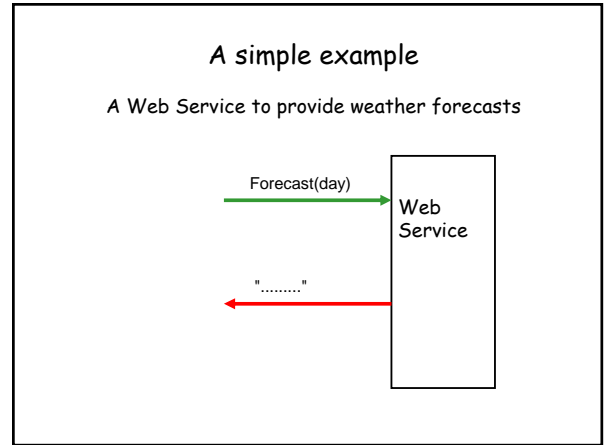
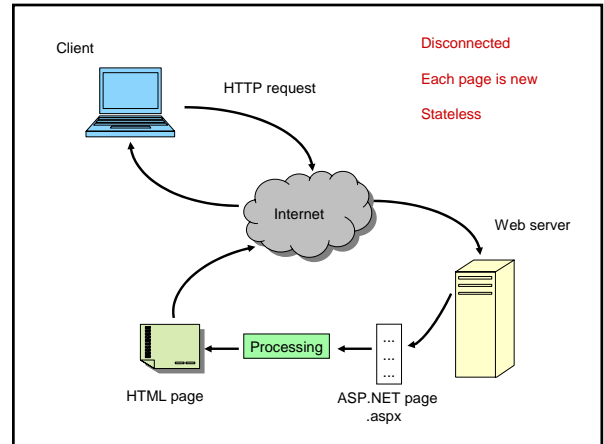


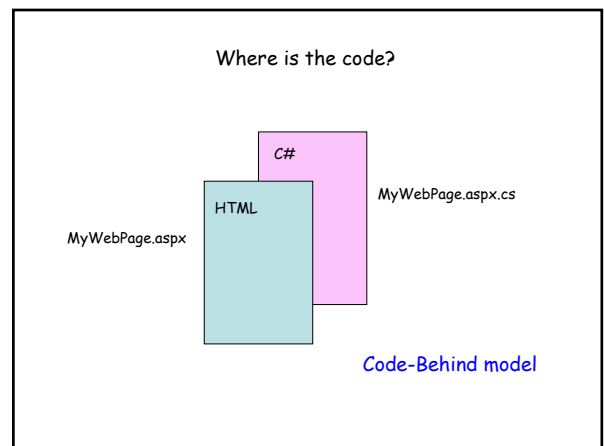
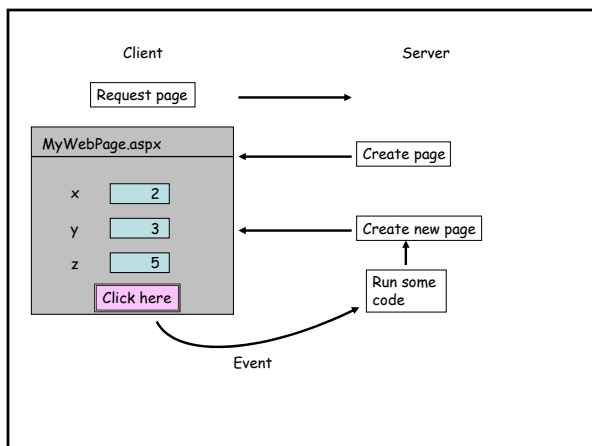
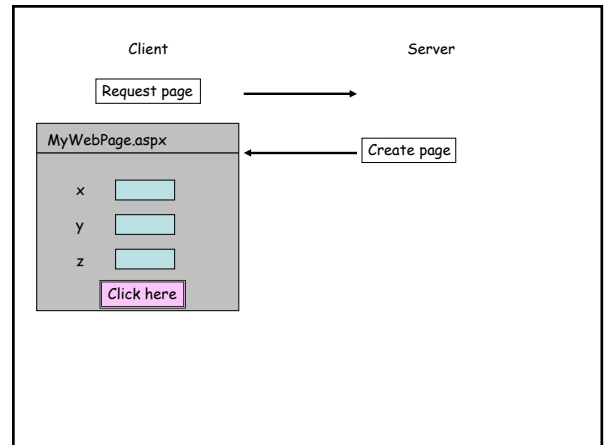
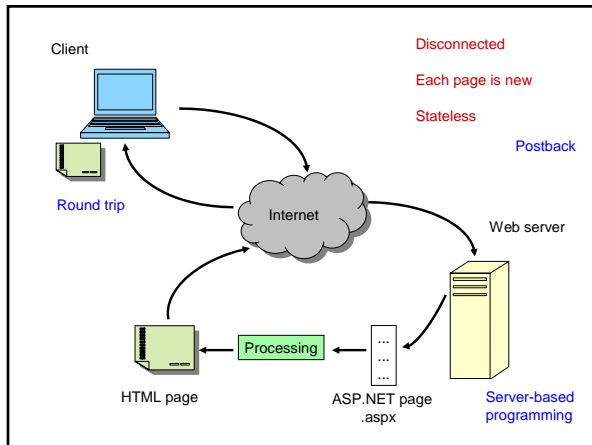
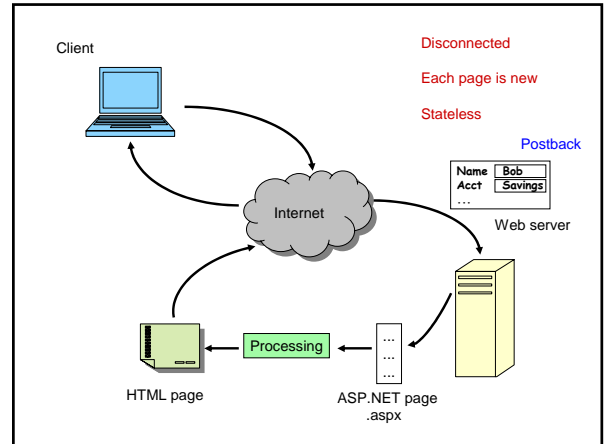
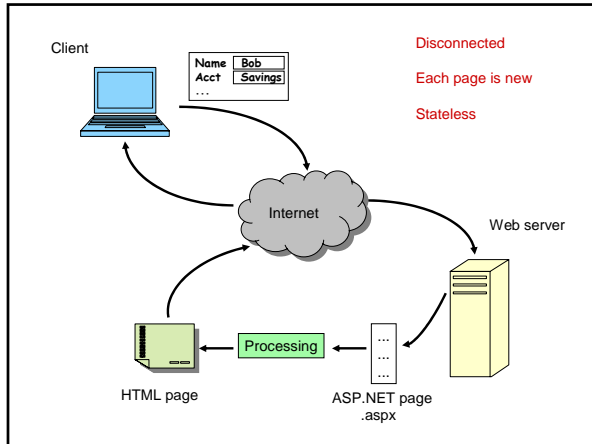
Comparison to the Server/Browser Model

Web page	Web Service
Has a UI	No UI
Interacts with users	Interacts with applications
Works with web browser clients	Works with any type of client



ASP.NET Web Pages





Advantages of Code-Behind

- Clean separation of HTML and code
- Code in any Visual Studio language (C#, VB, C++, J#)
- Easier reuse
- Simpler maintenance
- Deployment without source code

ASP.NET File Types

.aspx	Web forms
.ascx	Web forms user controls
.asmx	Web services
.vb	Visual Basic
.cs	C#
Global.asax	Defines application- and session-variables
Web.config	Configuration settings

Page-Level Events

There are some events that are fired by the page itself.

Init: fired when the page is first instantiated

Load: fired after the page's controls are initialized, but before the page is rendered.

Perfect place for run-time initialization.

Visual Studio generates a stub for this event.

Maintaining State

Input Controls

TextBox, ListBox, CheckBox, etc.

Automatically remember their state via "postback data" sent in the HTTP header

Display Controls

Label, DataGrid, etc. (i.e. controls not set by user)

Automatically remember their state via "viewstate", a hidden control. State is automatically serialized into and out of the viewstate. This behavior can be turned off if desired.

Maintaining State

Cookies

Small chunks of text that are sent back and forth between the browser and the server.

Cookies have names and values.

Session cookies are valid only until the browser closes.

Persistent cookies have a specified lifetime independent of the browser's lifetime.

The page's Request and Response objects have cookie collections.

Saving State on the Server

Session

Application

Session State

- Very important—shopping carts, data scrolling, etc.
- Can be released after a timeout period
- Can survive IIS or worker process restarts. State does not have to be not stored in the worker process.
- Scalable—works with web farms and web gardens
- Must work with browsers that reject cookies
- Good throughput required

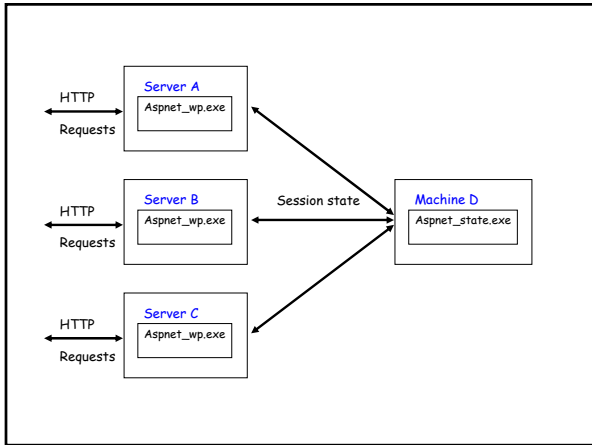
Session State

In-process mode

- State is stored in the worker process and lost if the process is recycled
- The advantage is speed

Out-of-process mode

- State server mode uses a Windows Service: `Aspnet_state.exe`. State is memory-based.



Session State

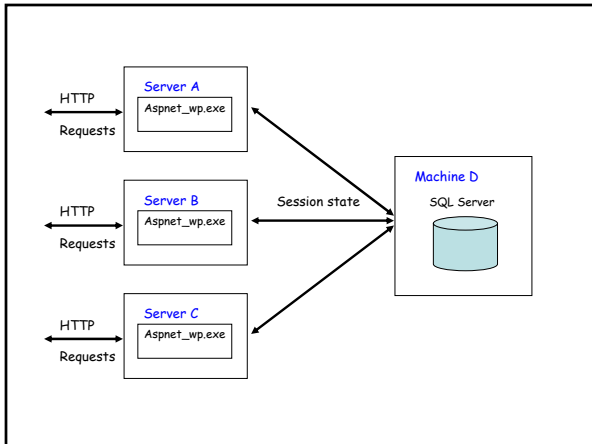
In-process mode

- State is stored in the worker process and lost if the process is recycled
- The advantage is speed

Out-of-process mode

- State server mode uses a Windows Service: `Aspnet_state.exe`. State is memory-based.
- SQL Server mode persists the state information to a SQL server. Servers could be clustered to replicate data.

The choice is made in configuration files



Demo

The Science Fiction Horror Movie Pocket Computer

by
Gahan Wilson

National Lampoon, November 1971, Vol. 1, No. 20

