, C, C++)
  - Rotten security model

# ActiveX



## Good

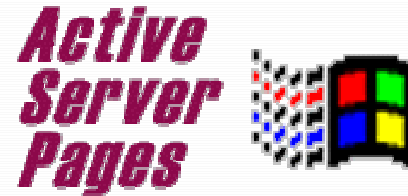
- **Easy** to develop (if you already work in Windows languages, like VB and VC++)
- Easy to **integrate** with Windows products
- **Full access** can solve some otherwise difficult problems

## Bad

- **Windows IE only**
- **Large downloads** like Java
- **Security model** encourages costly applet signing or lax browsing practices

# Active Server Pages (ASP)

- Windows only, server-side dynamic content
- Connects seamlessly to Windows products
- Lots of modules available
- Includes database connectivity
- Rapid development environment



# ASP



## Good

- **Easy** to develop
- Easy to **integrate** with Windows products
- **Development environment** is nice
- Many simple apps are easily developed

## Bad

- **Windows only**, not portable
- **Costs \$\$\$** at many levels
- **Computationally intense** at server, requires beefier hardware than other similar solutions

# Server Side Includes

- Processes simple “comments” to execute commands

- Includes some control flow (if, else, variables)
- Enables invocation of programs or decisions based on a few trivial criteria

```
<!--#include virtual="footer.html" -->
```



- Apache-only module
- Limited utility

# Server Side Includes



## Good

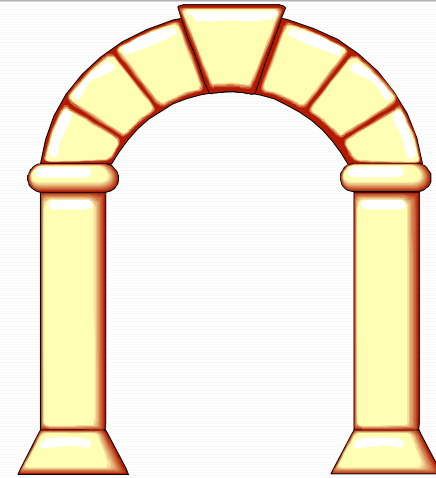
- **Quick** and **simple**
- Building block for more sophisticated content
- Enables modular site design
- Does not defeat caching

## Bad

- **Not sophisticated**
- Crude flow control
- Security issues under unix and windows
- No development environment
- Poor documentation

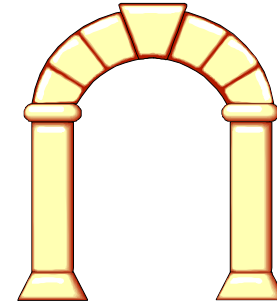
# Common Gateway Interface

- Allows execution of arbitrary programs
- Any language can be used with CGI
- Not platform-specific
- Source code is protected



- Program has a URL like any web page
- Must return known content type (e.g. HTML, GIF, text, etc)

# CGI



## Good

- **Maximally flexible**
- Allows developer to use their favorite environment
- Easily portable
- Can protect secrets or access proprietary data

## Bad

- **Debugging is hard**
- Executes on web server (**compute load**)
- **Security issues** (readily solved)

# PHP

- Powerful scripting language
- Like C, Perl, shell scripts
- Lots of modules standard
  - Oracle
  - MySQL
  - MS SQL
  - Informix
  - Etc...



- Integrates into web server
- Can be modular and efficient

# PHP (personal home page)

## Good

- **Powerful, fast, small**
- Source is hidden
- Includes support for tons of **databases**
- **Easy** to incorporate into HTML
- Similar to **familiar** languages

## Bad

- Security issues
- Different syntax
- No development environment
- Poor documentation

