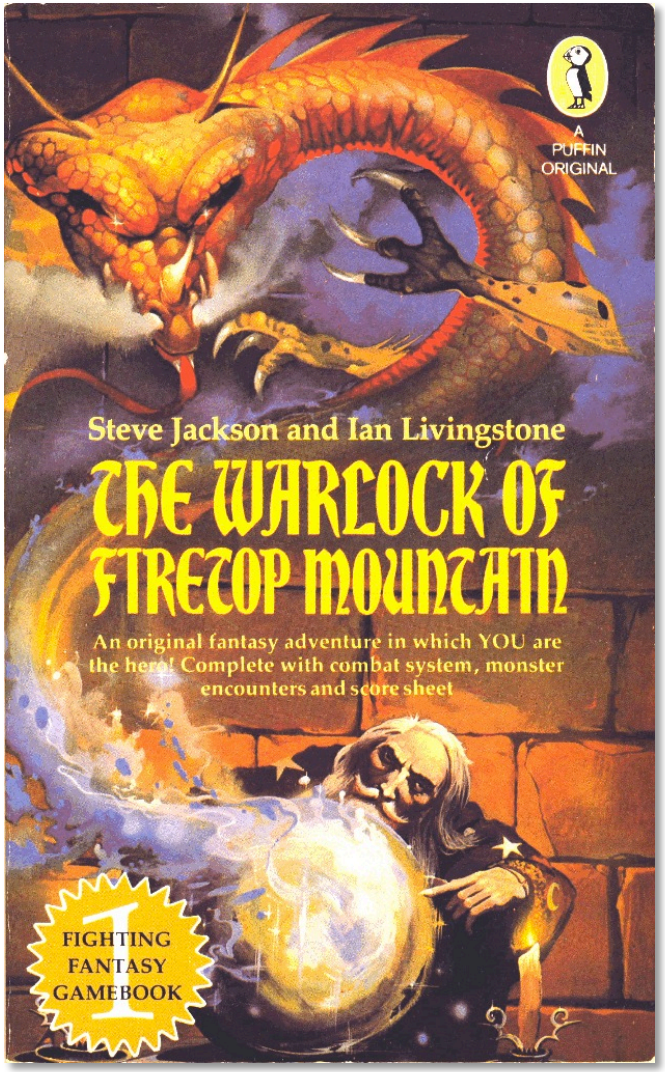
Getting Things Done with REST

http://ian**S**robinson.com
@ian**S**robinson

Getting Things Done

Pick your path to adventure



79–80

hinges. Listening at the door, you hear strange mutterings and the clatter of what could be pots and pans. Whatever is in there, there are several of them. Do you want to go through the door (turn to **159**) or turn back (turn to **237**)?

79

The passageway ends in front of you in a dead end If you wish to search for secret passageways, turn to **137**. If not, return to the crossroads at **267**

80

The key fits the lock and opens the door You find yourself in a large boathouse. Various boats, in different stages of construction, are lying around Apart from the door behind you there is another in the north wall. As you enter, the Skeletons stop their work and crane their bony necks around to look at you. They pick up planks of wood and hammers and advance towards you There are five of them. Do you.

Smile nervously and back out of the door into the passage? Turn to **129**
Tell them you've come about buying a boat? Turn to **123**
Tell them you're their new boss and order them back to work? Turn to **195**
Draw your sword and prepare for battle? Turn to **140**

195–197

flies through the air directly at him, stops centimetres from his chest and falls to the floor. He looks up and smiles at you with an evil, gloating smile. What can you do:

Draw your sword and advance? Turn to **142**
Try something else from your backpack? Turn to **105**

195

This is a rather unlikely story, considering that they see very few humans around. Nevertheless, Skeletons are pretty dim – you knew this and that's why you tried the story. Roll one die. If you roll a 1 or 2, they don't believe you and keep on advancing. Turn to **140**.

A 3 or 4 means that they aren't sure, and send two of their number off through the north door whilst the rest hold you at bay with their weapons. Turn to **164**

A roll of 5 or 6 means they've believed you and they all get back to work! Turn to **9**. Add 2 LUCK points

196

You search the room. Try as you may you cannot find the secret switch to open the door in the bookshelf – the old man must have locked it from the inside. You do find 5 Gold Pieces in a drawer in the table. You decide to return to the junction to the south Turn to **280**

197

At the top of the stairs the passage turns sharply to

Executing a specialism in a generalized way

Warlock of Firetop Mountain Protocol

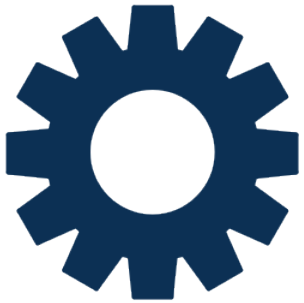
- Go north
- Defeat goblin
- Take key
- Unlock door
- Go east
- Solve riddle

Fighting Fantasy Transfer Protocol

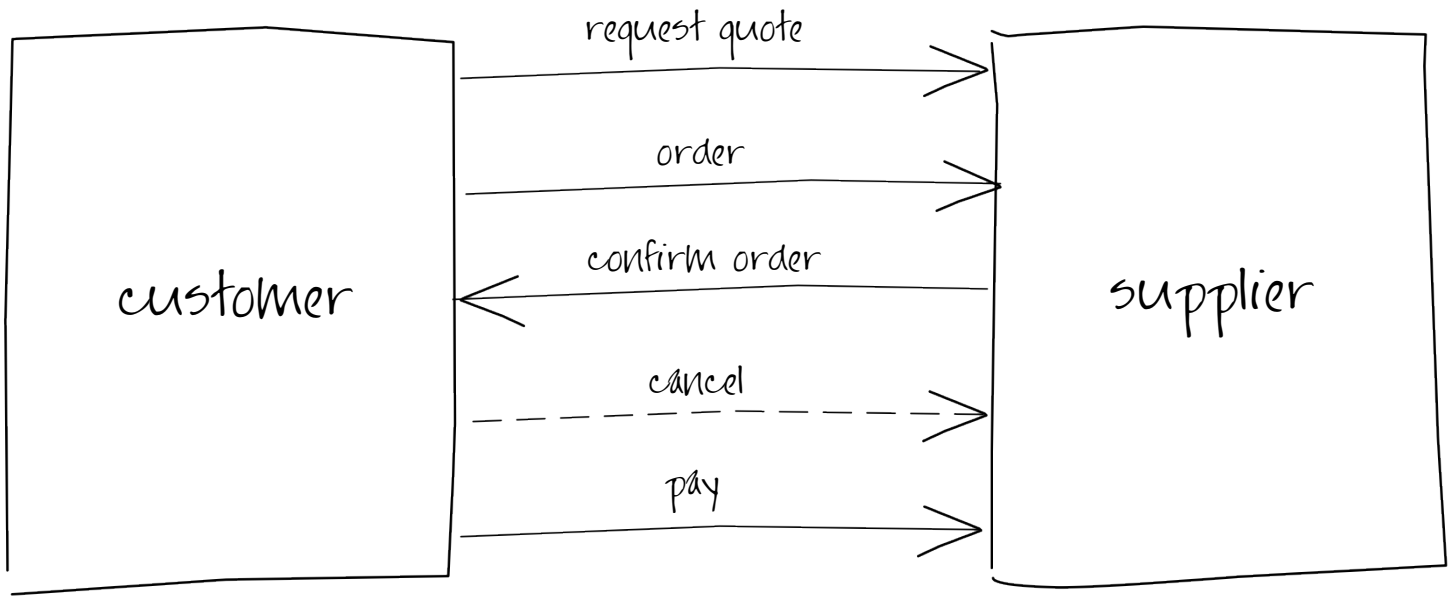
- Numbered prose paragraphs
- Multiple choices keyed to numbered paragraphs

The hypermedia constraint

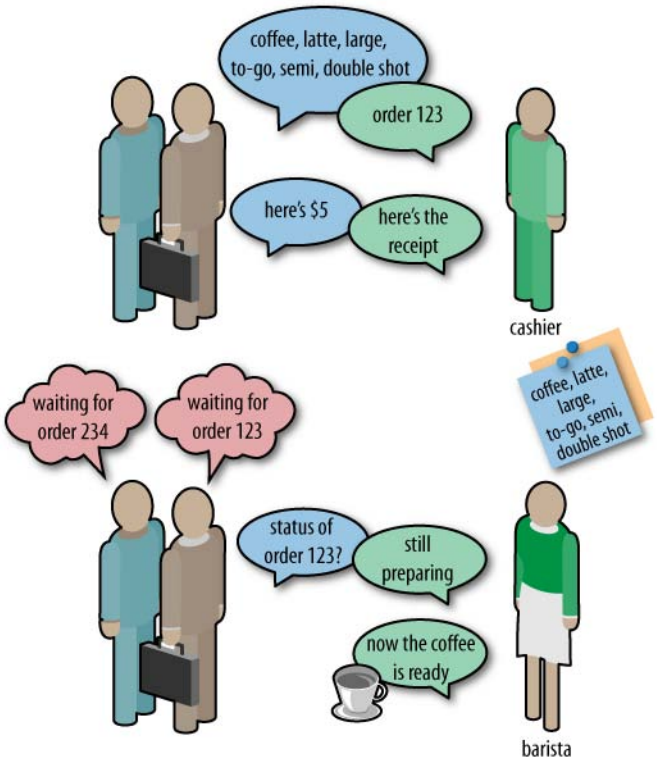
Hypermedia
As
the
Engine
of
Application
State

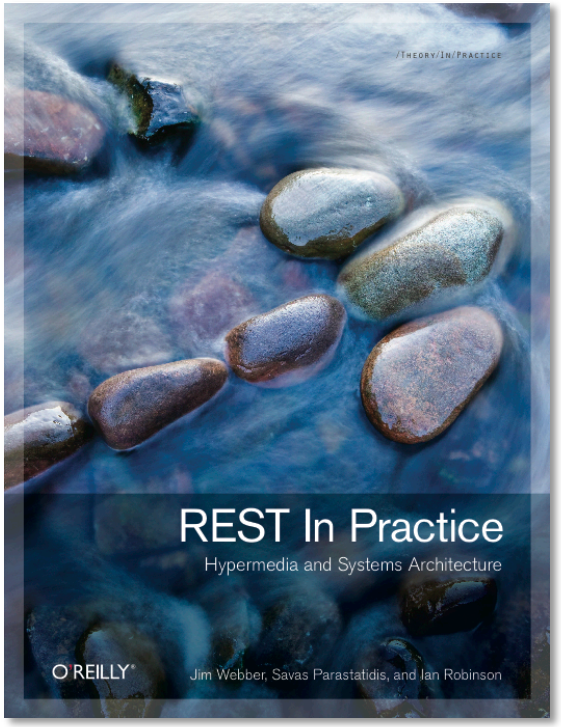


Procurement process



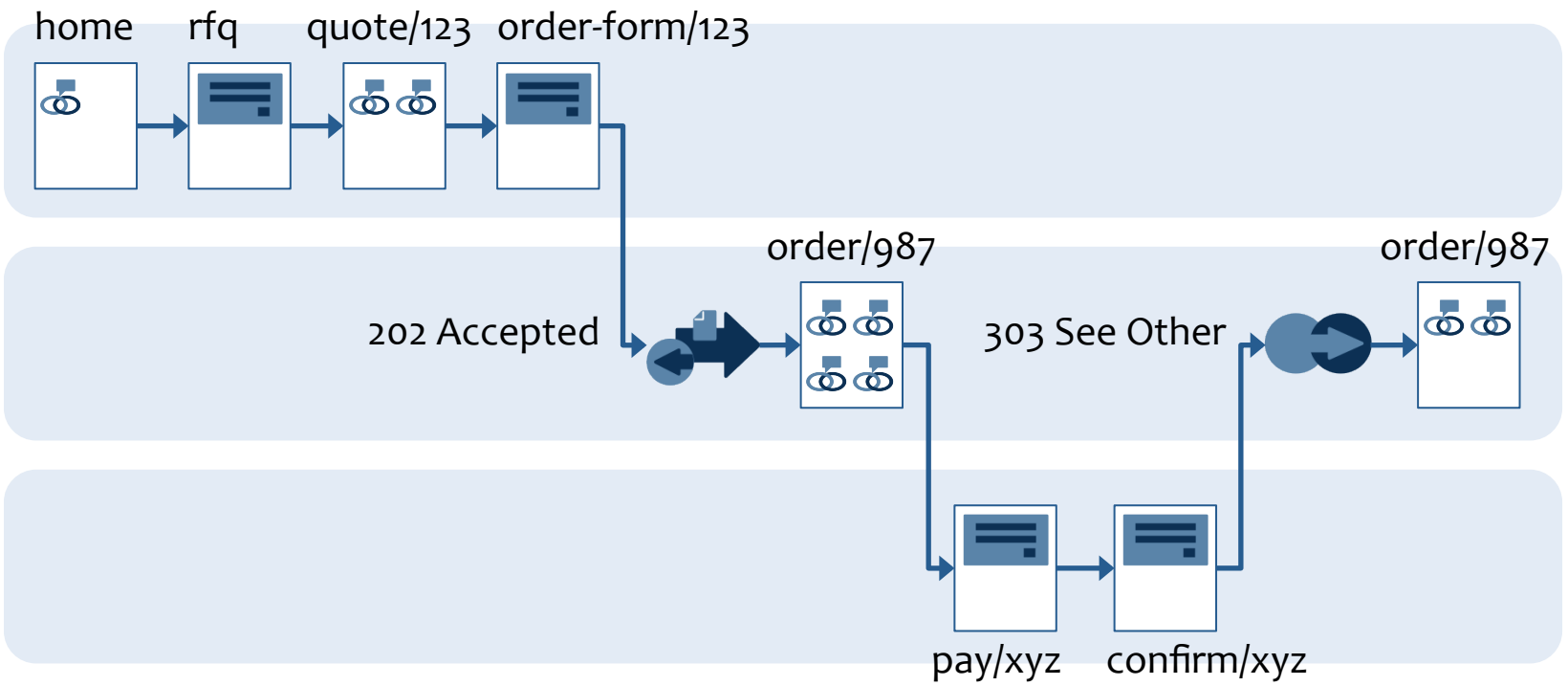
Restbucks


coffee, latte, large, to-go, semi, double shot
order 123
here's $5
here's the receipt
cashier
coffee, latte, large, to-go, semi, double shot
waiting for order 234
waiting for order 123
status of order 123?
still preparing
now the coffee is ready
barista

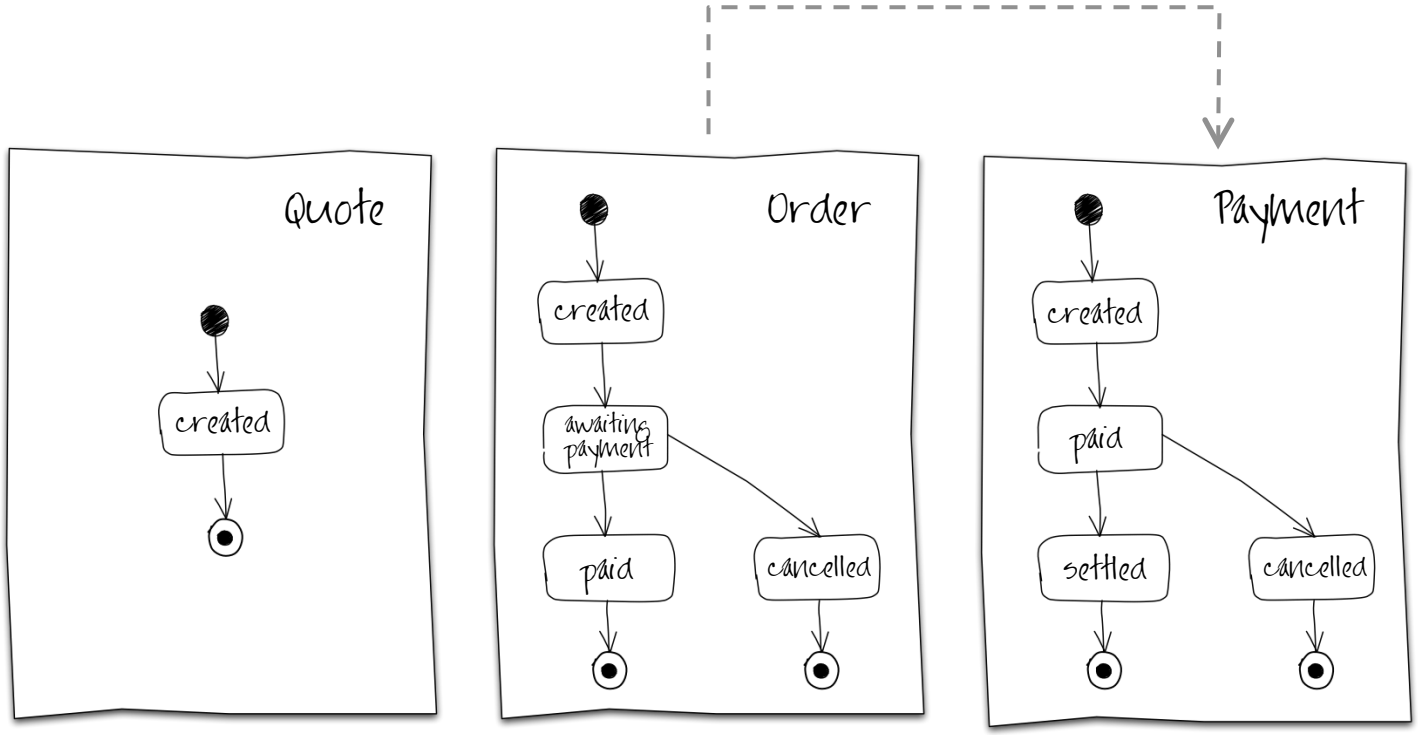

/Theory/In/Practice
REST In Practice
Hypermedia and Systems Architecture
O'REILLY
Jim Webber, Savas Parastatidis, and Ian Robinson

Server-side Development

Divide and conquer



Resources look after themselves



Quote resource

```
[ServiceContract]
public class Quote
{
  private readonly UriFactory uriFactory;
  private readonly IQuotationEngine quoteEngine;

  public Quote(UriFactory uriFactory, IQuotationEngine quoteEngine)
  {
    this.uriFactory = uriFactory;
    this.quoteEngine = quoteEngine;
  }

  [WebGet(UriTemplate = "{id}")]
  public Shop Get(string id, HttpRequestMessage request,
                  HttpResponseMessage response)
  {
    //Get quotation from quotation engine
    //Add HTTP headers to response
    //Return entity body
  }
}
```

Microsoft WCF Web APIs
http://wcf.codeplex.com

Develop them test-by-test

```
[Test]
public void ShouldReturn404NotFoundWhenGettingQuoteThatDoesNotExist()
{
  var id = Guid.Empty;

  var quoteEngine = MockRepository.GenerateStub<IQuotationEngine>();
  quoteEngine.Stub(e => e.GetQuote(id))
    .Throw(new KeyNotFoundException());

  var response = new HttpResponseMessage();

  var quoteResource = new Quote(DefaultUriFactory.Instance, quoteEngine);
  quoteResource.Get(
    id.ToString("N"),
    new HttpRequestMessage(),
    response);

  Assert.AreEqual(HttpStatusCode.NotFound, response.StatusCode);
}
```

Return 404 when quotation doesn't exist

```
public class Quote
{
  private readonly IQuotationEngine quoteEngine;

  ...

  [WebGet]
  public Shop Get(string id, HttpRequestMessage request,
                  HttpResponseMessage response)
  {
    Quotation quote;
    try
    {
      quote = quoteEngine.GetQuote(new Guid(id));
    }
    catch (KeyNotFoundException)
    {
      response.StatusCode = HttpStatusCode.NotFound;
      return null;
    }

    ...
  }
}
```

Test caching headers

```
[Test]
public void ResponseShouldExpire7DaysFromDateTimeQuoteWasCreated()
{
  var id = Guid.Empty;

  DateTimeOffset createdDateTime = DateTime.Now;
  var expiryDateTime = createdDateTime.AddDays(7.00) .UtcDateTime;

  var quoteEngine = MockRepository.GenerateStub<IQuotationEngine>();
  quoteEngine.Stub(e => e.GetQuote(id)).Return(
    new Quotation(id, createdDateTime, new LineItem[]{}));

  var response = new HttpResponseMessage();

  var quote = new Quote(DefaultUriFactory.Instance, quoteEngine);
  quote.Get(
    id.ToString("N"),
    new HttpRequestMessage{Uri = new Uri("http://localhost/quote/")},
    response);

  Assert.AreEqual("public", response.Headers.CacheControl.ToString());
  Assert.AreEqual(expiryDateTime, response.Headers.Expires);
}
```

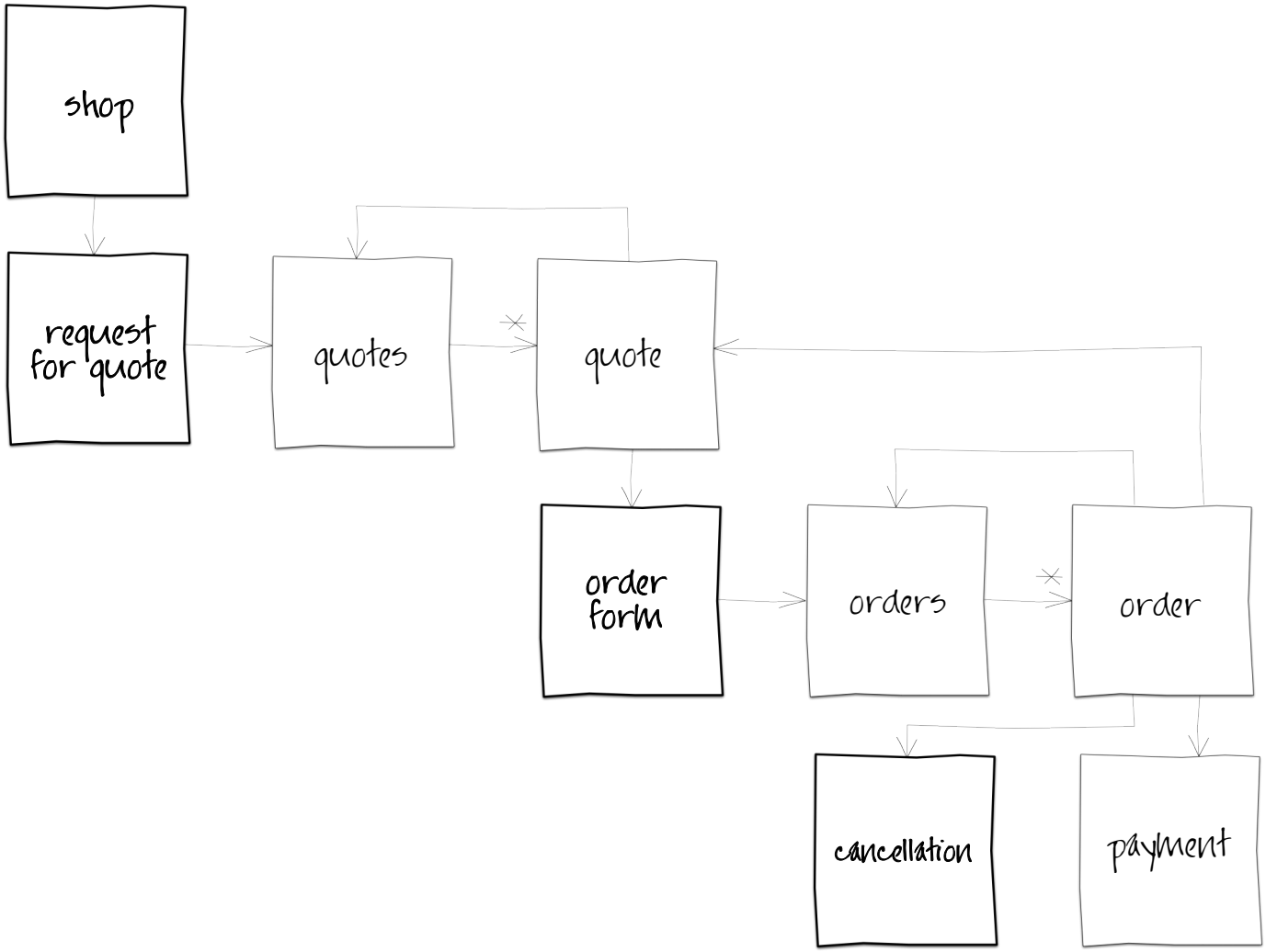
Add caching headers

```
public class Quote
{
   ...

   [WebGet]
   public Shop Get(string id, HttpRequestMessage request,
                   HttpResponseMessage response)
   {
     //Retrieve quote
     ...

     response.StatusCode = HttpStatusCode.OK;
     response.Headers.CacheControl = new CacheControl {Public = true};
     response.Headers.Expires = quote.CreatedDateTime
       .AddDays(7.0).UtcDateTime;

     ...
   }
}
```

Quote resource

```
[ServiceContract]
[UriTemplate("quote", "{id}")]
public class Quote
{
  private readonly UriFactory uriFactory;
  private readonly IQuotationEngine quoteEngine;

  public Quote(UriFactory uriFactory, IQuotationEngine quoteEngine)
  {
    this.uriFactory = uriFactory;
    this.quoteEngine = quoteEngine;
  }

  [WebGet]
  public Shop Get(string id, HttpRequestMessage request, HttpResponseMessage response)
  {
    Quotation quote;
    try
    {
      quote = quoteEngine.GetQuote(new Guid(id));
    }
    catch (KeyNotFoundException)
    {
      response.StatusCode = HttpStatusCode.NotFound;
      return null;
    }

    response.StatusCode = HttpStatusCode.OK;
    response.Headers.CacheControl = new CacheControl {Public = true};
    response.Headers.Expires = quote.CreatedDateTime.AddDays(7.0).UtcDateTime;

    var baseUri = uriFactory.CreateBaseUri<Quote>(request.Uri);

    return new ShopBuilder(baseUri, quote.LineItems.Select(
        li => new LineItemToItem(li).Adapt()))
      .AddLink(new Link(uriFactory.CreateRelativeUri<Quote>(quote.Id),
        RestbucksMediaType.Value, LinkRelations.Self))
      .AddLink(new Link(uriFactory.CreateRelativeUri<OrderForm>(quote.Id),
        RestbucksMediaType.Value, LinkRelations.OrderForm)).Build();
  }
}
```

Resources != domain



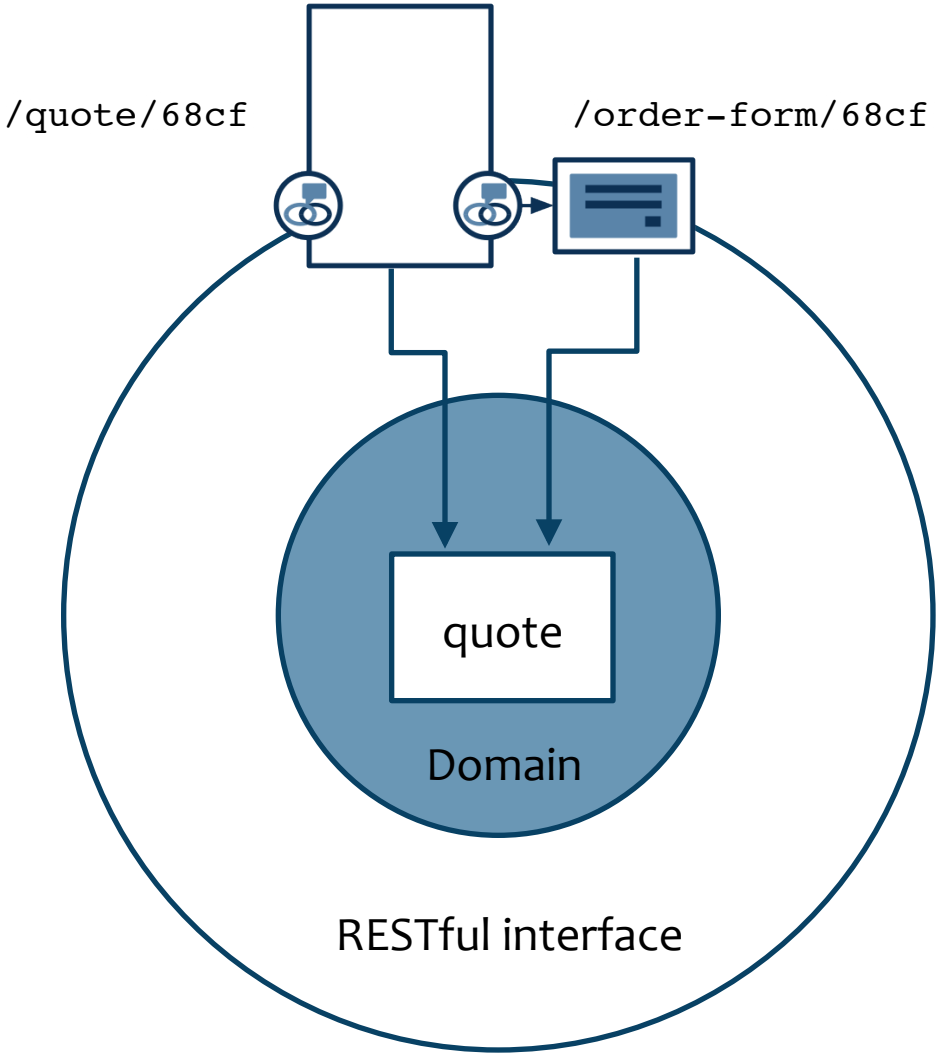
Quote

```
<shop xmlns:rb="http://relations.restbucks.com/"
      xml:base="http://restbucks.com/"
      xmlns="http://schemas.restbucks.com/shop">
  <items>
    <item>
      <description>coffee</description>
      <amount measure="g">125</amount>
      <price currency="GBP">1.25</price>
    </item>
  </items>
  <link rel="self"
        type="application/vnd.restbucks+xml"
        href="quote/68cff6e75a09474fa0098c9393aa6d4e" />
  <link rel="rb:order-form"
        type="application/vnd.restbucks+xml"
        href="order-form/68cff6e75a09474fa0098c9393aa6d4e" />
</shop>
```

Order form

```
<shop xml:base="http://restbucks.com/"
      xmlns="http://schemas.restbucks.com/shop">
  <model id="order" xmlns="http://www.w3.org/2002/xforms">
    <instance>
      <shop xml:base="http://restbucks.com/"
            xmlns="http://schemas.restbucks.com/shop">
        <items>
          <item>
            <description>coffee</description>
            <amount measure="g">125</amount>
            <price currency="GBP">1.25</price>
          </item>
        </items>
        <link rel="self"
              type="application/vnd.restbucks+xml"
              href="quote/68cff6e75a09474fa0098c9393aa6d4e" />
      </shop>
    </instance>
    <submission resource="/orders/?c=12345&amp;s=325"
                method="post"
                mediatype="application/vnd.restbucks+xml" />
  </model>
</shop>
```

Resources adapt the domain for hypermedia clients



Problem: URI redundancy

Routes

```
RouteTable.Routes.AddServiceRoute<Quote>("quote", configuration);
RouteTable.Routes.AddServiceRoute<OrderForm>("order-form", configuration);
```

Methods

```
[WebGet(UriTemplate = "{id}")]
public Shop Get(string id, HttpRequestMessage request,
  HttpResponseMessage response)
{
   ...
```

Links

```
return new ShopBuilder(new Uri("http://restbucks.com/"))
   .AddItem(new Item("coffee beans", new Amount("g", 250)))
   .AddLink(new Link(
     new Uri("quote/" + quoteId, UriKind.Relative),
     RestbucksMediaType.Value, LinkRelations.Self))
   .AddLink(new Link(
     new Uri("order-form/" + quoteId, UriKind.Relative),
     RestbucksMediaType.Value, LinkRelations.OrderForm))
   .Build();
```

Solution: UriFactory

```
[UriTemplate("quote", "{id}")]
public class Quote
{
}

[Test]
public void UriFactoryExample()
{
  var uriFactory = new UriFactory();
  uriFactory.Register<Quote>();

  Assert.AreEqual(
    new Uri("quote/1234", UriKind.Relative),
    uriFactory.CreateRelativeUri<Quote>(1234));

  Assert.AreEqual(
    new Uri("http://restbucks.com/quote/1234"),
    uriFactory.CreateAbsoluteUri<Quote>(
      new Uri("http://restbucks.com"), 1234));

  Assert.AreEqual(
    new Uri("http://restbucks.com/"),
    uriFactory.CreateBaseUri<Quote>(
      new Uri("http://restbucks.com/quote/1234")));
}
```

Registering resources at startup

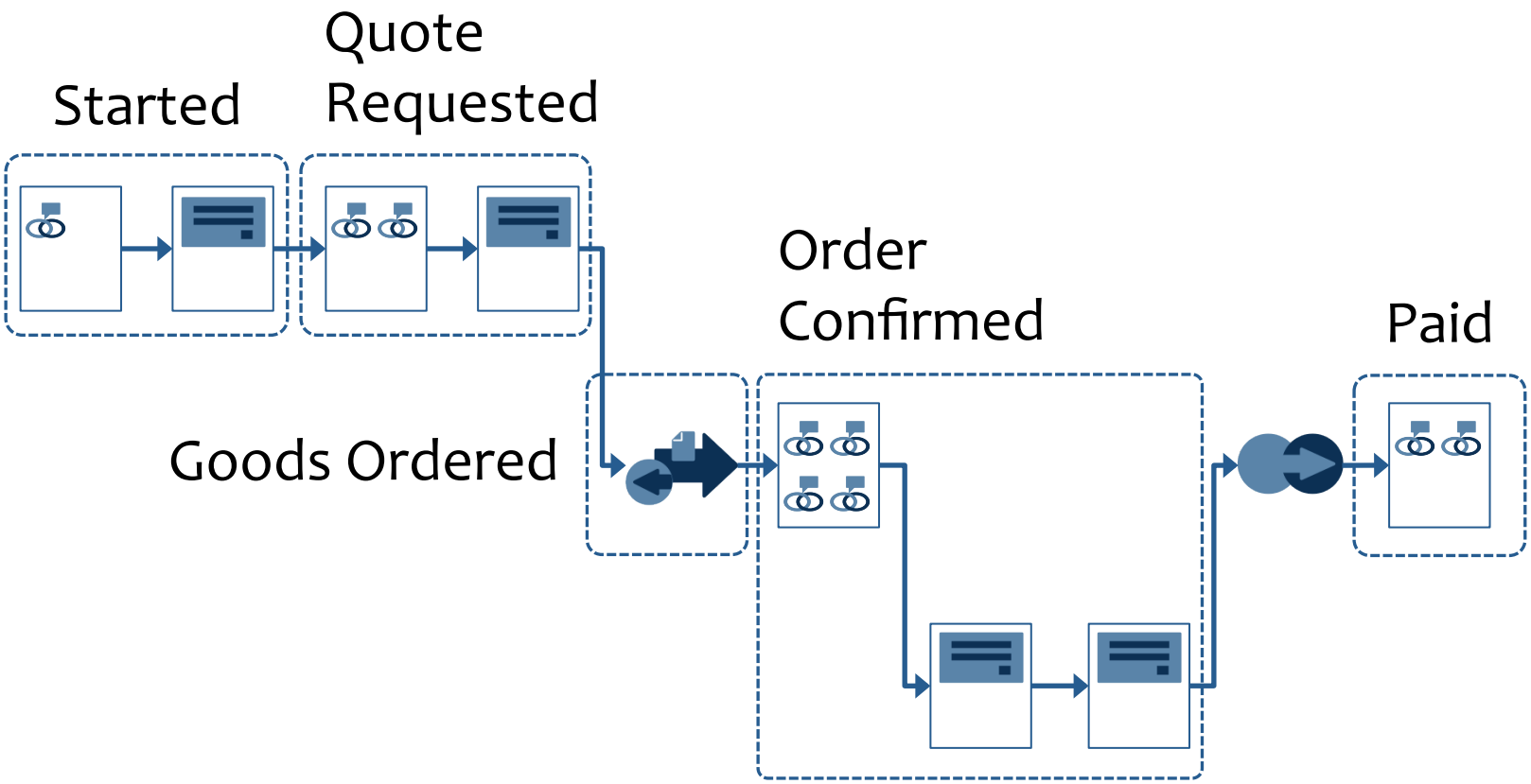
```
public void RegisterResourcesFor(Assembly assembly)
{
  var register = typeof(UriFactory).GetMethod("Register",
    BindingFlags.Instance | BindingFlags.NonPublic);

  var types = from t in assembly.GetTypes()
              where t.GetCustomAttributes(typeof (UriTemplateAttribute),
                false).Length > 0
              select t;

  types.ToList().ForEach(t =>
    {
      var genericMethod = register.MakeGenericMethod(new[] {t});
      genericMethod.Invoke(this, null);
    });
}
```

DRY URIs

```
[ServiceContract]
[UriTemplate("quote", "{id}")]
public class Quote
{
  ...

  [WebGet]
  public Shop Get(string id, HttpRequestMessage request,
                  HttpResponseMessage response)
  {
    ...

    var baseUri = uriFactory.CreateBaseUri<Quote>(request.Uri);

    return new ShopBuilder(baseUri, quote.LineItems.Select(
        li => new LineItemToItem(li).Adapt()))
      .AddLink(new Link(uriFactory.CreateRelativeUri<Quote>(quote.Id),
        RestbucksMediaType.Value, LinkRelations.Self))
      .AddLink(new Link(uriFactory
          .CreateRelativeUri<OrderForm>(quote.Id),
        RestbucksMediaType.Value, LinkRelations.OrderForm)).Build();
  }
}
```

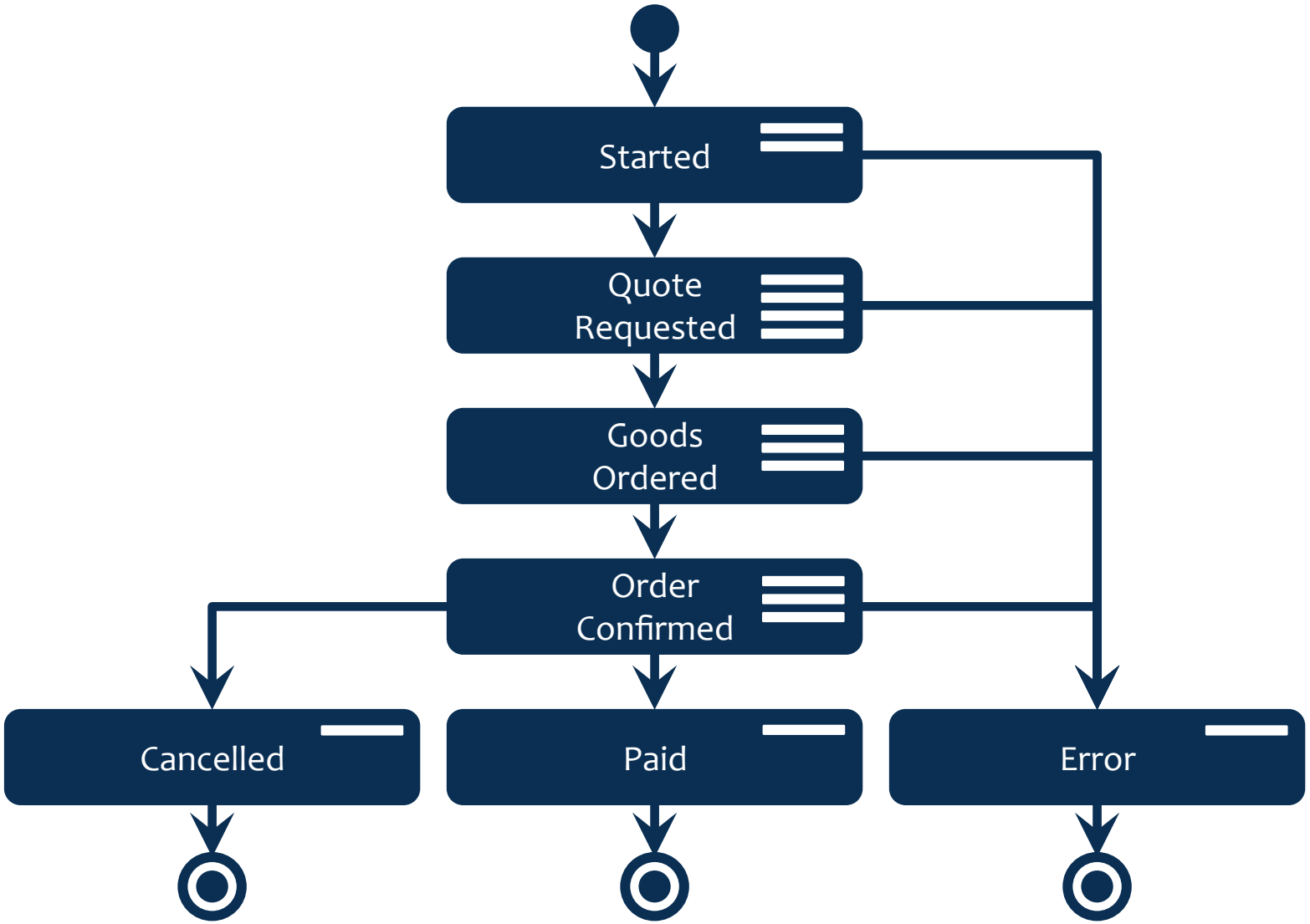
Client-side Development

The application is in the eye of the client

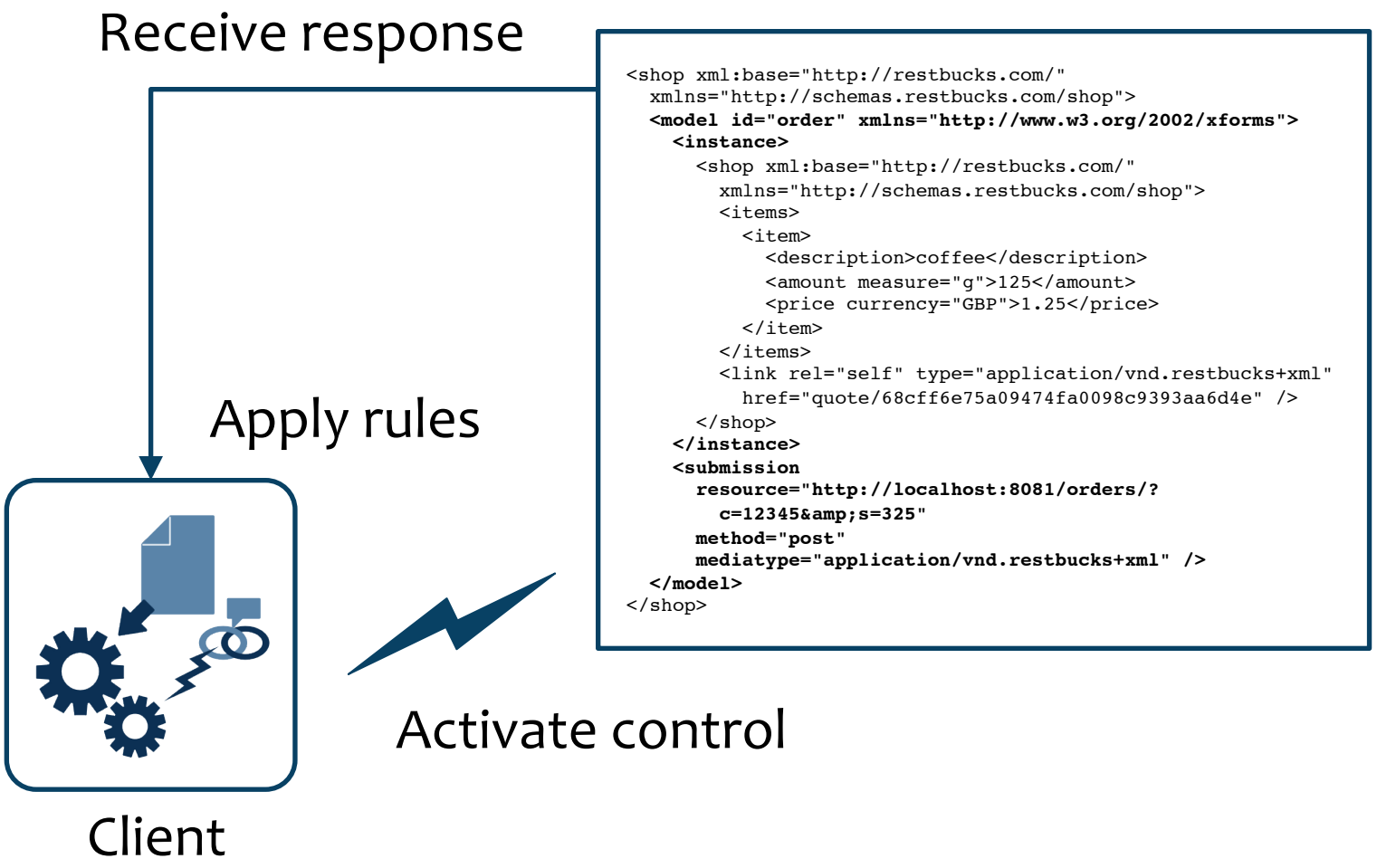


Clients *apply* resources to achieve an *application* goal

Client-side state machine: buckets of rules



Rules trigger events; hypermedia controls handle events



A bucket of rules

```
public class Started : IState
{
  public IState NextState(IClientCapabilities httpClient)
  {
    var rules = new Rules(
      When
        .IsTrue(response => prevResponse.ContainsForm(Forms.Rfq))
        .Invoke(actions  => actions.SubmitForm(Forms.Rfq))
        .Return(new[]{
           On.Status(HttpStatusCode.Created)
             .Do((response, vars) => new QuoteRequested(response, vars))}),
      When
        .IsTrue(response => prevResponse.ContainsLink(Links.Rfq))
        .Invoke(actions  => actions.ClickLink(Links.Rfq))
        .Return(new[]{
           On.Status(HttpStatusCode.OK)
             .Do((response, vars) => new Started(response, vars))}));

    return rules.Evaluate(previousResponse, stateVariables, httpClient);
  }

  public bool IsTerminalState { get { return false; } }
}
```

Get homepage

Request

```
GET /shop/ HTTP/1.1
Accept: application/vnd.restbucks+xml
Host: restbucks.com
```

Response

```
HTTP/1.1 200 OK
Cache-Control: max-age=86400, public
Content-Length: 285
Content-Type: application/vnd.restbucks+xml
Date: Wed, 11 May 2011 17:27:22 GMT

<shop xmlns:rb="http://relations.restbucks.com/"
      xml:base="http://restbucks.com/"
      xmlns="http://schemas.restbucks.com/shop">
  <link rel="rb:rfq prefetch"
        type="application/vnd.restbucks+xml"
        href="request-for-quote/" />
</shop>
```

Second rule fires

```
public class Started : IState
{
  public IState NextState(IClientCapabilities httpClient)
  {
    var rules = new Rules(
      When
        .IsTrue(response => prevResponse.ContainsForm(Forms.Rfq))
        .Invoke(actions  => actions.SubmitForm(Forms.Rfq))
        .Return(new[]{
           On.Status(HttpStatusCode.Created)
             .Do((response, vars) => new QuoteRequested(response, vars))}),
      When
        .IsTrue(response => prevResponse.ContainsLink(Links.Rfq))
        .Invoke(actions  => actions.ClickLink(Links.Rfq))
        .Return(new[]{
           On.Status(HttpStatusCode.OK)
             .Do((response, vars) => new Started(response, vars))}));

    return rules.Evaluate(previousResponse, stateVariables, httpClient);
  }

  public bool IsTerminalState { get { return false; } }
}
```

Get request for quote form

Request

```
GET /request-for-quote/ HTTP/1.1
Accept: application/vnd.restbucks+xml
Host: restbucks.com
```

Response

```
HTTP/1.1 200 OK
Cache-Control: max-age=86400, public
Content-Length: 385
Content-Type: application/vnd.restbucks+xml
Date: Wed, 11 May 2011 22:12:52 GMT

<shop xml:base="http://restbucks.com/"
      xmlns="http://schemas.restbucks.com/shop">
  <model id="request-for-quote"
         schema="http://schemas.restbucks.com/shop"
         xmlns="http://www.w3.org/2002/xforms">
    <instance />
    <submission resource="quotes/"
                method="post"
                mediatype="application/vnd.restbucks+xml" />
  </model>
</shop>
```

First rule fires

```
public class Started : IState
{
  public IState NextState(IClientCapabilities httpClient)
  {
    var rules = new Rules(
      When
        .IsTrue(response => prevResponse.ContainsForm(Forms.Rfq))
        .Invoke(actions  => actions.SubmitForm(Forms.Rfq))
        .Return(new[]{
           On.Status(HttpStatusCode.Created)
             .Do((response, vars) => new QuoteRequested(response, vars))}),
      When
        .IsTrue(response => prevResponse.ContainsLink(Links.Rfq))
        .Invoke(actions  => actions.ClickLink(Links.Rfq))
        .Return(new[]{
           On.Status(HttpStatusCode.OK)
             .Do((response, vars) => new Started(response, vars))}));

    return rules.Evaluate(previousResponse, stateVariables, httpClient);
  }

  public bool IsTerminalState { get { return false; } }
}
```

Alternate homepage (inlined form)

Request

```
GET /shop/ HTTP/1.1
Accept: application/vnd.restbucks+xml
Host: restbucks.com
```

Response

```
HTTP/1.1 200 OK
Cache-Control: max-age=86400, public
Content-Length: 470
Content-Type: application/vnd.restbucks+xml
Date: Wed, 11 May 2011 17:25:31 GMT

<shop xml:base="http://restbucks.com/"
      xmlns="http://schemas.restbucks.com/shop">
  <link rel="search" type="application/opensearchdescription+xml"
        href="search/" />
  <model id="request-for-quote"
         schema="http://schemas.restbucks.com/shop"
         xmlns="http://www.w3.org/2002/xforms">
    <instance />
    <submission resource="quotes/" method="post"
                mediatype="application/vnd.restbucks+xml" />
  </model>
</shop>
```

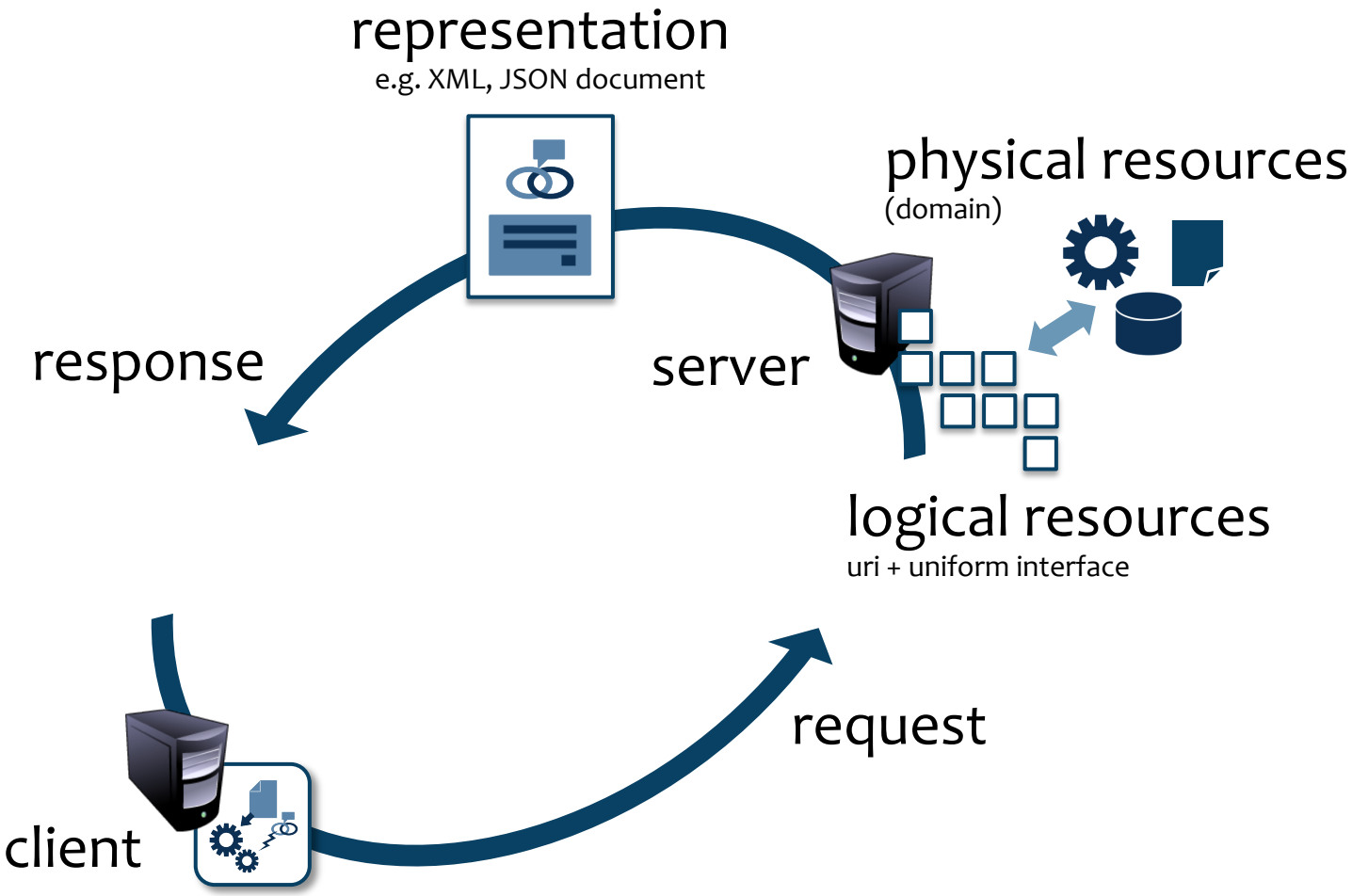
First rule fires

```
public class Started : IState
{
  public IState NextState(IClientCapabilities httpClient)
  {
    var rules = new Rules(
      When
        .IsTrue(response => prevResponse.ContainsForm(Forms.Rfq))
        .Invoke(actions  => actions.SubmitForm(Forms.Rfq))
        .Return(new[]{
           On.Status(HttpStatusCode.Created)
             .Do((response, vars) => new QuoteRequested(response, vars))}),
      When
        .IsTrue(response => prevResponse.ContainsLink(Links.Rfq))
        .Invoke(actions  => actions.ClickLink(Links.Rfq))
        .Return(new[]{
           On.Status(HttpStatusCode.OK)
             .Do((response, vars) => new Started(response, vars))}));

    return rules.Evaluate(previousResponse, stateVariables, httpClient);
  }

  public bool IsTerminalState { get { return false; } }
}
```

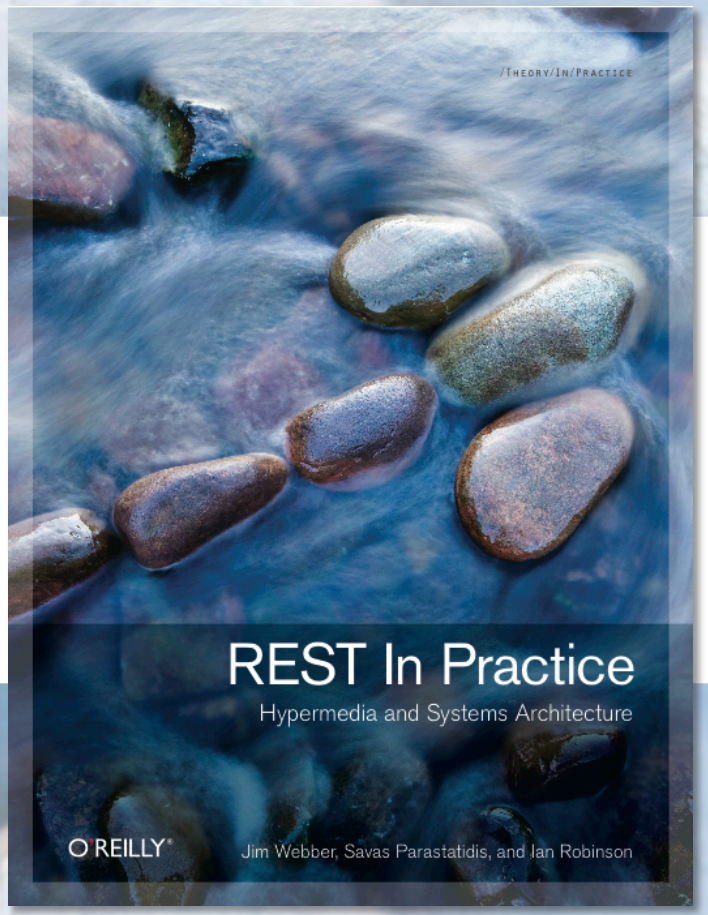
Running the client

```
var variables = ...

var state = new Uninitialized(variables);
var nextState = state.NextState(ClientCapabilities.Instance);

while (!nextState.IsTerminalState)
{
  nextState = nextState.NextState(ClientCapabilities.Instance);
}
```

Result: a hypermedia system



Questions?



http://ian**S**robinson.com
@ian**S**robinson