Beyond REST

An approach to creating stable, evolve-able Web applications

Mike Amundsen
@mamund

Preamble

- Mike Amundsen
- Developer, Architect, Presenter
- Hypermedia Junkie

- “I program the Internet.”

- *Designing Hypermedia APIs with Node and HTML5 Fall 2011*

Preamble

- Beyond REST
- Not “Better than” REST
- Not “After” REST
- Just “Beyond” REST

Overview

- The Question

Overview

- The Question
- A Few Observations

Overview

- The Question
- A Few Observations
- One Approach

Overview

- The Question
- A Few Observations
- One Approach
- Some Techniques

The Question

The question is...

The question is...

How can we design

The question is...

How can we design
and implement

The question is...

How can we design
and implement
distributed network solutions
that remain

The question is...

How can we design
and implement
distributed network solutions
that remain
stable

The question is...

How can we design
and implement
distributed network solutions
that remain
stable
and
flexible

The question is...

How can we design
and implement
distributed network solutions
that remain
stable
and
flexible
over time?

The question is...

How can we design
and implement
distributed network solutions
that remain
stable
and
flexible
over time?

The question is...

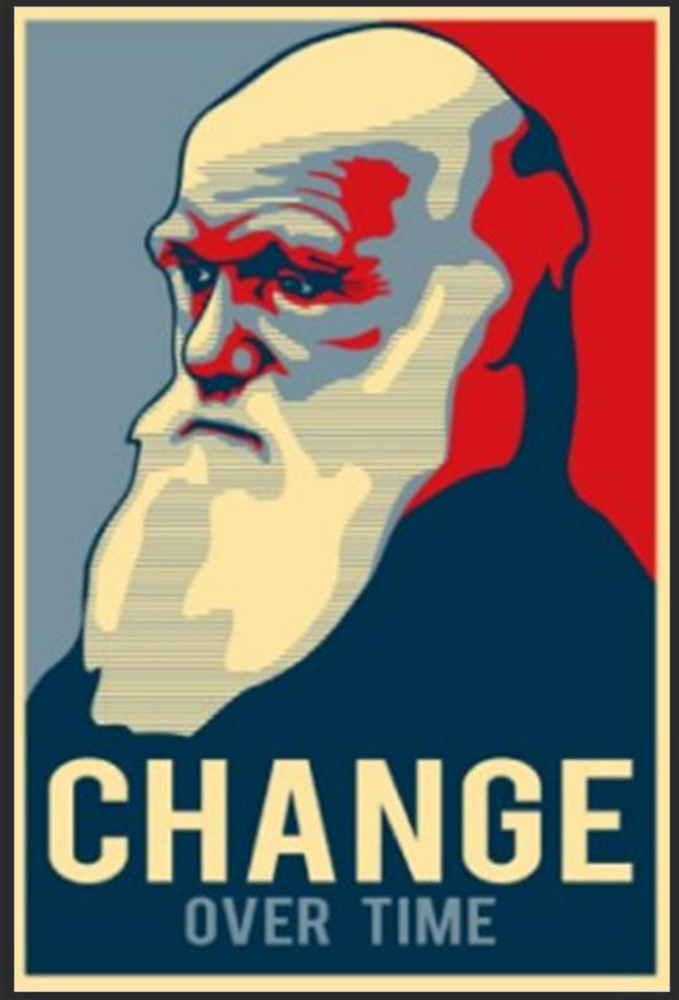
How can we design
and implement
distributed network solutions
that remain
stable
and
flexible
over time?

Evolvable systems.

The question is...

How can we design
and implement
distributed network solutions
that remain
stable
and
flexible
over time?

Evolvable systems.



A Few Observations

Definitions

Definitions

- Stability

Definitions

- Stability
  - "the strength to stand or endure" - Webster

Stability

Stability

Definitions

- Stability
    - "the strength to stand or endure" - Webster

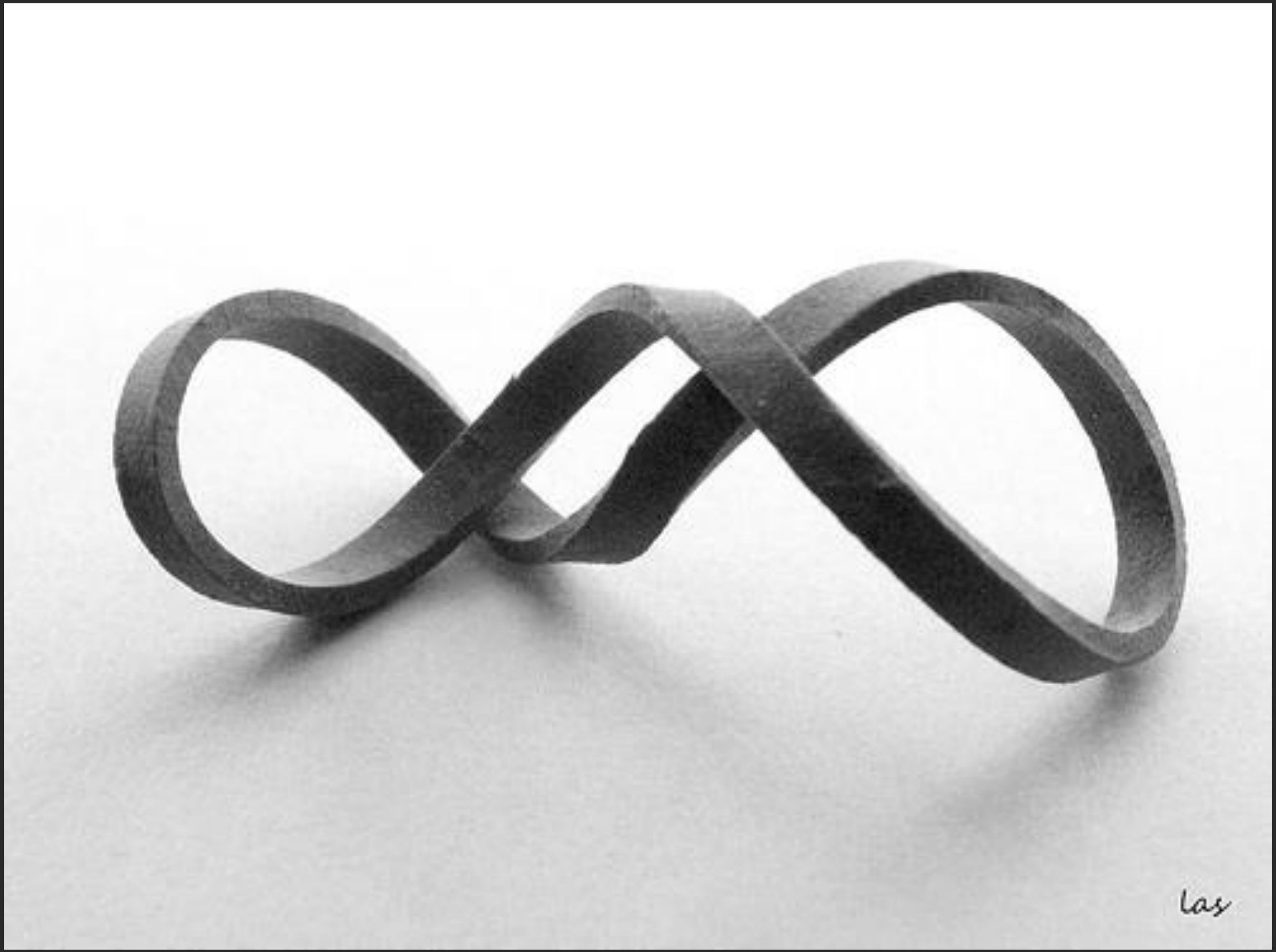
Definitions

- Stability
  - "the strength to stand or endure" - Webster
- Flexibility

Definitions

- Stability
  - "the strength to stand or endure" - Webster
- Flexibility
  - "characterized by a ready capability to adapt to new, different, or changing requirements." - Webster

Flexibility



Flexibility

Definitions

- Stability
  - "the strength to stand or endure" - Webster
- Flexibility
  - "characterized by a ready capability to adapt to new, different, or changing requirements." - Webster

Definitions

- Stability
  - "the strength to stand or endure" - Webster
- Flexibility
  - "characterized by a ready capability to adapt to new, different, or changing requirements." - Webster
- Time

Definitions

- Stability
  - "the strength to stand or endure" - Webster
- Flexibility
  - "characterized by a ready capability to adapt to new, different, or changing requirements." - Webster
- Time
  - "a nonspatial continuum that is measured in terms of events which succeed one another from past through present to future." - Webster

Time

Time

Time

On the scale of years

“Most of REST's constraints are focused on preserving independent evolvability over time, which is only measurable on the scale of years.

Roy Fielding, August 2010

Another way to see it...

Flexible

Stable AND Flexible

Stable AND Flexible Over Time

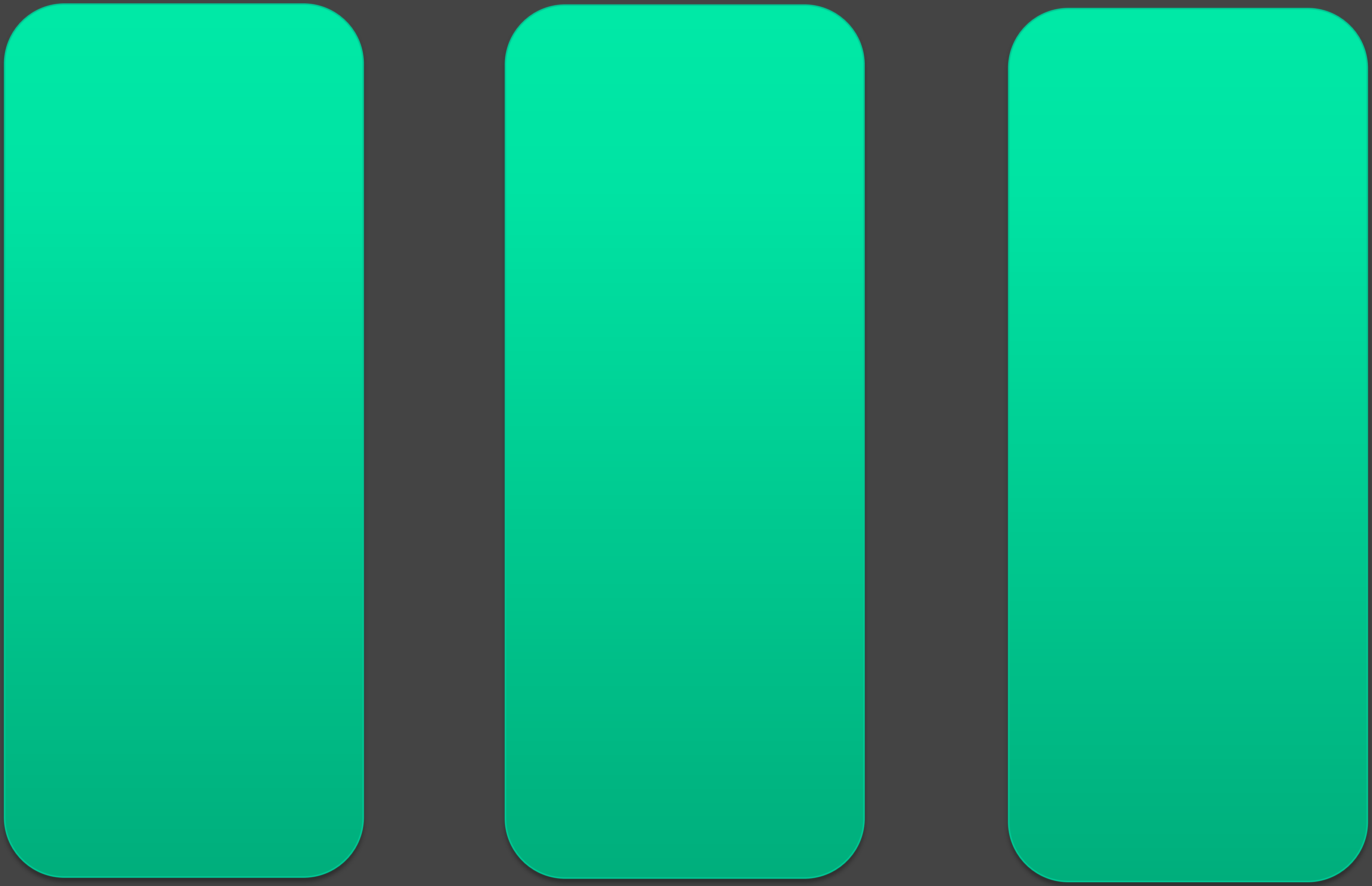
Alive … Stable

“In short, saying these [patterns] are alive is more or less the same as saying they are stable.”

Christopher Alexander, 1979

One Approach

A Model

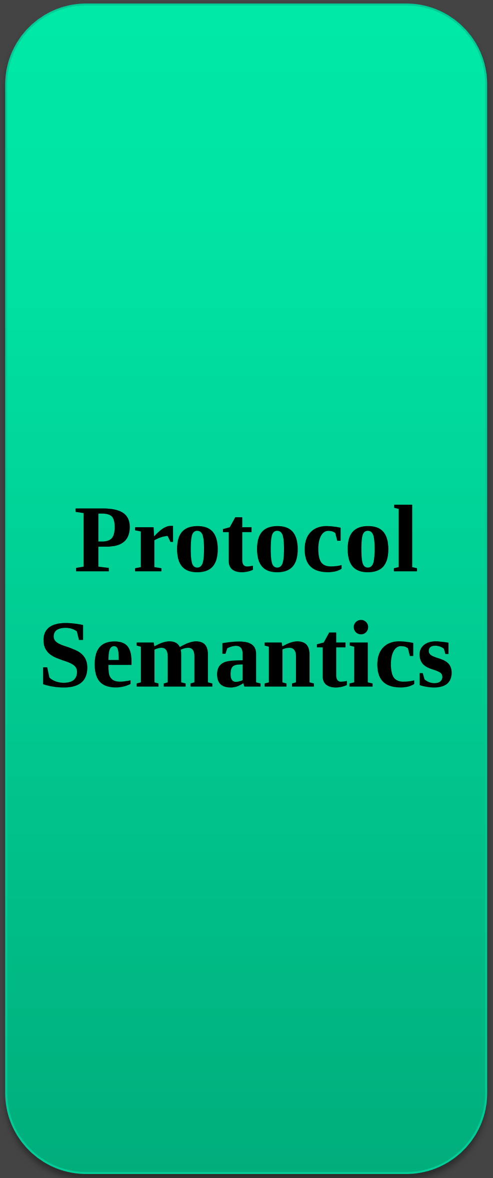


A Model

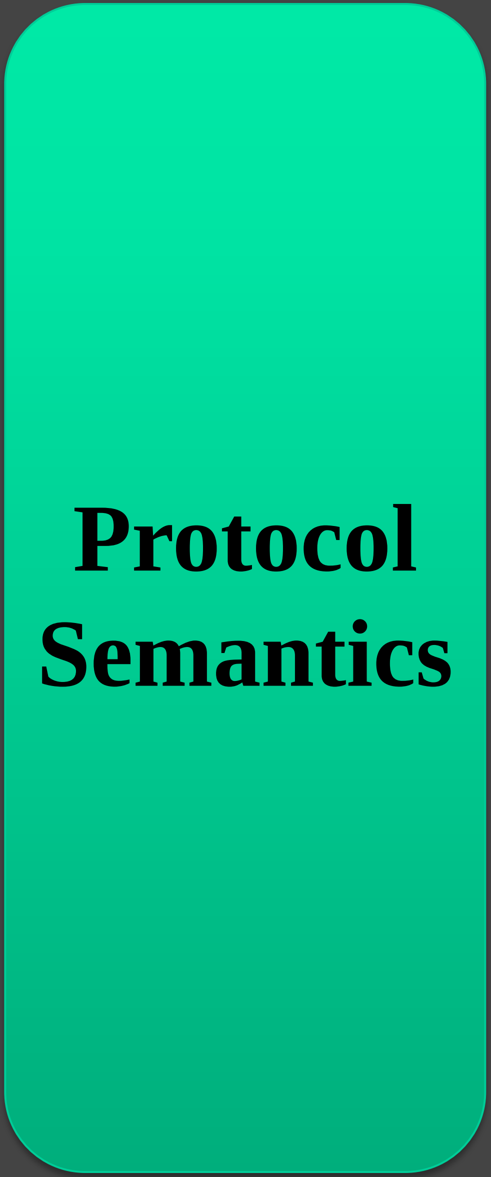
“Design depends largely on constraints”

Charles Eames, 1972

A Model



A Model



- The transfer protocol
- HTTP, FTP, etc.
- Standardized (RFCs, etc.)
- Slowest changing

- Shared by all participants

- Use “as-is”
- The stable foundation

A Model

Protocol Semantics

State Handling

A Model

Protocol Semantics

State Handling

A Model

State Handling

- Identification
- Sharing
- Storage

- Transient
- Unique for each participant

- Create/Manipulate as Needed

- Identify & share via message
  - Media type
  - Headers
- Store locally

A Model

Protocol Semantics

State Handling

A Model

Protocol Semantics

Domain Semantics

State Handling

A Model

Protocol Semantics

Domain Semantics

State Handling

A Model

Domain Semantics

- Application-level semantics
- Discover
- Encapsulate
- Describe

- Shared Understanding
- For acknowledged participants

- Evolvable over time

- Message-Enabled
  - Media type
  - Semantic Profile, etc

A Model

Connector

Protocol Semantics

Data

Domain Semantics

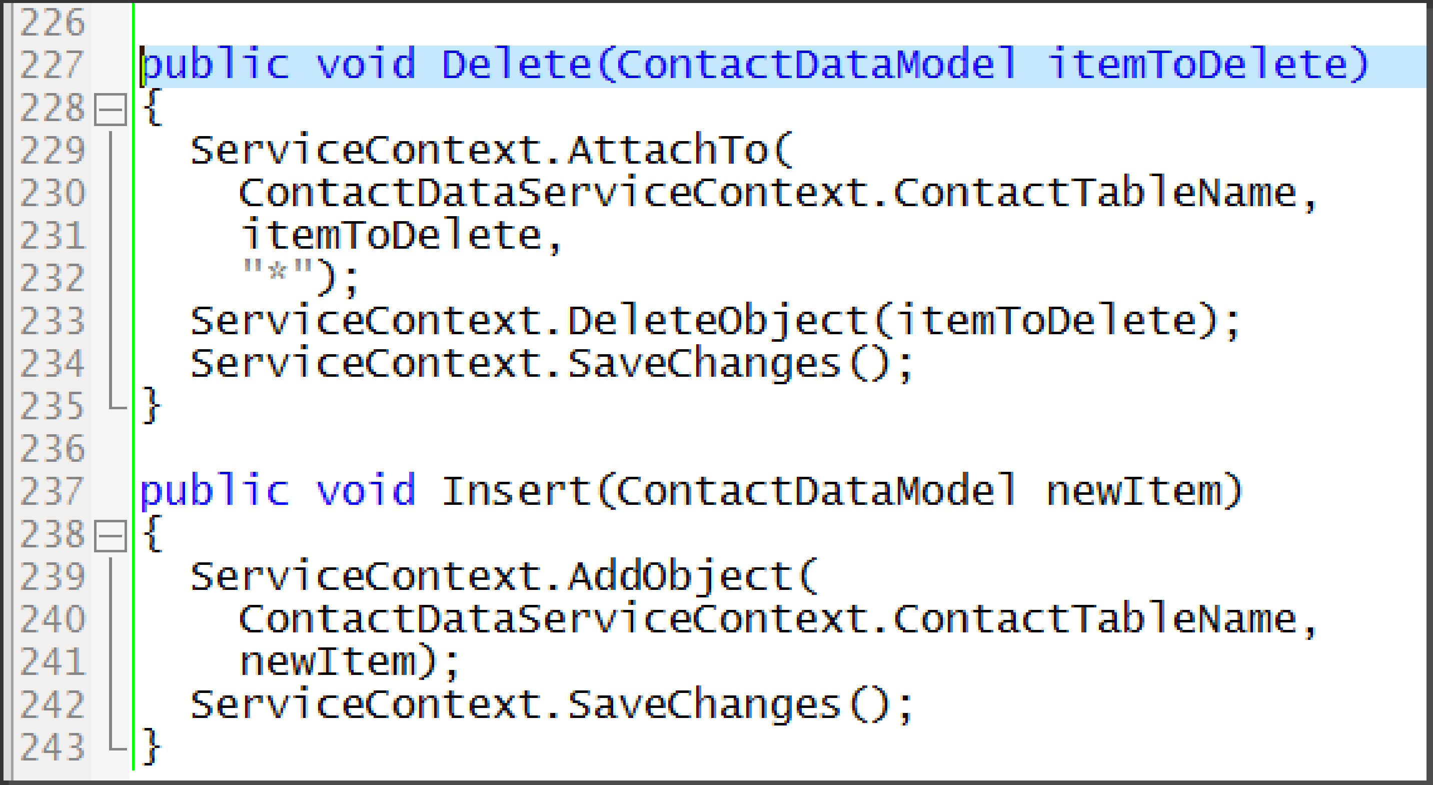
Component

State Handling

Some Techniques

Connector | Protocol Techniques

- Embrace HTTP as the “network interface”
  - Methods
  - Response Codes
  - Headers
  - Media types

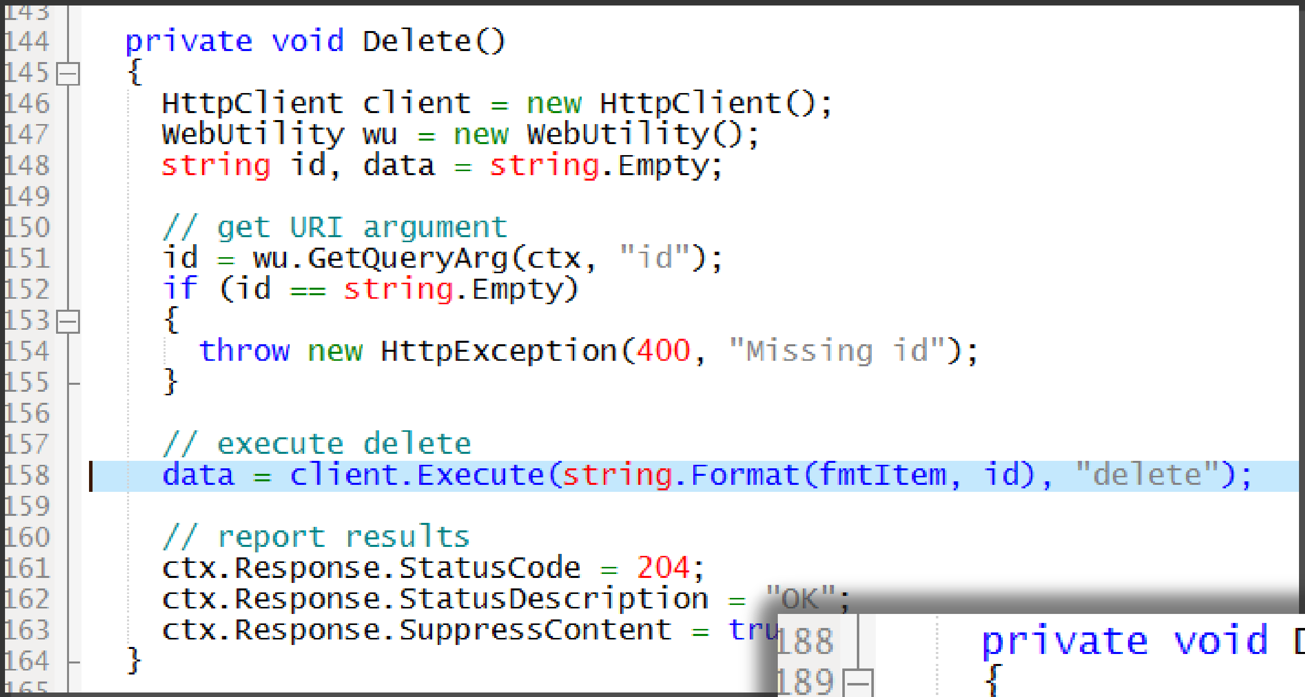


```
public void Delete(ContactDataModel itemToDelete)
{
  ServiceContext.AttachTo(
    ContactDataServiceContext.ContactTableName,
    itemToDelete,
    "*");
  ServiceContext.DeleteObject(itemToDelete);
  ServiceContext.SaveChanges();
}

public void Insert(ContactDataModel newItem)
{
  ServiceContext.AddObject(
    ContactDataServiceContext.ContactTableName,
    newItem);
  ServiceContext.SaveChanges();
}
```

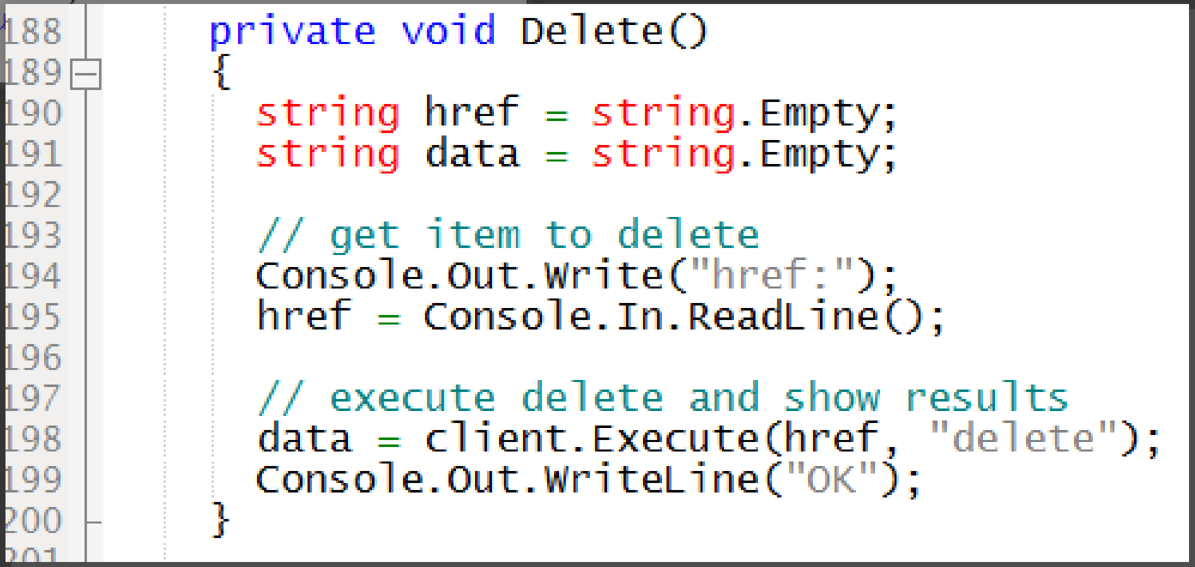
This NOT an HTTP Interface

```
pub                                                  elete)
{
    S
                                                     ame,


    S                                                ;
    S
}

pub                                                  )
{
    S
                                                     ame,

    S
}
```

HTTP Is NOT The Interface



```
private void Delete()
{
  HttpClient client = new HttpClient();
  WebUtility wu = new WebUtility();
  string id, data = string.Empty;

  // get URI argument
  id = wu.GetQueryArg(ctx, "id");
  if (id == string.Empty)
  {
    throw new HttpException(400, "Missing id");
  }

  // execute delete
  data = client.Execute(string.Format(fmtItem, id), "delete");

  // report results
  ctx.Response.StatusCode = 204;
  ctx.Response.StatusDescription = "OK";
  ctx.Response.SuppressContent = tru
}
```

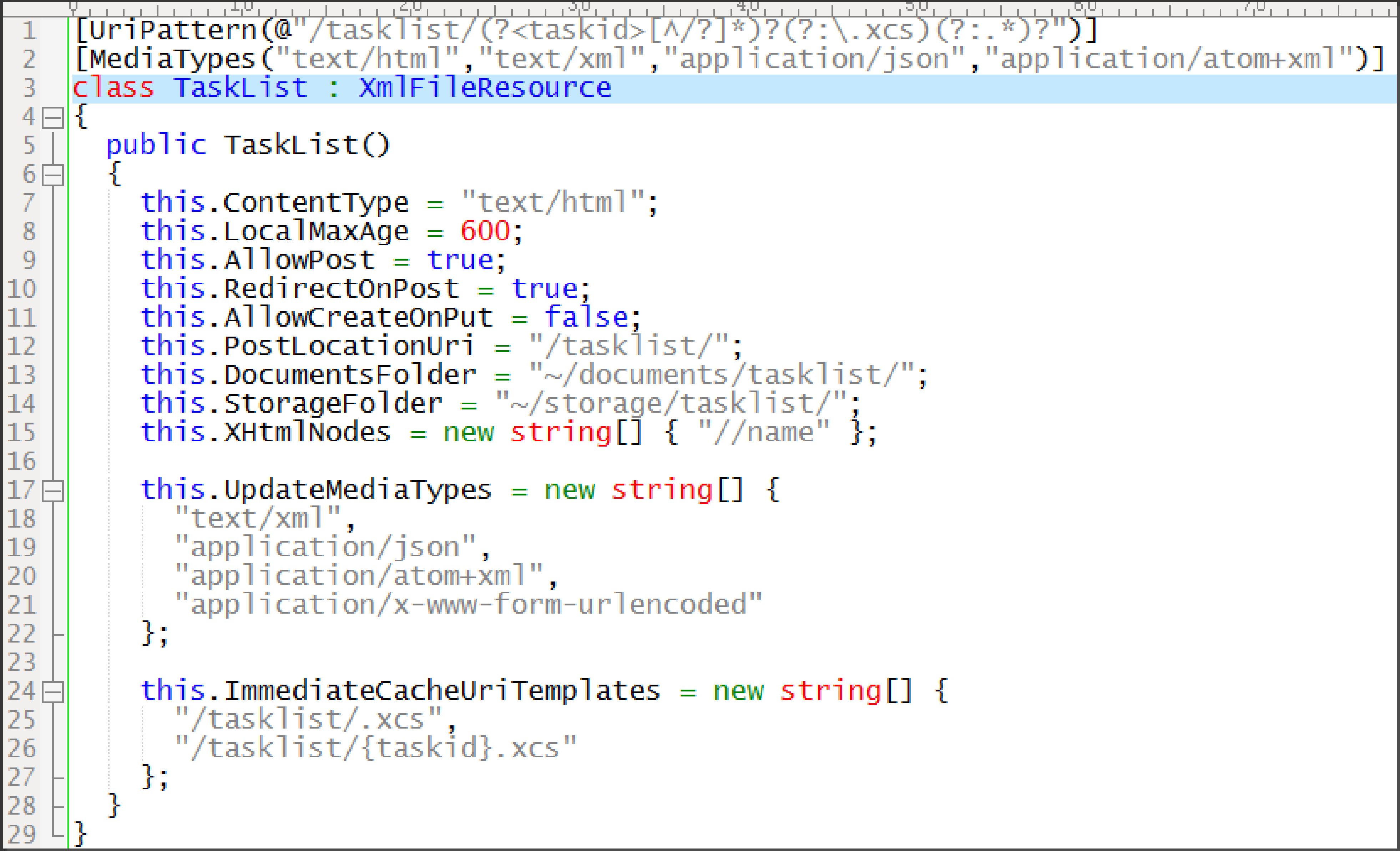


```
private void Delete()
{
  string href = string.Empty;
  string data = string.Empty;

  // get item to delete
  Console.Out.Write("href:");
  href = Console.In.ReadLine();

  // execute delete and show results
  data = client.Execute(href, "delete");
  Console.Out.WriteLine("OK");
}
```

HTTP Is The Interface

Connector | Protocol Techniques

- Reduce HTTP Impedance Mismatch
- Use frameworks that “talk” HTTP
- Avoid libraries that hide HTTP

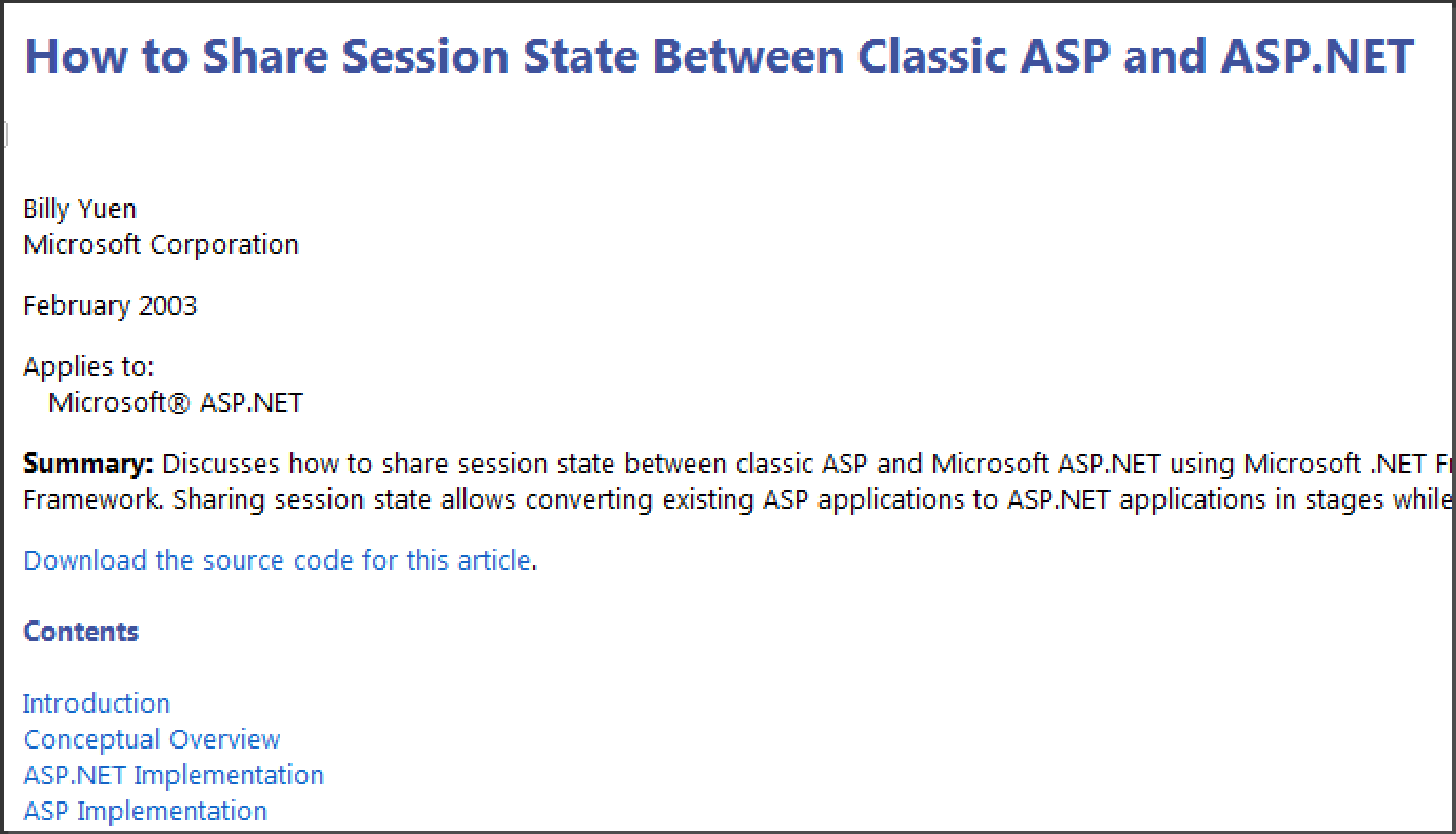


```
[UriPattern(@"/tasklist/(?<taskid>[^/?]*)?(?:\.xcs)(?:.*)?")]
[MediaTypes("text/html","text/xml","application/json","application/atom+xml")]
class TaskList : XmlFileResource
{
  public TaskList()
  {
    this.ContentType = "text/html";
    this.LocalMaxAge = 600;
    this.AllowPost = true;
    this.RedirectOnPost = true;
    this.AllowCreateOnPut = false;
    this.PostLocationUri = "/tasklist/";
    this.DocumentsFolder = "~/documents/tasklist/";
    this.StorageFolder = "~/storage/tasklist/";
    this.XHtmlNodes = new string[] { "//name" };

    this.UpdateMediaTypes = new string[] {
      "text/xml",
      "application/json",
      "application/atom+xml",
      "application/x-www-form-urlencoded"
    };

    this.ImmediateCacheUriTemplates = new string[] {
      "/tasklist/.xcs",
      "/tasklist/{taskid}.xcs"
    };
  }
}
```

Reduce HTTP Impedance Mismatch

Component | State Techniques

- Honor State Boundaries
- Publicly state-less
- privately state-ful


How to Share Session State Between Classic ASP and ASP.NET

Billy Yuen
Microsoft Corporation

February 2003

Applies to:
Microsoft® ASP.NET

Summary: Discusses how to share session state between classic ASP and Microsoft ASP.NET using Microsoft .NET Fr
Framework. Sharing session state allows converting existing ASP applications to ASP.NET applications in stages while

Download the source code for this article.

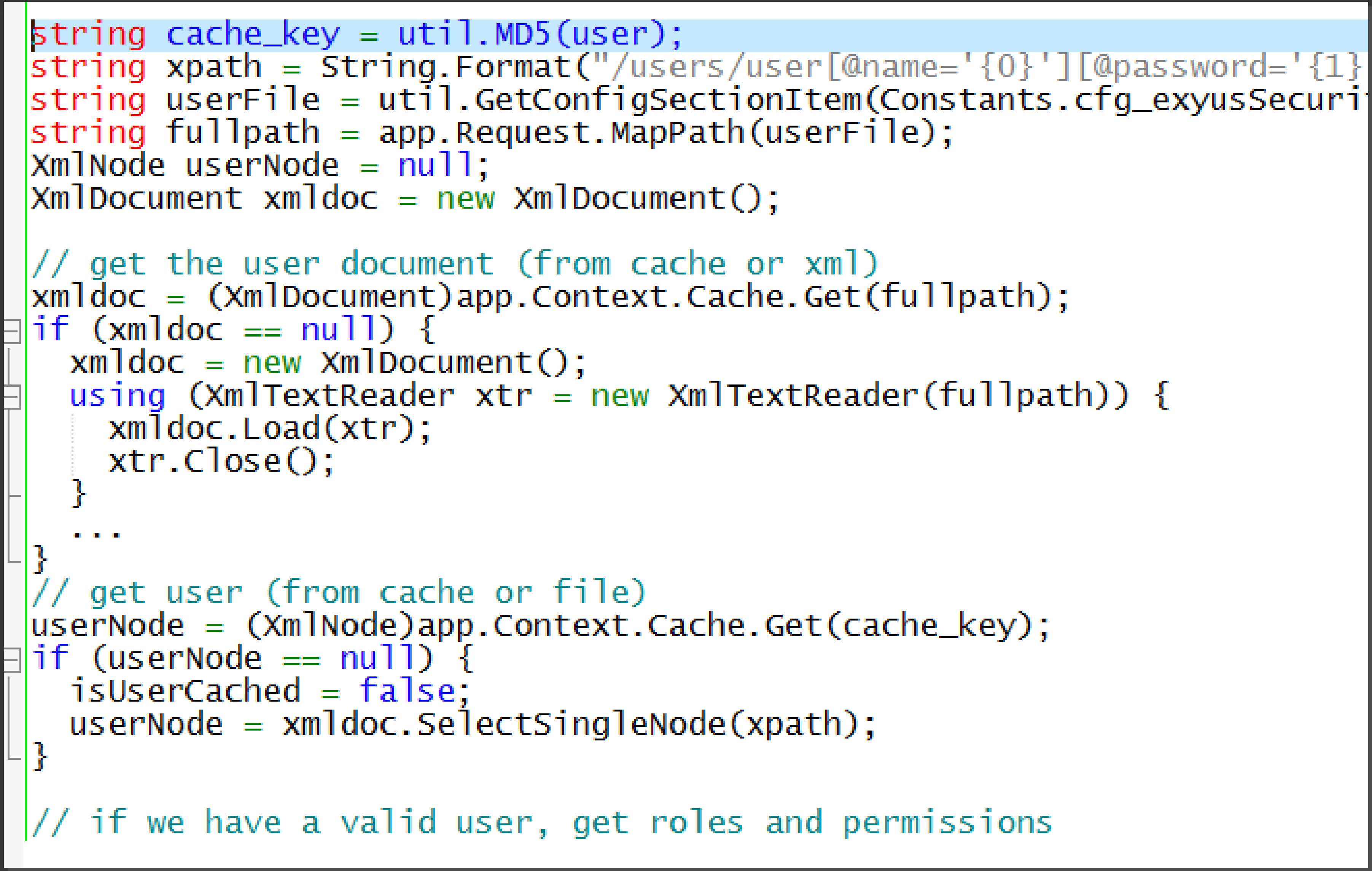
Contents

Introduction
Conceptual Overview
ASP.NET Implementation
ASP Implementation

State Boundaries

Component | State Techniques

- Avoid Session Tracking
- All you need to share is “who”

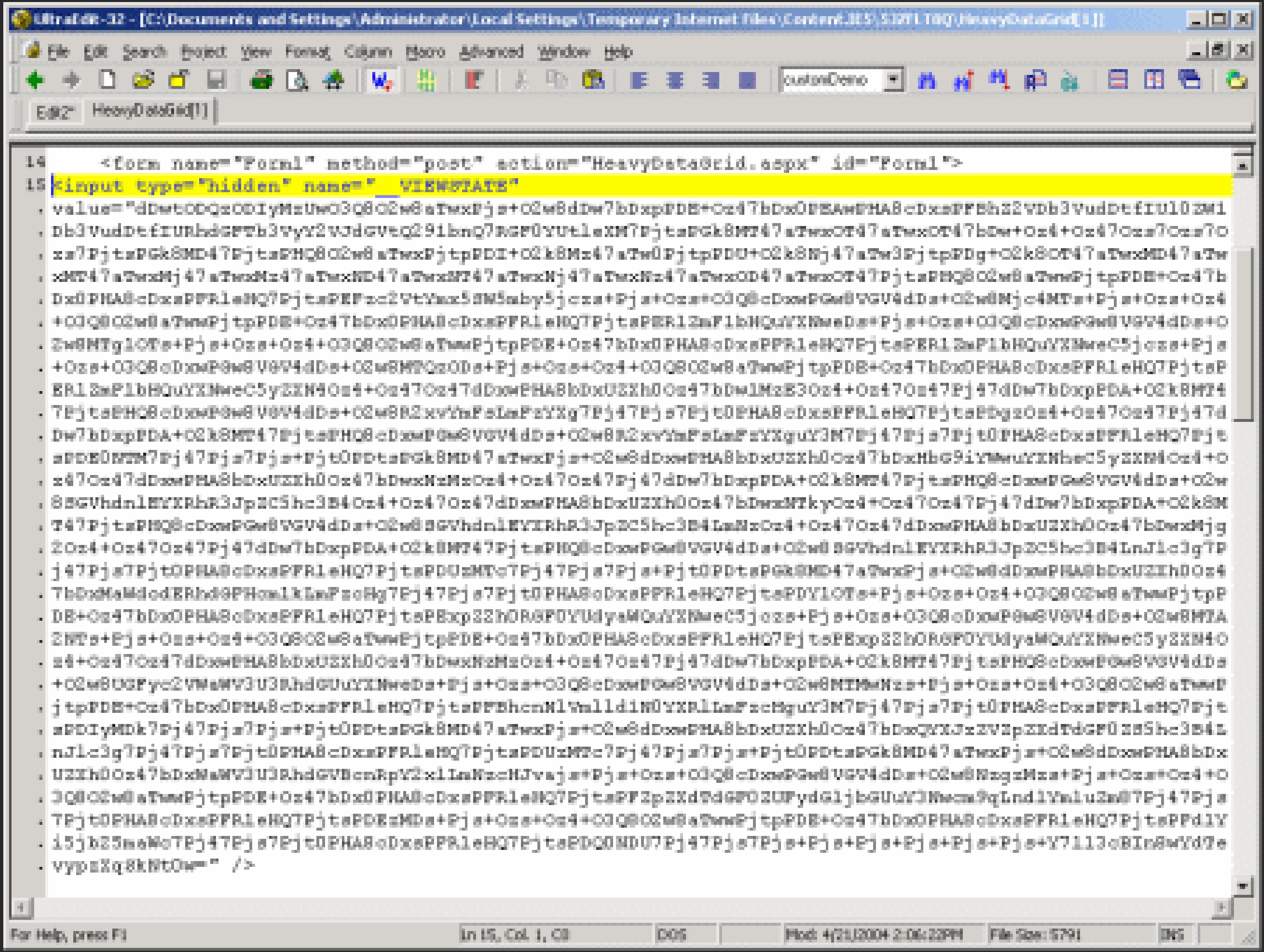


```
string cache_key = util.MD5(user);
string xpath = String.Format("/users/user[@name='{0}'][@password='{1}
string userFile = util.GetConfigSectionItem(Constants.cfg_exyusSecuri
string fullpath = app.Request.MapPath(userFile);
XmlNode userNode = null;
XmlDocument xmldoc = new XmlDocument();

// get the user document (from cache or xml)
xmldoc = (XmlDocument)app.Context.Cache.Get(fullpath);
if (xmldoc == null) {
  xmldoc = new XmlDocument();
  using (XmlTextReader xtr = new XmlTextReader(fullpath)) {
    xmldoc.Load(xtr);
    xtr.Close();
  }
  ...
}
// get user (from cache or file)
userNode = (XmlNode)app.Context.Cache.Get(cache_key);
if (userNode == null) {
  isUserCached = false;
  userNode = xmldoc.SelectSingleNode(xpath);
}

// if we have a valid user, get roles and permissions
```

All you need to share is “who”

Component | State Techniques

- Avoid State “Design” Leaking
- State belongs to components, not connectors



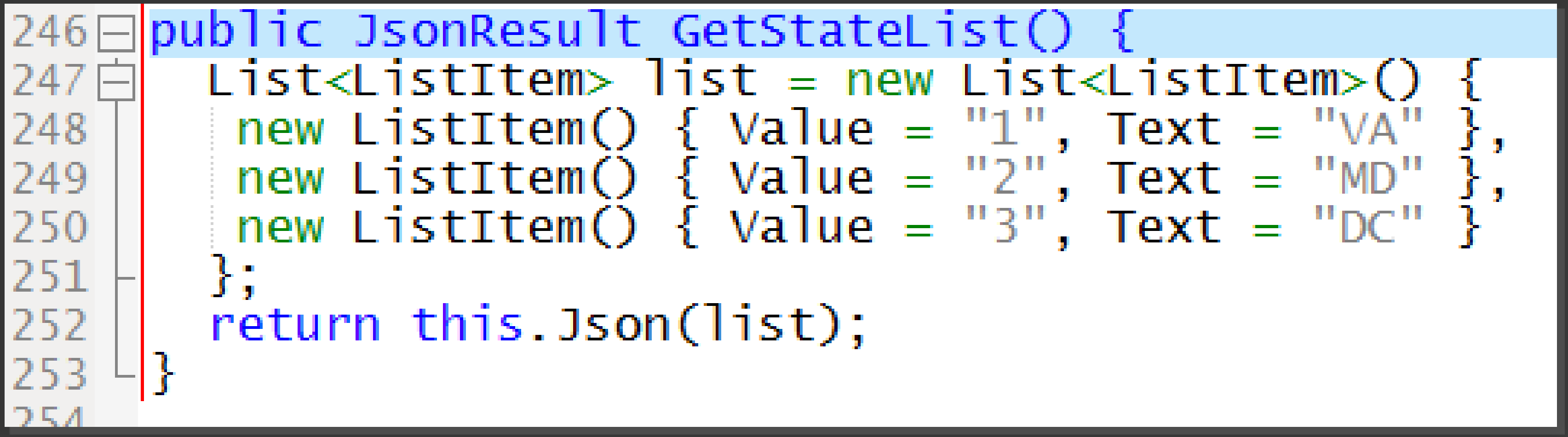
```
    <form name="Form1" method="post" action="HeavyDataGrid.aspx" id="Form1">
<input type="hidden" name="__VIEWSTATE"
```

```
value="dDwtODQzODIyMzUwO3Q8O2w8aTwxPjs+O2w8dDw7bDxpPDE+Oz47bDx0PEAwPHA8cDxsPFBhZ2VDb3VudDtfIUl0ZW1
Db3VudDtfIURhdGFTb3VyY2VJdGVtQ291bnQ7RGF0YUtleXM7Pjt sPGk8MTs7aTwxOT47aTwxOT47bDw+Oz4+Oz47Ozs7Ozs7O
zs7Pjt sPGk8MD47Pjt sPHQ8O2w8aTwxPjtpPDI+O2k8Mz47aTw0PjtpPDU+O2k8Nj47aTw3PjtpPDg+O2k8OT47aTwxMD47aTw
xMT47aTwxMj47aTwxMz47aTwxND47aTwxNT47aTwxNj47aTwxNz47aTwxOD47aTwxOT47Pjt sPHQ8O2w8aTwwPjtpPDE+Oz47b
Dx0PHA8cDxsPFRleHQ7Pjt sPEFsc2VtYmx5SW5mby5jcz s+Pjs+Ozs+O3Q8cDxwPGw8VGV4dDs+O2w8Mjc4MTs+Pjs+Ozs+Oz4
+O3Q8O2w8aTwwPjtpPDE+Oz47bDx0PHA8cDxsPFRleHQ7Pjt sPER1ZmF1bHQuYXNweDs+Pjs+Ozs+O3Q8cDxwPGw8VGV4dDs+O
Zw8MTg1OTs+Pjs+Ozs+Oz4+O3Q8O2w8aTwwPjtpPDE+Oz47bDx0PHA8cDxsPFRleHQ7Pjt sPER1ZmF1bHQuYXNweC5jczs+Pjs
+Ozs+O3Q8cDxwPGw8VGV4dDs+O2w8MTQzODs+Pjs+Ozs+Oz4+O3Q8O2w8aTwwPjtpPDE+Oz47bDx0PHA8cDxsPFRleHQ7PjtsP
ER1ZmF1bHQuYXNweC5yZXN4Oz4+Oz47Oz47dDxwPHA8bDxUZXh0Oz47bDwlMzE3Oz4+Oz47Oz47Pj47dDw7bDxpPDA+O2k8MT4
7PjtsPHQ8cDxwPGw8VGV4dDs+O2w8R2xvYmFsLmFzYXg7Pj47Pjs7PjtOPHA8cDxsPFRleHQ7PjtsPDgzOz4+Oz47Oz47Pj47d
Dw7bDxpPDA+O2k8MT47PjtsPHQ8cDxwPGw8VGV4dDs+O2w8R2xvYmFsLmFzYXguY3M7Pj47Pjs7PjtOPHA8cDxsPFRleHQ7Pjt
sPDE0NTM7Pj47Pjs7Pjs+PjtOPDtsPGk8MD47aTwxPjs+O2w8dDxwPHA8bDxUZXh0Oz47bDxMbG9iYWwuYXNheC5yZXN4Oz4+O
z47Oz47dDxwPHA8bDxUZXh0Oz47bDwxNzMzOz4+Oz47Oz47Pj47dDw7bDxpPDA+O2k8MT47PjtsPHQ8cDxwPGw8VGV4dDs+O2w
8SGVhdnlEYXRhR3JpZC5hc3B4Oz4+Oz47Oz47dDxwPHA8bDxUZXh0Oz47bDwxNTkyOz4+Oz47Oz47Pj47dDw7bDxpPDA+O2k8M
T47PjtsPHQ8cDxwPGw8VGV4dDs+O2w8SGVhdnlEYXRhR3JpZC5hc3B4LmNzOz4+Oz47Oz47dDxwPHA8bDxUZXh0Oz47bDwxMjg
2Oz4+Oz47Oz47Pj47dDw7bDxpPDA+O2k8MT47PjtsPHQ8cDxwPGw8VGV4dDs+O2w8SGVhdnlEYXRhR3JpZC5hc3B4LnJlc3g7P
j47Pjs7PjtOPHA8cDxsPFRleHQ7PjtsPDUzMTc7Pj47Pjs7Pjs+PjtOPDtsPGk8MD47aTwxPjs+O2w8dDxwPHA8bDxUZXh0Oz4
7bDxMaWdodERhdGFHcmlkLmFzcHg7Pj47Pjs7PjtOPHA8cDxsPFRleHQ7PjtsPDY1OTs+Pjs+Ozs+Oz4+O3Q8O2w8aTwwPjtpP
DE+Oz47bDx0PHA8cDxsPFRleHQ7PjtsPExpZ2h0RGF0YUdyaWQuYXNweC5jczs+Pjs+Ozs+O3Q8cDxwPGw8VGV4dDs+O2w8MTA
ZNTs+Pjs+Ozs+Oz4+O3Q8O2w8aTwwPjtpPDE+Oz47bDx0PHA8cDxsPFRleHQ7PjtsPExpZ2h0RGF0YUdyaWQuYXNweC5yZXN4O
z4+Oz47Oz47dDxwPHA8bDxUZXh0Oz47bDwxNzMzOz4+Oz47Oz47Pj47dDw7bDxpPDA+O2k8MT47PjtsPHQ8cDxwPGw8VGV4dDs
+O2w8UGFyc2VNaWV3U3RhdGUuYXNweDs+Pjs+Ozs+O3Q8cDxwPGw8VGV4dDs+O2w8MTMwNzs+Pjs+Ozs+Oz4+O3Q8O2w8aTwwP
jtpPDE+Oz47bDx0PHA8cDxsPFRleHQ7PjtsPFBhcnNlVmlldlN0YXRlLmFzcHguY3M7Pj47Pjs7PjtOPHA8cDxsPFRleHQ7Pjt
sPDIyMDk7Pj47Pjs7Pjs+PjtOPDtsPGk8MD47aTwxPjs+O2w8dDxwPHA8bDxUZXh0Oz47bDxQYXJzZVZpZXdTdGF0ZS5hc3B4L
nJlc3g7Pj47Pjs7PjtOPHA8cDxsPFRleHQ7PjtsPDUzMTc7Pj47Pjs7Pjs+PjtOPDtsPGk8MD47aTwxPjs+O2w8dDxwPHA8bDx
UZXh0Oz47bDxWaWV3U3RhdGVBcnRpY2xlLmNzcHJvajs+Pjs+Ozs+O3Q8cDxwPGw8VGV4dDs+O2w8NzgzMzs+Pjs+Ozs+Oz4+O
3Q8O2w8aTwwPjtpPDE+Oz47bDx0PHA8cDxsPFRleHQ7PjtsPFZpZXdTdGF0ZUFydGljbGUuY3Nwcm9qLndlYmluZm87Pj47Pjs
7PjtOPHA8cDxsPFRleHQ7PjtsPDEzMDs+Pjs+Ozs+Oz4+O3Q8O2w8aTwwPjtpPDE+Oz47bDx0PHA8cDxsPFRleHQ7PjtsPFd1Y
i5jb25maWc7Pj47Pjs7PjtOPHA8cDxsPFRleHQ7PjtsPDQ0NDU7Pj47Pjs7Pjs+Pjs+Pjs+Pjs+Pjs+Pjs+Y7113cBIn8wYdTe
VypzXq8kNtOw=" />
```

Avoid Leaking State Design

Data | Domain Techniques

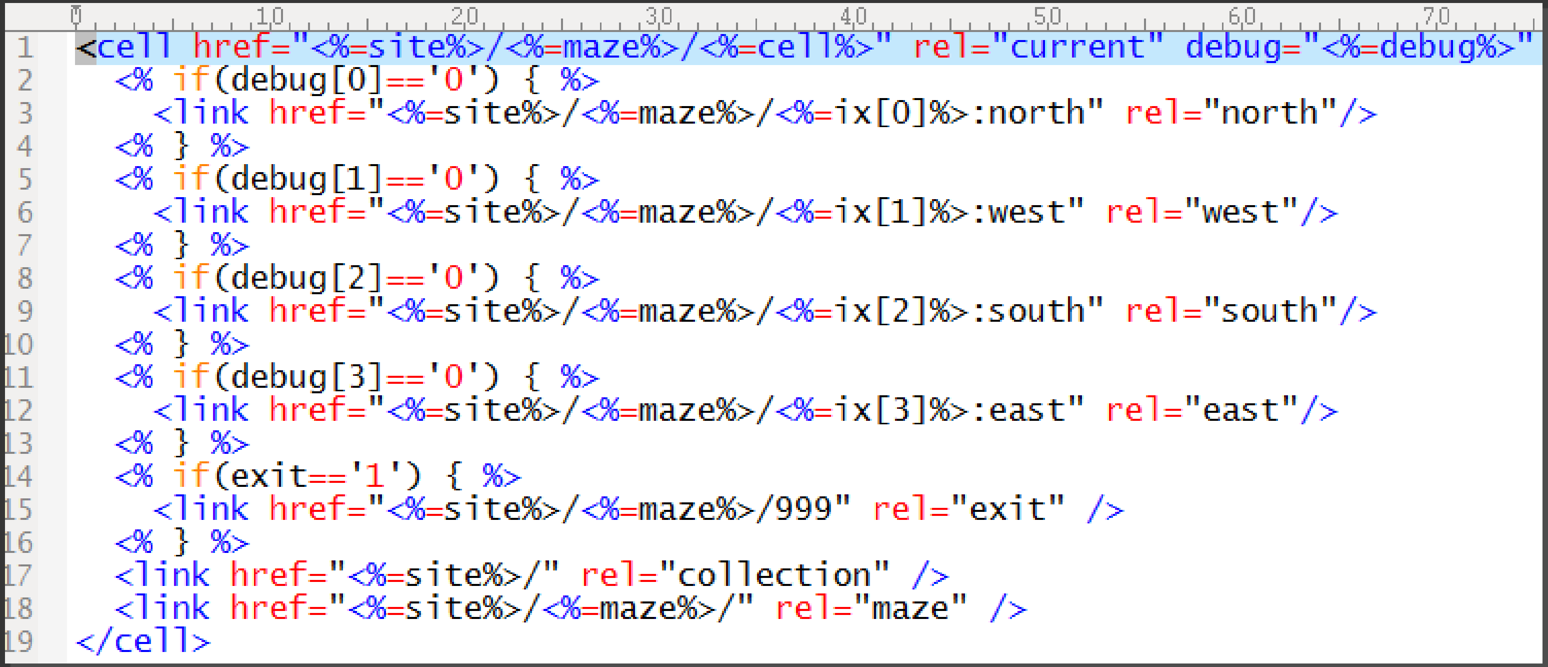
- Avoid Type Marshalling|Object Serialization



```
public JsonResult GetStateList() {
    List<ListItem> list = new List<ListItem>() {
     new ListItem() { Value = "1", Text = "VA" },
     new ListItem() { Value = "2", Text = "MD" },
     new ListItem() { Value = "3", Text = "DC" }
    };
    return this.Json(list);
}
```

Data | Domain Techniques

- Avoid Type Marshalling|Object Serialization
  - Use Templating Instead



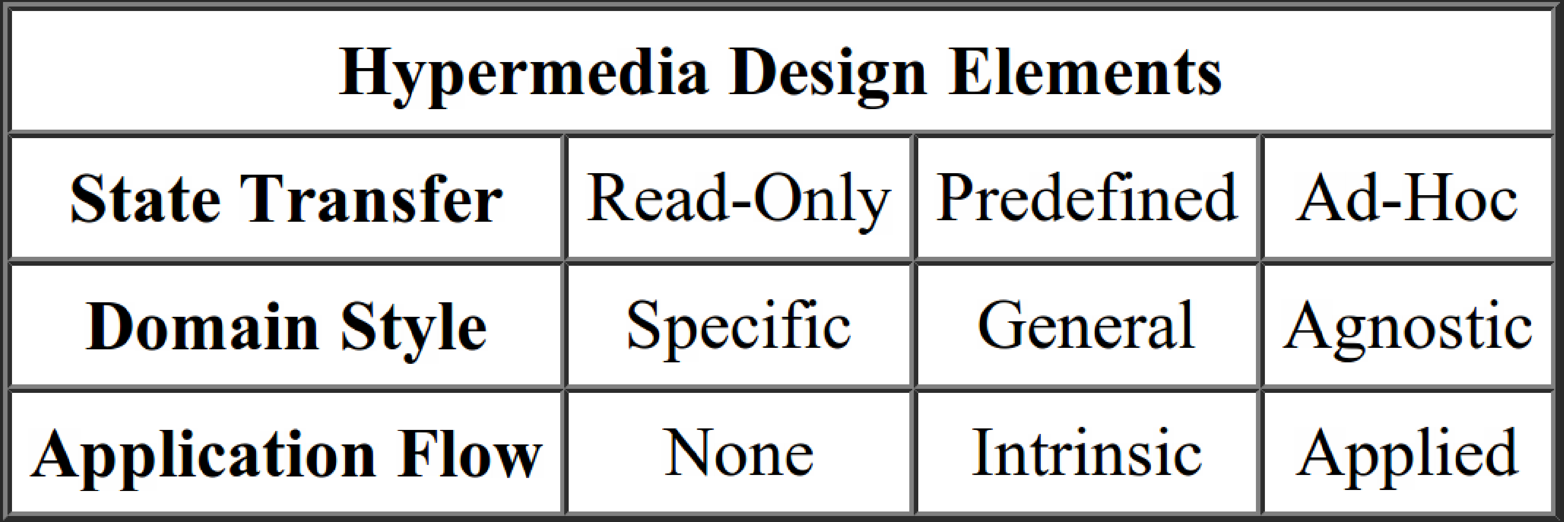
```
<cell href="<%=site%>/<%=maze%>/<%=cell%>" rel="current" debug="<%=debug%>"
  <% if(debug[0]=='0') { %>
    <link href="<%=site%>/<%=maze%>/<%=ix[0]%>:north" rel="north"/>
  <% } %>
  <% if(debug[1]=='0') { %>
    <link href="<%=site%>/<%=maze%>/<%=ix[1]%>:west" rel="west"/>
  <% } %>
  <% if(debug[2]=='0') { %>
    <link href="<%=site%>/<%=maze%>/<%=ix[2]%>:south" rel="south"/>
  <% } %>
  <% if(debug[3]=='0') { %>
    <link href="<%=site%>/<%=maze%>/<%=ix[3]%>:east" rel="east"/>
  <% } %>
  <% if(exit=='1') { %>
    <link href="<%=site%>/<%=maze%>/999" rel="exit" />
  <% } %>
  <link href="<%=site%>/" rel="collection" />
  <link href="<%=site%>/<%=maze%>/" rel="maze" />
</cell>
```

Data | Domain Techniques

- Use MVC Wisely
- “Fat” M (not just DB serialization)
- “Loose” V (templates, not codecs)
- “Decoupled” C (SoC for addressing)

Data | Domain Techniques

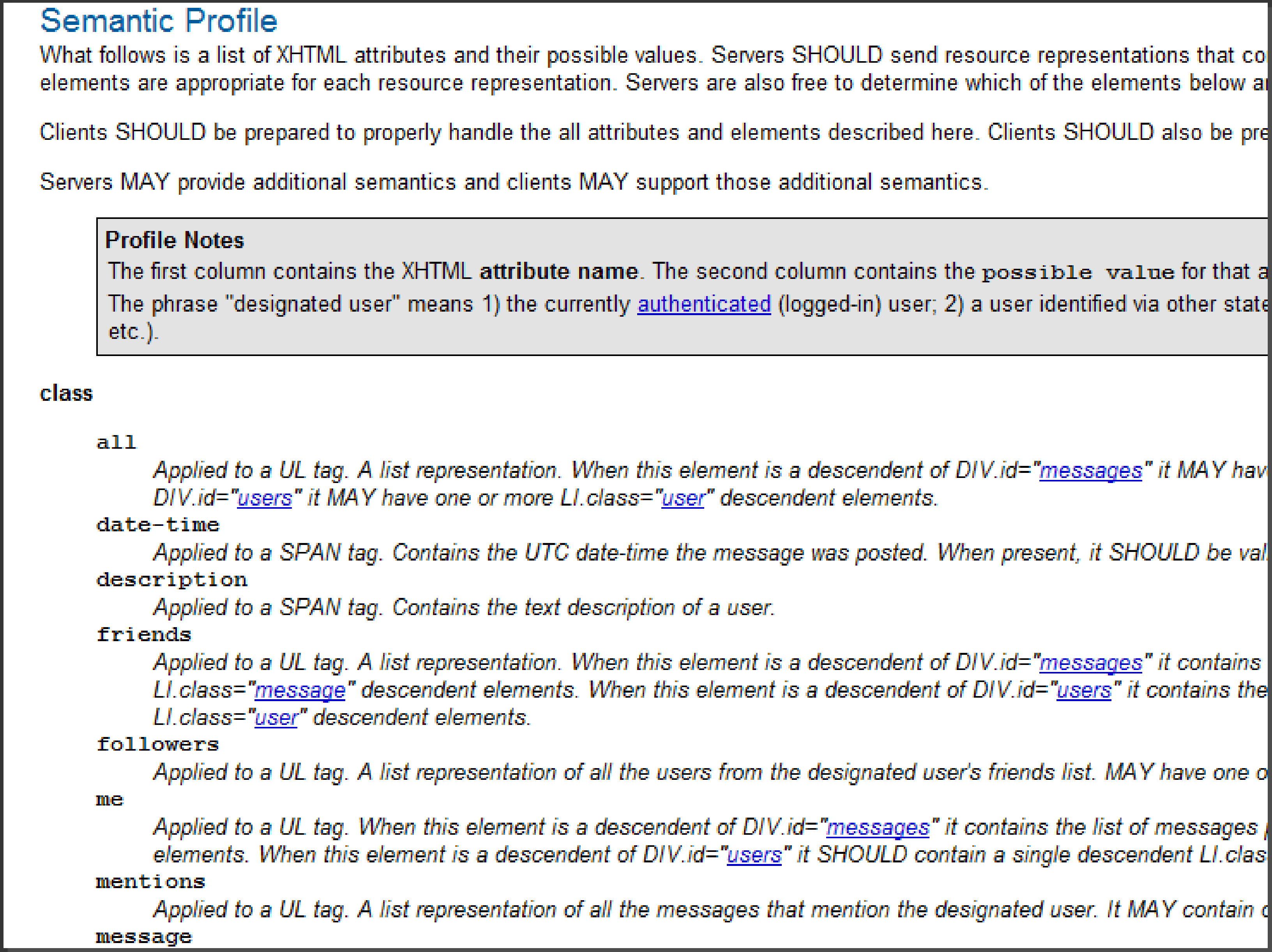
- Maximize Hypermedia
  - State Transfer
  - Domain Style
  - App Flow



Hypermedia Design Elements			
State Transfer	Read-Only	Predefined	Ad-Hoc
Domain Style	Specific	General	Agnostic
Application Flow	None	Intrinsic	Applied

Data | Domain Techniques

- Model w/ Media Types
  - Document Media Type, not interactions
  - Replace RPC designs w/ Media Type designs



Semantic Profile

What follows is a list of XHTML attributes and their possible values. Servers SHOULD send resource representations that co
elements are appropriate for each resource representation. Servers are also free to determine which of the elements below ar

Clients SHOULD be prepared to properly handle the all attributes and elements described here. Clients SHOULD also be pre

Servers MAY provide additional semantics and clients MAY support those additional semantics.

Profile Notes
The first column contains the XHTML **attribute name**. The second column contains the `possible value` for that a
The phrase "designated user" means 1) the currently authenticated (logged-in) user; 2) a user identified via other state
etc.).

class

`all`
Applied to a UL tag. A list representation. When this element is a descendent of DIV.id="messages" it MAY hav
DIV.id="users" it MAY have one or more LI.class="user" descendent elements.

`date-time`
Applied to a SPAN tag. Contains the UTC date-time the message was posted. When present, it SHOULD be val

`description`
Applied to a SPAN tag. Contains the text description of a user.

`friends`
Applied to a UL tag. A list representation. When this element is a descendent of DIV.id="messages" it contains
LI.class="message" descendent elements. When this element is a descendent of DIV.id="users" it contains the
LI.class="user" descendent elements.

`followers`
Applied to a UL tag. A list representation of all the users from the designated user's friends list. MAY have one o

`me`
Applied to a UL tag. When this element is a descendent of DIV.id="messages" it contains the list of messages
elements. When this element is a descendent of DIV.id="users" it SHOULD contain a single descendent LI.clas

`mentions`
Applied to a UL tag. A list representation of all the messages that mention the designated user. It MAY contain c

`message`

Model with Media Types

Summary

Summary

- How can we make implementations more flexible and stable over time?
- Make time your ally
- Design “living” systems
- Embrace Connector+Component+Data Model
- Adopt techniques that
  - Embrace the protocol
  - Play to strengths in each design element (CCD)
  - Recognize clear boundaries

Design … Discipline

“Design without discipline is anarchy, an exercise of irresponsibility”

Massimo Vignelli, 2010

Beyond REST

An approach to creating stable, evolve-able Web applications

Mike Amundsen
@mamund