

ASP.NET programming

Basic concepts

Tuomo Ketomäki

tuomo.ketomaki@iki.fi

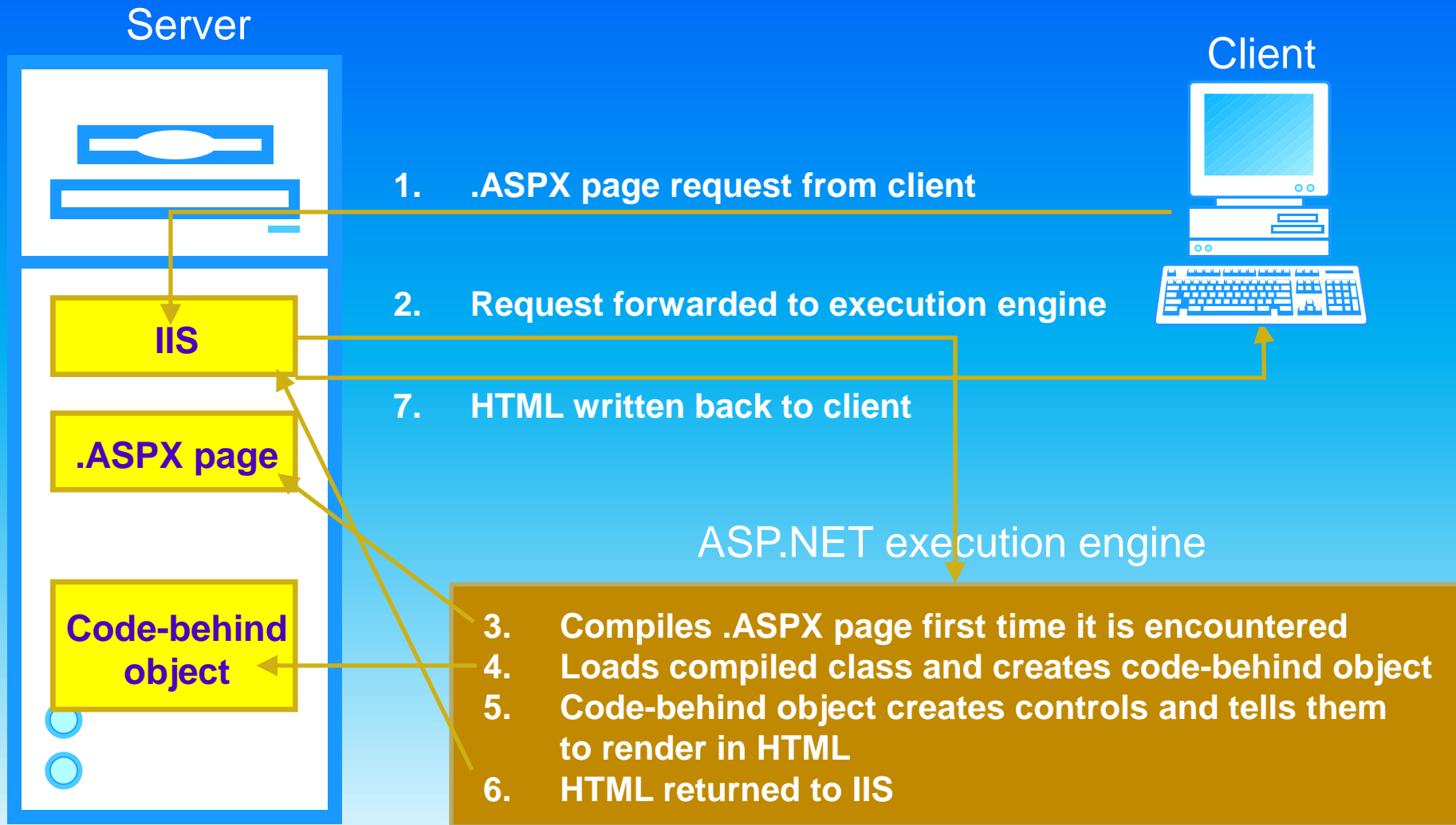
Optimal audience profile

- Understanding of the basics of OOP
- Basic knowledge of implementation of GUI in Web or Windows environment
- Basic knowledge of HTML and server side programming using any of the existing technologies (ASP, PHP, Java etc.)

DEMO

"Hello World"

ASP.NET application model



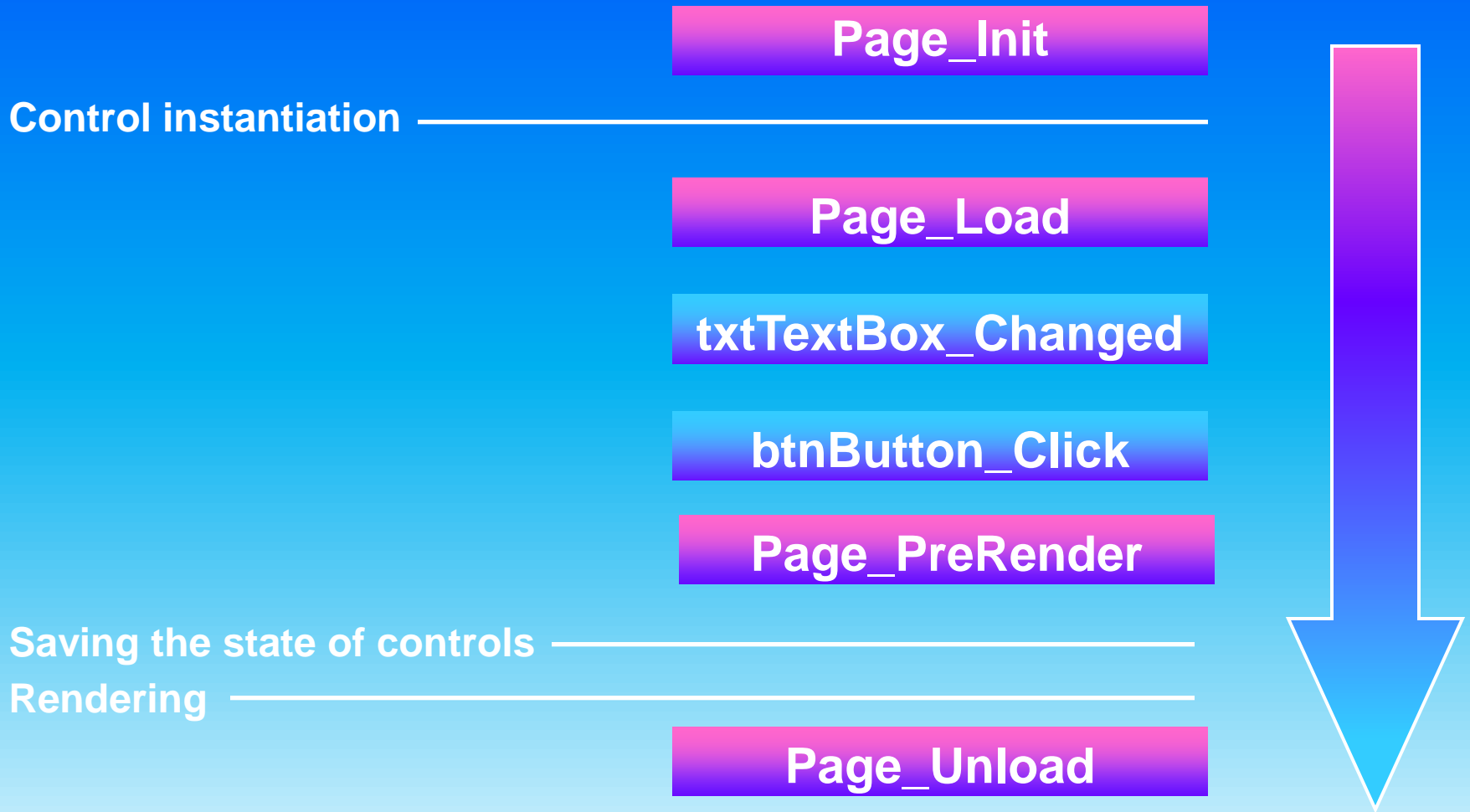
ASP.NET in a nutshell

- ASP.NET syntax defines
 - ASP.NET page controls and lay-out
 - Application functionality
 - Application configuration
- Some concepts of ASP.NET
 - Layout and business logic can be separated
 - Code behind technique
 - Whole .NET Framework class library can be utilized
 - C#, VB.NET, and other .NET languages can be used

Page class – Mother of all web pages

- All the .aspx pages inherit from Page class
- Includes methods, properties and events like any class
- Page declaration can include
 - Directives
 - `<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>`
 - Web Server Control Syntax
 - `<asp:Button ID="Button1" runat="server" Text="Submit" />`
 - HTML Server Control Syntax
 - `<input id="Text1" type="text" runat="server" />`
 - Code Declaration Blocks
 - ```
<script runat="server">
 void Button1_Click(Object sender, EventArgs e) {
 Label1.Text = "Clicked at " + DateTime.Now.ToString();
 }
</script>
```

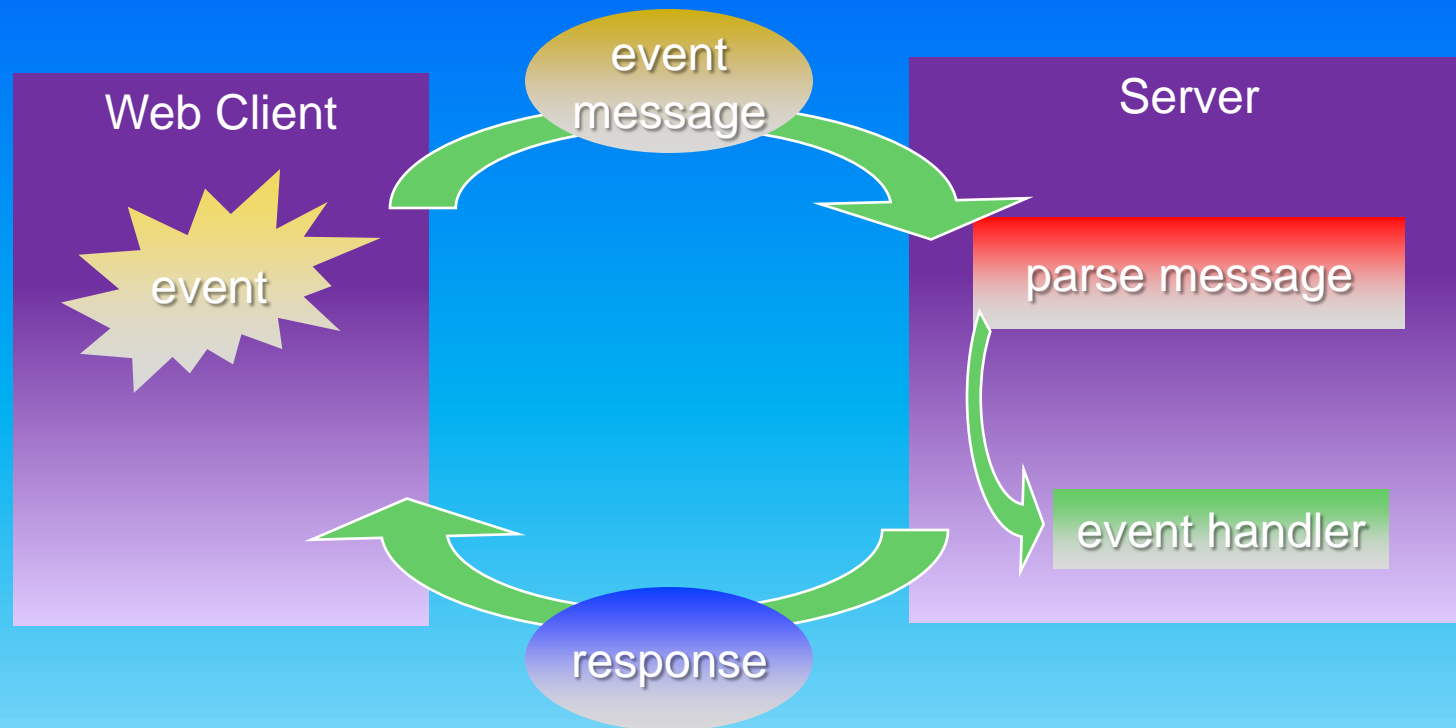
# Page lifespan



# ASP.NET event handling model

Events are raised on client....

and handled on server



- Round trip to server causes a performance penalty
  - Unintentional events (like MouseMove) are not supported
- AutoEventWireup property value is "False" by default, but Web Form template in VS sets it to "True" in Page directive



# DEMO

## Event handling

# ASP.NET server controls

- Programming model of Web Forms is similar to that of Windows Forms as closely as possible
  - Server controls are objects having properties, methods and events
- User controls (.ascx) can be defined
- ASP.NET includes over 70 controls
- Different controls for different purposes
  - Intrinsic controls
  - Data-bound controls
  - Validation controls
  - Rich controls

AccessDataSource, AdRotator, AppearanceEditorPart, BehaviorEditorPart, BulletedList, Button, Calendar, CatalogZone, ChangePassword, CheckBox, CheckBoxList, CompareValidator, ConnectionsZone, Content, ContentPlaceHolder, CreateUserWizard, CrystalReportPartsViewer, CrystalReportSource, CrystalReportViewer, CustomValidator, DataGrid, DataList, DeclarativeCatalogPart, DetailsView, DropDownList, EditorZone, FileUpload, FormView, GridView, HiddenField, HyperLink, Image, ImageButton, ImageMap, ImportCatalogPart, Label, LayoutEditorPart, LinkButton, ListBox, Literal, Localize, Login, LoginName, LoginStatus, LoginView, Menu, MultiView, ObjectDataSource, PageCatalogPart, Panel, PasswordRecovery, Placeholder, PropertyGridEditorPart, ProxyWebPartManager, RadioButton, RadioButtonList, RangeValidator, RegularExpressionValidator, Repeater, RequiredFieldValidator, SiteMapDataSource, SiteMapPath, SqlDataSource, Substitution, TableWeb, TableCell, TableRow, TextBox, TreeView, ValidationSummary, View, WebPartManager, WebPartZone, Wizard, Xml, XmlDataSource

# TextBox control properties

- Element generated depending on TextMode property

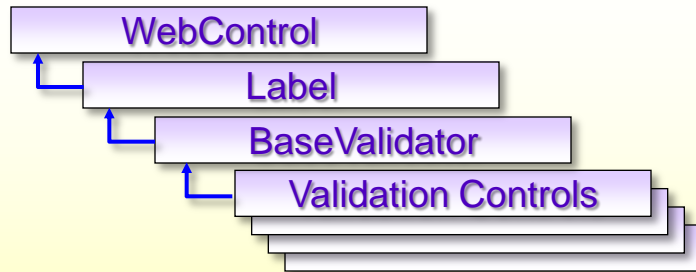
| Setting             | HTML element            | Remarks                      |
|---------------------|-------------------------|------------------------------|
| SingleLine(default) | <input type "text">     |                              |
| Multiline           | <textarea>              | Vertical scrollbar if needed |
| Password            | <input type "password"> | No encryption!!              |

- **AutoPostBack**
  - If false (default), TextChanged event does not cause posting to server
  - If true, TextChanged event causes posting when focus is moved from TextBox



# Validation controls

- Inheritance hierarchy



- RequiredFieldValidator
- CompareValidator
- RangeValidator
- RegularExpressionValidator
- CustomValidator
- ValidationSummary

- Validate associated input control
  - ControlToValidate property
  - Can apply multiple validation controls to a single control
- Support client-side validation (default)
  - Avoiding posting to server for validation
  - Delivering JavaScript to the client (no coding necessary!)
- Error message will be displayed if validation fails

# Required field validator

- Validates the user input against initial value of the input control
  - Validation fails if input is same as initial value

```
<asp:RequiredFieldValidator
 ID="vldReqName"
 runat="server"
 ErrorMessage="You must enter your name"
 InitialValue="[You must replace this initial value]"
 ControlToValidate="txtName">
</asp:RequiredFieldValidator>
```

Please enter your name  
[You must replace this initial value]  
You must enter your name

# DEMO

## Input validation

# Links

- <http://www.asp.net>
- <http://ajax.asp.net>
- <http://weblogs.asp.net/scottgu>
- <http://www.asp.net/learn>  
(Learning videos)