

VSLive!
SAN FRANCISCO



Data-Driven ASP.NET Ajax

Jeffrey McManus
CEO, Platform Associates
Developer, Approver.com

Pre-requisites for this presentation:

- 1) Knowledge of ASP.NET
- 2) Knowledge of ADO.NET

Level: Intermediate



About Jeffrey McManus

- Developer of document-sharing site Approver.com (which uses ASP.NET Ajax and MySQL)
- CEO of Platform Associates, helping new and established businesses bring platform products to developer communities (platformassociates.com)
- Author/co-author of six books on Microsoft development technologies including *Database Access with Visual Basic* and *C# Developer's Guide to ASP.NET, XML, and ADO.NET*
- Founded and led platform evangelism team at eBay
- Founded and led Yahoo! Developer Network
 - Productized and open-sourced the Y! UI Ajax Library, in addition to many other developer products
- Gave first VBITS/VSLive session back in 1996

About This Talk

- Information about Ajax in general, and Microsoft's implementation of it
- Creating data-driven Ajax user experiences
- Lots of practical examples and demos

What is Data-Driven Ajax?

- Architecting and developing a data-driven Web application around the notion of asynchronous postbacks
- **Asynchronous postbacks** enable an out-of-band communication channel
 - Combined with dynamic HTML, this enables you to update part of the page without a complete page postback
 - Preserves page state more efficiently than ASP.NET viewstate
 - Can be (or appear to be) more efficient than a full page postback
 - Enables the UI designer to prevent user interface modality
- Many Ajax libraries (including ASP.NET Ajax) also provide UI widget libraries that combine DHTML with Ajax postbacks in interesting ways
- Using dynamic data (either relational or service-based) in an Ajax application is the focus of this talk

Availability

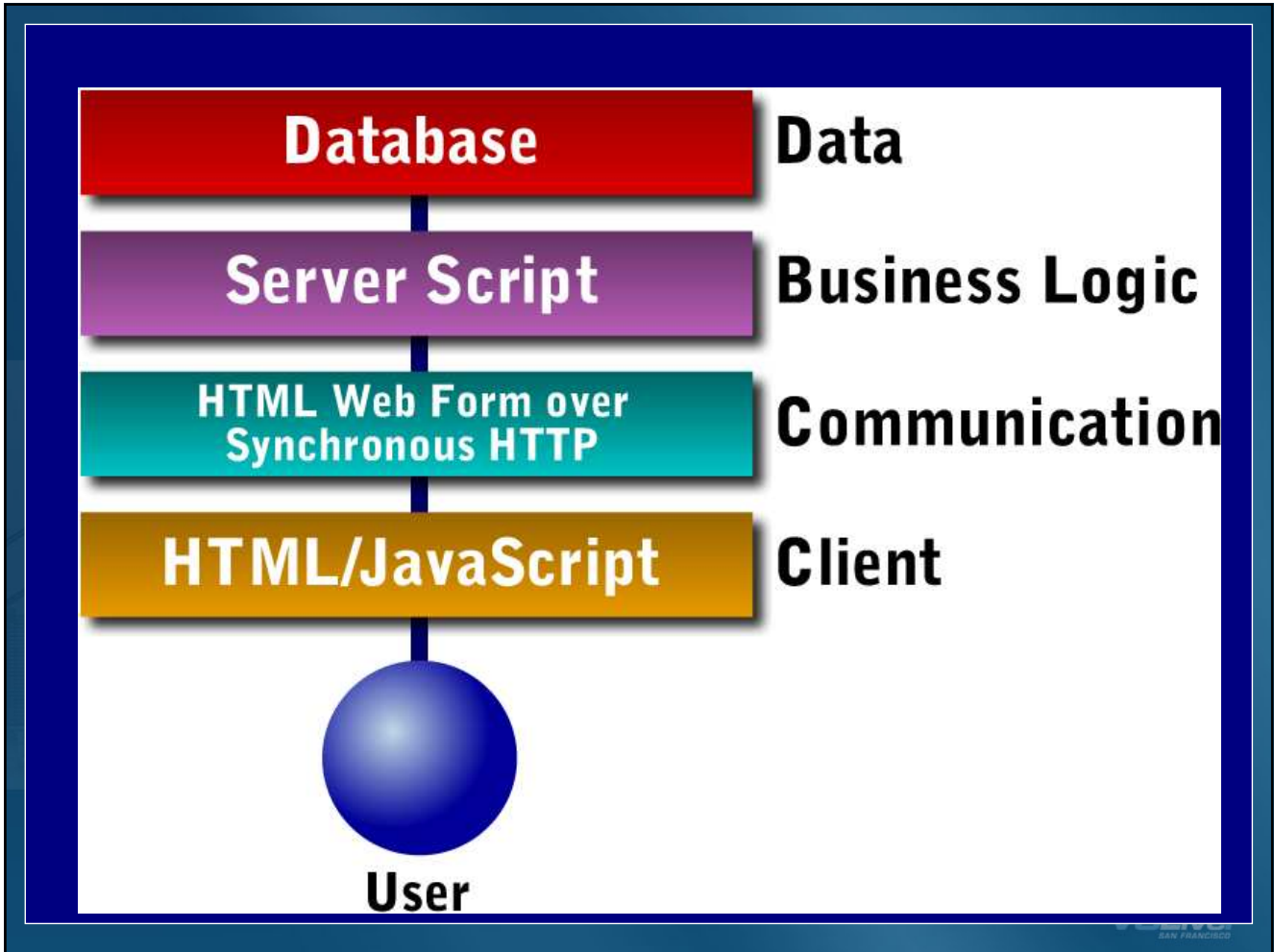
- ASP.NET Ajax is built into ASP.NET 3.5
- ASP.NET Ajax an optional free install for ASP.NET 2.0
 - My code examples were built using 2.0 but should run on 3.5

Guidelines for Architecting Ajax into a Data-Driven Web Application

- Avoid using Ajax just for the sake of using it
- Follow the “principle of least astonishment”
- Add Ajax features to your application to satisfy specific user needs:
 - Provide information to the user quickly, economically, and in context
 - Draw the user’s attention to important events in the lifecycle of the application
 - Provide visual feedback during long-running events

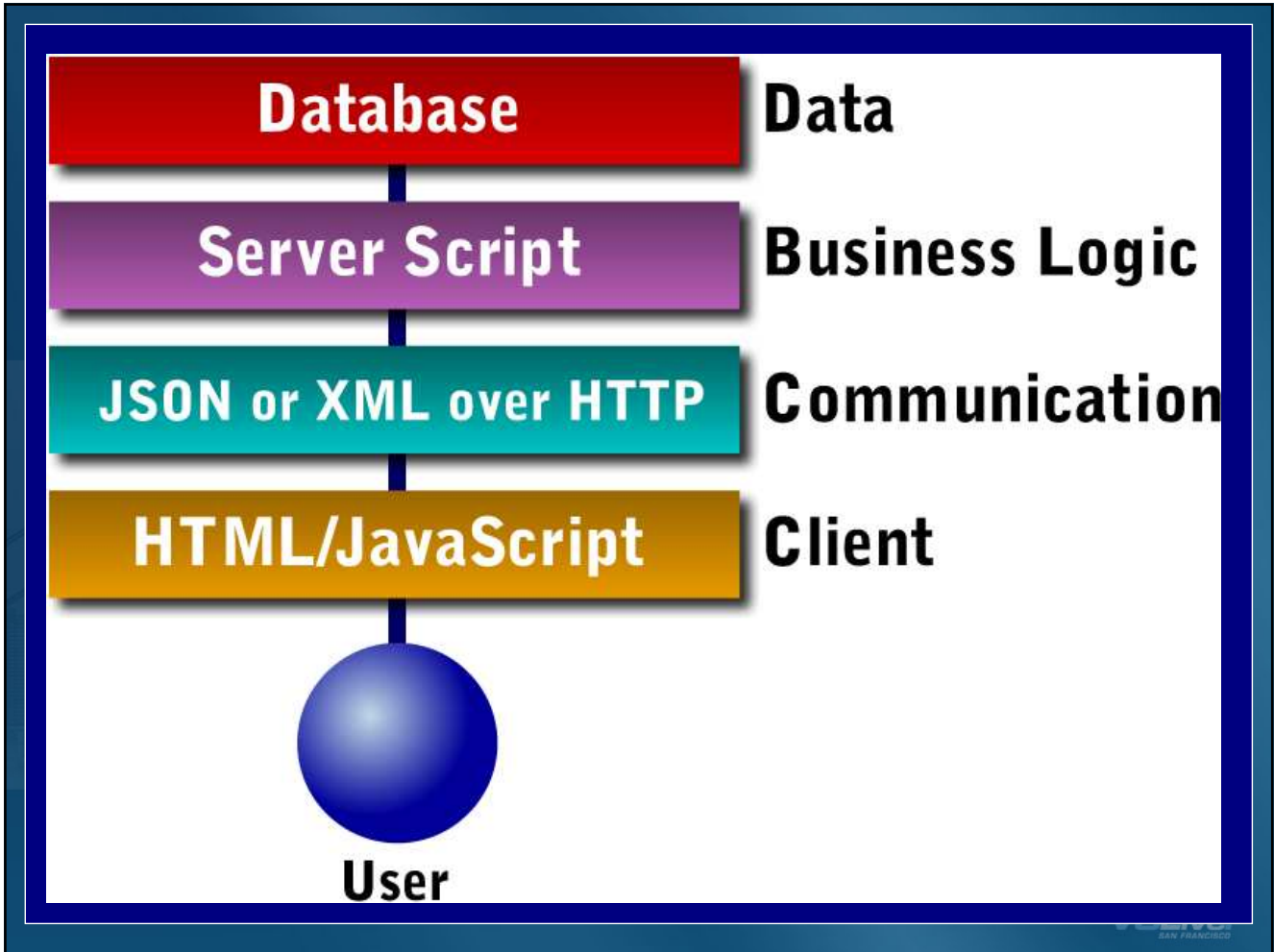
Basic Postback Features of ASP.NET

- Conventional (non-Ajax) ASP.NET gives you a few features of Ajax web apps
 - Preservation of page state when form is posted
 - This comes at the cost of a full page postback (including viewstate) which is almost always more costly than an Ajax post
- Client-side validation (via DHTML/Javascript)
- If these are the only Ajax-esque features your page requires, you may not need to do anything special to your page



Ajax Postback

- Uses out-of-band HTTP channel to post back and receive results from the server side
- Objective is to **post back** and **process** as little data as possible to get the job done
 - Instead of posting back the whole page/form, you only post back the data that is relevant to the user's action
 - What format is this data in? How is it marshaled back to the server?



About JSON

- JavaScript object notation
 - <http://json.org>
- A very efficient way to handle data in a browser client
 - Standards based (comprised of Javascript key-value pairs)
 - Native data enables fast use in Javascript without parsing or XML's metadata overhead
 - In use by big web service providers today (Yahoo!, del.icio.us, Flickr, Google, many more)
 - **Used internally by ASP.NET Ajax to marshal data from client to server with an UpdatePanel**
- Maddeningly, ASP.NET Ajax does not expose JSON bindings to the application developer!
 - If you want to use ASP.NET Ajax to consume a JSON web service, it requires hackery

.NET JSON Libraries

- Enable you to serialize/deserialize objects as JSON
- JSON.NET
 - Codeplex: <http://www.codeplex.com/Json>
- Jayrock
 - <http://jayrock.berlios.de/>
- Both are free/open-source

System.Data.SQLite

- Embedded relational database
- Supports multiple OS platforms
- Free, open source and public domain
- In .NET, data engine and ADO.NET Provider are statically bound in a 220K DLL
- This makes it easy for me to give all my source code (and data) to you
 - Zero configuration demonstration

ASP.NET Ajax Demands SOAP

- This works OK, but has some unfortunate limitations:
 - UpdatePanel marshals data using highly efficient JSON while your application can be stuck using much less efficient SOAP
 - You have to write a SOAP proxy for just about any data you'd ever want to access (including remote web services as well as relational data stored on your server)

Building Data-Driven Ajax UIs: Practical Examples

Data Browsing in GridView

- Demo: GridView browsing (no Ajax):
employees.aspx
- Demo: GridView browsing (with Ajax):
employees-ajax.aspx
- We use LIMIT and OFFSET in both cases to avoid marshalling all 1,000,000 records over the wire
- Ajax example still appears more responsive (even on localhost – the difference is even more apparent on the network)

Demo: Basic Data Entry Postback

- The “hello world” example
- Non-Ajax: basic.aspx
- Ajax: basic-ajax.aspx
 - This includes the trick that lets you set focus to a control on the page

Employee SOAP Service

- Lots of ASP.NET Ajax functionality requires SOAP for callbacks
- In our demo, the Employee service has GetNote, SaveNote, and Search methods
- You can also do these as inline static WebMethods
- Employee.asmx

Demo: Autocomplete

- Objective: Display a large list of choices to the user
- Requires ajaxToolkit:AutoCompleteExtender with callback to a SOAP service that has a specific signature
- Autocomplete.aspx

Demo: Pop-Up Form

- Hide sensitive or verbose data until it is requested
- Simplify a complex data entry page by hiding optional or seldom-used fields
- `Popup.aspx`
- Same thing with a static web method: `popup-save-static.aspx`

Display Progress Control During Long-Running Processes

- A must for any Ajax postback
 - Typical browser “wait” functionality does not kick in automatically during an Ajax postback (because it’s not a real postback)
 - Even if the request seems to happen instantly on localhost, there is always more latency in production
 - If the postback takes more than a half second or so, users will think something is wrong
 - Fortunately in ASP.NET Ajax it’s really easy to show progress using the UpdateProgress control

UpdateProgress

- Two parts to this: The declarative part and the CSS part
- Demo: progress.aspx

```
<asp:UpdateProgress ID="PostProgress" runat="server" DynamicLayout="false">  
  <ProgressTemplate>  
    <div id="PostProgress">Loading...</div>  
  </ProgressTemplate>  
</asp:UpdateProgress>
```

```
#PostProgress  
{  
  font-family: Arial, Helvetica, sans-serif;  
  background-color: #FFFFCC;  
  color: #666666;  
  top: 0px;  
  right: 0px;  
  position: fixed;  
  padding: 3px;  
  border: 2px solid #666666;  
}
```

Demo: Submit One of Several Forms on the Same Page Without Clicking

- Surprisingly tricky; requires some inline Javascript
- `Submit.aspx`

Alternatives to ASP.NET Ajax

- There are an unbelievable number of Ajax libraries out there
 - I'd heard a count of more than 140 separate projects back in 2006
- The most mature include: Y!UI, Ext JS, Dojo, Prototype, Mochikit
- Coolite provides a .NET wrapper around Ext JS
- There is an interoperability initiative underway (OpenAjax Alliance, openajax.org)
 - Intention is for various vendors' Ajax stacks to not break when used with each other
 - This is still a work in progress
 - Not a lot of resources that actually demonstrate, for example, how to use Ext JS widgets with ASP.NET Ajax

More Information

- ASP.NET Ajax: <http://asp.net/ajax/>
- Coolite (Ext JS for ASP.NET):
<http://www.coolite.com/>